Helsinki University of Technology
Publications in Telecommunications Software and Multimedia
Teknillisen korkeakoulun tietoliikenneohjelmistojen ja multimedian julkaisuja
Espoo 2007 TML-A20

THEORY AND ALGORITHMS FOR EFFICIENT PHYSICALLY-BASED ILLUMINATION

Jaakko Lehtinen

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 31st of August, 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.tkk.fi/Diss/2007/isbn9789512289004/

ISBN 978-951-22-8899-1 (printed version)
ISSN 1456-7911
ISBN 978-951-22-8900-4 (electronic version)
ISSN 1455-9722

Multiprint Oy/Otamedia
Espoo 2007

ABSTRACT

Author Jaakko Lehtinen
Title Theory and Algorithms for Efficient Physically-Based Illumination

Realistic image synthesis is one of the central fields of study within computer graphics. This thesis treats efficient methods for simulating light transport in situations where the incident illumination is produced by non-pointlike area light sources and distant illumination described by environment maps. We describe novel theory and algorithms for physically-based lighting computations, and expose the design choices and tradeoffs on which the techniques are based.

Two publications included in this thesis deal with precomputed light transport. These techniques produce interactive renderings of static scenes under dynamic illumination and full global illumination effects. This is achieved through sacrificing the ability to freely deform and move the objects in the scene. We present a comprehensive mathematical framework for precomputed light transport. The framework, which is given as an abstract operator equation that extends the well-known rendering equation, encompasses a significant amount of prior work as its special cases. We also present a particular method for rendering objects in low-frequency lighting environments, where increased efficiency is gained through the use of compactly supported function bases.

Physically-based shadows from area and environmental light sources are an important factor in perceived image realism. We present two algorithms for shadow computation. The first technique computes shadows cast by low-frequency environmental illumination on animated objects at interactive rates without requiring difficult precomputation or a priori knowledge of the animations. Here the capability to animate is gained by forfeiting indirect illumination. Another novel shadow algorithm for off-line rendering significantly enhances a previous physically-based soft shadow technique by introducing an improved spatial hierarchy that alleviates redundant computations at the cost of using more memory.

This thesis advances the state of the art in realistic image synthesis by introducing several algorithms that are more efficient than their predecessors. Furthermore, the theoretical contributions should enable the transfer of ideas from one particular application to others through abstract generalization of the underlying mathematical concepts.

UDC 004.925, 004.925.3
Keywords computer graphics, shading, global illumination, indirect illumination, precomputed radiance transfer

TIIVISTELMÄ

Tekijä	Jaakko Lehtinen
Työn nimi	Theory and Algorithms for Efficient Physically-Based Illumination

Tämä tutkimus käsittelee realististen kuvien syntetisointia tietokoneella tilanteissa, jossa virtuaalisen ympäristön valonlähteet ovat fysikaalisesti mielekkäitä. Fysikaalisella mielekkyydellä tarkoitetaan sitä, että valonlähteet eivät ole idealisoituja eli pistemäisiä, vaan joko tavanomaisia pinta-alallisia valoja tai kaukaisia ympäristövalokenttiä (environment maps). Väitöskirjassa esitetään uusia algoritmeja, jotka soveltuvat matemaattisesti perusteltujen valaistusapproksimaatioiden laskentaan erilaisissa käyttötilanteissa.

Esilaskettu valonkuljetus on yleisnimi reaaliaikaisille menetelmille, jotka tuottavat kuvia staattisista ympäristöistä siten, että valaistus voi muuttua ajon aikana vapaasti ennalta määrätyissä rajoissa. Tässä työssä esitetään esilasketulle valonkuljetukselle kattava matemaattinen kehys, joka selittää erikoistapauksinaan suuren määrän aiempaa tutkimusta. Kehys annetaan abstraktin lineaarisen operaattoriyhtälön muodossa, ja se yleistää tunnettua kuvanmuodostusyhtälöä (rendering equation). Työssä esitetään myös esilasketun valonkuljetuksen algoritmi, joka parantaa aiempien vastaavien menetelmien tehokkuutta esittämällä valaistuksen funktiokannassa, jonka ominaisuuksien vuoksi ajonaikainen laskenta vähenee huomattavasti.

Fysikaalisesti mielekkäät valonlähteet tuottavat pehmeäreunaisia varjoja. Työssä esitetään uusi algoritmi pehmeiden varjojen laskemiseksi liikkuville ja muotoaan muuttaville kappaleille, joita valaisee matalataajuinen ympäristövalokenttä. Useimmista aiemmista menetelmistä poiketen algoritmi ei vaadi esitietoa siitä, kuinka kappale voi muuttaa muotoaan ajon aikana. Muodonmuutoksen aiheuttaman suuren laskentakuorman vuoksi epäsuoraa valaistusta ei huomioida. Työssä esitetään myös toinen uusi algoritmi pehmeiden varjojen laskemiseksi, jossa aiemman varjotilavuuksiin (shadow volumes) perustuvan algoritmin tehokkuutta parannetaan merkittävästi uuden hierarkkisen avaruudellisen hakurakenteen avulla. Uusi rakenne vähentää epäoleellista laskentaa muistinkulutuksen kustannuksella.

Työssä esitetään aiempaa tehokkaampia algoritmeja fysikaalisesti perustellun valaistuksen laskentaan. Niiden lisäksi työn esilaskettua valonkuljetusta koskevat teoreettiset tulokset yleistävät suuren joukon aiempaa tutkimusta ja mahdollistavat näin ideoiden siirron erityisalalta toiselle.

UDC	004.925, 004.925.3
Avainsanat	tietokonegrafiikka, realistinen kuvasynteesi, esilaskettu valonkuljetus, varjoalgoritmit

Äitini muistolle

PREFACE

This research was carried out during 2003-2007 in the Telecommunications Software and Multimedia Laboratory, Helsinki University of Technology, Espoo, Finland.

I sincerely thank my supervisor Prof. Lauri Savioja for the invaluable support he has given me and the whole graphics research group throughout the exciting period that in a way ends with the publication of this thesis. He has spared no effort in making the group a place to let ideas take their natural course without having to worry too much about the extraneous.

Second, I am most indebted to my instructor, Dr. Timo Aila, for his concrete and to-the-point guidance and collaboration during the preparation of this thesis and the articles it consists of. His no-nonsense approach to doing research has been, and still is, a great example.

I thank my co-authors Dr. Jan Kautz and Dr. Samuli Laine for the fruitful collaboration. I am lucky to have had the chance to work with such brilliant people. In addition, I wish to thank my former colleagues Dr. Janne Kontkanen, Jussi Räsänen, and Ari Silvennoinen for the many intriguing discussions we have had over the years about computer graphics and almost all other subjects imaginable. Furthermore, I am grateful to Ville Miettinen for many insightful discussions. I feel his incredible energy and passion for graphics have been instrumental in shaping the vital Helsinki graphics community we see today.

My work at Remedy Entertainment was the initial impulse that led me to pursue a doctorate in computer graphics. I express my gratitude to all the good people at Remedy, particularly Markus Mäki and Matias Myllyrinne, for being flexible in a way that has allowed me to combine exciting industry work and research in a most satisfying manner.

Furthermore, I thank the pre-examiners of this thesis, Dr. Nicolas Holzschuch and Dr. Tim Weyrich, for their positive feedback and constructive comments.

This work has been generously supported by an NVIDIA Fellowship 2006-2007, Helsinki Graduate School on Computer Science and Engineering (HeCSE), Anima Vitae, AMD Finland (ex-Bitboys), Nokia, NVIDIA Helsinki (ex-Hybrid Graphics), Remedy Entertainment, and the National Technology Agency of Finland (TEKES).

Finally, I thank my family, father and Juha, for support during the ups and downs encountered along the way.

Otaniemi, Espoo, 3rd August 2007

Jaakko Lehtinen

CONTENTS

LIST OF PUBLICATIONS

This thesis summarizes the following articles and publications, referred to as [P1]–[P4]:

[P1] J. Lehtinen. A Framework for Precomputed and Captured Light Transport. Accepted for publication in *ACM Transactions on Graphics*, ACM Press. 36 pages.

[P2] J. Lehtinen and J. Kautz. Matrix Radiance Transfer. In *Proceedings of ACM Siggraph 2003 Symposium on Interactive 3D Graphics*, pages 59–64, ACM Press, 2003.

[P3] J. Kautz, J. Lehtinen and T. Aila. Hemispherical Rasterization for Self-Shadowing of Dynamic Objects. In *Rendering Techniques 2004* (Eurographics Symposium on Rendering), pages 179–184, Eurographics Association, 2004.

[P4] J. Lehtinen, S. Laine and T. Aila. An Improved Physically-Based Soft Shadow Volume Algorithm. *Computer Graphics Forum*, 25(3):303–312, Eurographics Association, 2006.

LIST OF ABBREVIATIONS

2D	Two-dimensional
3D	Three-dimensional
4D	Four-dimensional
BRDF	Bidirectional reflectance distribution function
CPU	Central processing unit
GPU	Graphics processing unit
FEM	Finite element method
PCA	Principal component analysis
PRT	Precomputed radiance transfer
SH	Spherical harmonics

1 INTRODUCTION

This thesis deals with the computerized generation of pictures of virtual scenes. This *realistic image synthesis* is one of the central fields of study within computer graphics; its goal is to efficiently produce images of virtual scenes such that the image is perceived as "realistic" by viewers. In this context, realism is thought to be achieved through the rigorous solution of a physically-based equation that describes the optical effects that determine the appearance of the scene. The road to realism is a long and arduous one, beginning with the difficulty of modeling and representing the environment, i.e., its geometry and surface materials, continuing with the difficult task of efficiently simulating the propagation of light within the scene until its arrival on a virtual image sensor, and ending with the problems associated to presenting the final radiometric result to the user in a way that best suits the application. The whole virtual imaging pipeline may thus be described as follows.

1. Model geometry and materials
2. Simulate lighting
3. Present to user (tone map)

While only Step 2 is dealt with in this thesis, the other two are rich research topics of their own, each with their particular problems and algorithms. Modeling and capturing the geometry and reflectance properties of real scenes is an active and difficult research topic that continuously grows in importance as the computational capabilities of computers grow, and modeling the scenes of increasing complexity using traditional modeling tools consequently becomes constantly more laborious. Also, presenting the simulation result to the user is a difficult task due to the still limited dynamic ranges of current display devices and the uncontrollable lighting conditions in which the results are often viewed. The difficulties are compounded in applications such as games, where the aim is to convey a particular *mood* to the viewer, i.e., artistic requirements come into play.

Realistic image synthesis means generating pictures of virtual scenes through the simulation of the propagation of light, so-called *light transport*, within the scene. Usually, light transport is modeled by the *rendering equation* [49], an integral equation that describes common visual effects to a mostly satisfactory degree. These effects include soft shadows, indirect illumination and arbitrary surface reflectance.

Solving the rendering equation is challenging for two main reasons. First, it is an integral equation, i.e., its unknown is a function. Because of this, computing exact solutions is impossible except in the most trivial and thus practically uninteresting cases. As a consequence, the solutions must be approximated by numerical methods. Second, the propagation of light is an inherently recursive process: Light emitted from the light sources hits the surfaces of the scene, reflects or refracts there, propagates further after this surface interaction to hit other surfaces, and so on. This means that in principle, the illumination and reflective properties at one location in the scene affect all other locations as well. This perplexing complexity is directly reflected in the numerical methods used for approximating the solutions. The term “global illumination” has been coined to denote the solution of the full recursive rendering equation. Many methods omit indirect illumination from the mathematical model for performance reasons.

This thesis treats only well-founded numerical methods. Well-foundedness is here understood to mean that the numerical solutions converge towards the true solution under the chosen mathematical model as more time and resources are invested. This is in contrast to heuristic techniques often employed in scenarios with strict timing constraints (such as games), where the correctness of the solution can be traded for fast execution times at the cost of only qualitatively mimicing the true solution.

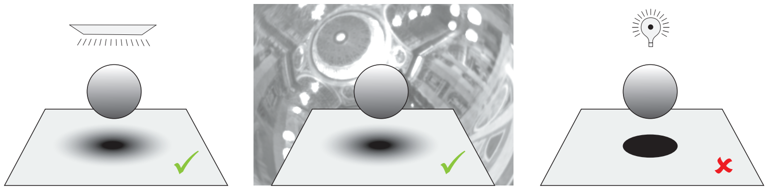
The above-mentioned difficulties associated with generating realistic images are mostly of a practical nature. Indeed, simple numerical techniques that produce approximations arbitrarily close to the true solution in all well-posed situations do exist. However, no guarantees of their performance can be given, i.e., they may take an arbitrarily long time to produce the answer. In fact, it is easy to construct geometric configurations where the time taken by many algorithms for rendering a correct picture can be made arbitrarily large.[1] Such unpredictable behavior is, naturally, unacceptable in practical usage situations.

Due to the these difficulties, no single one-for-all algorithm that would suit all realistic image synthesis problems has surfaced as yet. Thus, practical algorithms must be adapted to the situations they are to be used in, and some features of light transport must be forfeited in order to gain efficiency. This adaptation consists of two steps: Fixing the requirements, and then subsequently designing the algorithm to compute a solution that fulfills these requirements (and ideally nothing else) in an as-efficient-as-possible manner, utilizing all prior knowledge of the geometric situation and other known aspects of the problem at hand. In practice, this process is not one-way, but can rather be thought of as making trade-offs. For instance, if it is acceptable that indirect illumination is not accounted for or only crudely approximated, interactive deformation of the scene becomes much more tractable. In an opposite vein, fixing the scene to remain static enables interactive relighting in precomputed light transport techniques. All techniques described in this thesis are based on such trade-offs.

1.1 Scope of this Thesis

This thesis deals with realistic image synthesis using *physically-based incident illumination*. By this we mean that not only is the propagation of light computed using well-founded approximations of the governing equations as described above, but also that the sources of illumination have some physical significance. Such lights include area and distant environmental sources, but exclude point-like sources. These are illustrated in Figure 1.1. The use of area lights adds significantly to the perceived realism of images

[1] For example, situations where the light must bounce a large number of times before reaching the virtual camera.



(a) (b) (c)

Figure 1.1: *Physically-based incident illumination.* This thesis treats realistic image synthesis using area light sources (a) and distant environmental illumination (b), but not point light sources (c). Background lighting environment courtesy of Paul Debevec (`www.debevec.org`).

[73], owing to the fact that the shadows produced by area light sources are smooth; the perfectly sharp shadow edges caused by point light sources are mostly regarded as unrealistic by human observers.

This thesis rests on four publications [P1]–[P4]. They describe novel theory and techniques for efficient realistic image synthesis using physically-based incident illumination in various scenarios: [P1] and [P2] deal with precomputed light transport in fixed scenes, while [P3] and [P4] describe algorithms for computing physically-based shadows. See Table 1.1 for a classification.

We assume the common ray optics model for light transport. Thus wave effects like diffraction, seen for instance in the colored reflections on CDs, are not accounted for. Fluorescence and phosphorescence are also not dealt with. We follow the common convention of writing equations for a single wavelength of light, and note that in practice the equations hold and are solved separately for a number of color bands, typically red, green and blue. We further limit the scope to non-participating media.

We assume the reader has a basic working knowledge of realistic image synthesis and the associated mathematics. Accordingly, the primer given in Chapter 1.3 is rather concise.

Publication	**Category**	**Interactive**	**Deformation**	**Indirect**
[P1]	PLT (theory)	yes	no	yes
[P2]	PLT	yes	no	yes
[P3]	shadows	yes	yes	no
[P4]	shadows	no	no	no

Table 1.1: An overview of the algorithms presented in the publications that make up this thesis. Publication [P1] presents a theoretical framework for precomputed light transport (PLT). Publications [P2] and [P3] describe interactive techniques for global illumination and shadows on deformable objects, respectively, while Publication [P4] presents a physically-based soft shadow algorithm for off-line use.

1.2 Organization of the Thesis

Section 1.3 gives a brief mathematical introduction to realistic image synthesis. Chapter 2 reviews previous related research. Chapter 3 describes the novel theory and techniques for realistic image synthesis from Publications [P1]–[P4], while Chapter 4 summarizes the contributions made by the author.

1.3 Mathematical Preliminaries for Realistic Image Synthesis

Integral Operators and Equations. The mathematical language of global illumination is that of functional analysis [57] and integral operators. It can be thought of as the generalization of elementary linear algebra to linear vector spaces of infinite dimensions; here, functions defined on some domain take the role of vectors, and matrices are replaced with linear integral operators.[2] An example of a space like this is the set of all possible diffuse radiosity functions defined on the surfaces of a three-dimensional scene. The dimension of such a space is infinite, since no finite number of parameters can describe an arbitrary function, while, in contrast, a vector in $\mathbb{R}^n$ is completely specified by n coordinates. Particularly, an integral operator $\mathcal{K} : X \to Y$ acts on a function f in vector space X and produces another function, denoted by $\mathcal{K}f$, in vector space Y. It is defined through

$$(\mathcal{K}f)(s) = \int_S f(t)\, k(s,t)\, dt. \tag{1.1}$$

Here S denotes the domain on which f is defined, and k is the *kernel* of the operator. Note that this definition closely resembles matrix-vector multiplication; only the finite sum has been replaced by an integral, and the "matrix" k is consequently a function of the continuous variables s and t instead of a table indexed by rows and columns. As in the case of non-square matrices in finite-dimensional vector spaces, an integral operator need not map from a vector space onto itself, i.e., X does not necessarily equal Y. One of these spaces may also be a usual finite-dimensional vector space – Publication [P1] makes use of such operators.

The generalized counterparts to usual finite linear systems of equations are linear integral equations [56, 57, 7, 6, 39]. The equations that describe light transport have the special form

$$L = E + \mathcal{T}L \quad \Leftrightarrow \quad (\mathcal{I} - \mathcal{T})L = E. \tag{1.2}$$

Here L is the unknown function, $\mathcal{T}$ is a so-called *transport operator* which will be described below, and E is a function that describes the light sources within the scene. Equations that have the form (1.2) are called *Fredholm equations of the second kind*. Formally, the solution of such an equation is given by $L = (\mathcal{I} - \mathcal{T})^{-1}E$. The difficulty lies in the fact that the inverse

[2] Other types of linear operators, such as differential operators, are important in many applications of functional analysis, including other branches of computer graphics. However, we limit the exposition to integral operators for the reason that they are the ones utilized in image synthesis.

operator is itself an integral operator, and its explicit computation is most often impossible; how do you invert an infinite matrix? This complication forces solutions to be sought in other ways.

The Neumann series. Provided that the operator $\mathcal{T}$ satisfies a condition that can be described loosely speaking as "not adding energy when acting on a function", Equation (1.2) has the formal series solution

$$L = \sum_{i=0}^{\infty} \mathcal{T}^i E = E + \mathcal{T}E + \mathcal{T}^2 E + \ldots \tag{1.3}$$

This is called the *Neumann series*. More precisely, the above-mentioned condition is that the so-called norm of $\mathcal{T}$ must be strictly less than unity, but we skip the details for brevity, and assure the reader that indeed this is the case in well-posed global illumination problems, and refer to textbooks on functional analysis [57, 7] for further details. The interested reader may consult Arvo [3] on the choice of norm for image synthesis problems.

Subspaces, Bases, and Projectors. A *subspace* X_h of a vector space X is a subset of the elements of X such that the subset itself forms a vector space. A simple, familiar example is the set of points on a plane that goes through the origin in $\mathbb{R}^3$: The points on the plane form a subspace of $\mathbb{R}^3$. Now if we pick a finite collection $\{\phi_i\}_{i=1}^n$ of n elements from a vector space X, a particularly useful finite-dimensional subspace X_h is the *linear span* of these elements, i.e., the set of all linear combinations of these functions. Such a combination may always be written as

$$X_h \ni f = \sum_{i=1}^{n} \alpha_i \, \phi_i, \quad \text{such that } f(s) = \sum_{i=1}^{n} \alpha_i \, \phi_i(s), \tag{1.4}$$

i.e., f is completely specified by the n real numbers α_i. The collection $\{\phi_i\}_{i=1}^n$ is called a *basis* for X_h, and the individual ϕ_i are referred to as *basis functions*. The sum in Equation (1.4) is called a *basis expansion*. The dimension of X_h is the number of linearly independent ϕ_i.

A *projector* is a special type linear operator that can be used to force a function f from X onto a subspace X_h such that the difference between f and the projection $\mathcal{P}f$ is "small". Most notable projectors are *least squares* projectors and *interpolatory* projections. Least squares projectors produce the coefficients α_i in such a way that the squared difference $(f - \mathcal{P}f)^2$ integrated over the domain (this is called the L_2 norm [57]) is minimized, while interpolatory projectors produce functions that interpolate f in some n prescribed points within the domain. Projectors have the property that $\mathcal{P}^2 = \mathcal{P}$, i.e., the projector does not change a function that already lies in the subspace.

The Finite Element Method. Finite element methods (FEM) form a large class of numerical methods for solving operator equations. Most of them can be described in an abstract form using projectors. The basic idea is to search for an approximate solution as an expansion in some suitable function basis, and use a corresponding projector to *discretize* Equation (1.2). Denoting the sought-after approximate solution by L_h and applying $\mathcal{P}$ to Equation (1.2) from the left yields [6]

$$\mathcal{P}(\mathcal{I} - \mathcal{T})L_h = \mathcal{P}E, \quad L_h \in X_h. \tag{1.5}$$

Substituting the basis expansion (1.4) for L_h and recalling that $\mathcal{P}$ produces basis expansions reveals that this is actually the finite linear system of equations

$$(\boldsymbol{I} - \boldsymbol{T})\boldsymbol{\alpha} = \boldsymbol{\beta}, \tag{1.6}$$

where the ij-th entry of the discretized transport operator $\boldsymbol{T}$ is the i-th projection coefficient of $\mathcal{T}\phi_j$, the vector $\boldsymbol{\alpha}$ gives L_h in terms of the basis functions through Equation (1.4), and $\boldsymbol{\beta}$ is the vector $\mathcal{P}E$ of projection coefficients of the right-hand side E. See for instance the references [6, 7, 39] for details. Many realistic image synthesis techniques employ finite elements, as will be seen in Chapter 2.

Monte Carlo Integration. The mathematics of realistic image synthesis involves a significant amount of integrals. Consequently, computer programs designed to numerically solve such operator equations, even in their discretized form, must evaluate a large number of them. In almost all cases of practical interest this is impossible in closed form, i.e., through symbolic manipulations, and thus numerical approximations must be employed. The most widely-used class of such numerical techniques is Monte Carlo integration in its many forms. The basic idea is to draw a number of independent random point samples from the domain of the integrand and average the function values at those points, while simultaneously accounting for the distribution used for generating the point samples. This average, call it I, is itself a random variable, and it turns out that its expected value equals the value of the integral. Formally:

$$I := \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}, \quad E\{I\} = \int_S f(s)\,ds. \tag{1.7}$$

Here p is the probability distribution function (PDF) according to which the samples x_i are drawn, and $E\{\cdot\}$ denotes expected value. The simplest form of Monte Carlo integration is to use a uniform PDF, i.e., $p \equiv 1/|S|$, while more sophisticated *importance sampling* techniques construct PDFs whose form follows that of f; this results in lower *variance*, i.e. less noise, in the results. Veach [87, Chap. 2] provides a concise exposition of Monte Carlo techniques and the related statistical machinery.

Radiance and the Rendering Equation. *Radiance* is the fundamental quantity that describes the spatioangular steady-state distribution of light energy, assuming the ray optics model. It is defined as power per unit projected area per unit solid angle [24]. One of its key characteristics is that it stays constant along straight lines in a non-scattering medium that has a constant index of refraction, i.e., light travels unchanged along straight lines.

The rendering equation [49] describes the propagation of light in a scene devoid of participating medium. Its solution is a function that determines the radiance at each point and each direction on the surfaces of the scene; lighting features such as shadows and indirect illumination are thus also present in the solution. The equation is written in its modern form as

$$L(x \to \omega_{\text{out}}) = E(x \to \omega_{\text{out}}) + \int_{\Omega(x)} L(x \leftarrow \omega_{\text{in}})\, f_r(x, \omega_{\text{in}} \to \omega_{\text{out}})\, \cos\theta \, d\omega_{\text{in}}. \quad (1.8)$$

Here x is a point on the surface of the scene, $L(x \to \omega_{\text{out}})$ is the unknown function that denotes the radiance leaving x towards direction ω_{out}, $L(x \leftarrow \omega_{\text{in}})$ is the radiance arriving at x from direction ω_{in}, $\Omega(x)$ is the hemisphere oriented to the direction of the surface normal at x, f_r is the bidirectional reflectance distribution function (BRDF) [69] that captures the local reflectance properties of the surface, and θ is the angle between ω_{in} and the surface normal at x. E is the radiance emitted from x towards ω_{out}. The equation states simply that the radiance leaving x towards ω_{out} is the sum of emitted radiance and reflected radiance, which is expressed by the reflectance integral.[3] The equation is recursive, since due to the constancy of radiance along straight lines, the incident radiance $L(x \leftarrow \omega_{\text{in}})$ is exactly the radiance *leaving* the scene point that is seen when looking from x towards ω_{in}.

We will now rewrite the rendering equation using operator notation to get to the abstract form of Equation (1.2). We first define a *propagation operator* $\mathcal{G}$ and a *local reflection operator* $\mathcal{R}$. This formulation, apart from minor notational changes, is borrowed from Arvo et al. [4]. Let us first define the propagation operator $\mathcal{G}$ that turns a function that describes outgoing light from surface points into a function that describes incident lighting by transporting the light along straight lines:

$$(\mathcal{G}L)(x \leftarrow \omega_{\text{in}}) = L(r(x, \omega_{\text{in}}) \to -\omega_{\text{in}}). \quad (1.9)$$

Here $r(x, \omega)$ is the ray-casting function that returns the closest point to x in the direction ω. This operator merely looks from x towards ω_{in} and evaluates the radiance leaving the surface point $r(x, \omega_{\text{in}})$ seen in that direction towards x. Because of this function, *visibility* has a significant role in global illumination. It is the cause of shadows, and its evaluation eats up a large portion of the computational resources spent on image synthesis. Next, we define the local reflection operator $\mathcal{R}$ as

$$(\mathcal{R}L)(x \to \omega_{\text{out}}) = \int_{\Omega(x)} L(x \leftarrow \omega_{\text{in}})\, f_r(x, \omega_{\text{in}} \to \omega_{\text{out}})\, \cos\theta \, d\omega_{\text{in}}. \quad (1.10)$$

[3] The rendering equation is only an approximation, although a very common one, to surface reflection. This is because the BRDF model assumes that incident light reflects off the surface at the point of incidence. In contrast, many real materials exhibit subsurface scattering, i.e., the light penetrates the material and travels some distance before exiting at another location. Human skin is a familiar example of such behavior.

This operator takes the incident illumination $L(x \leftarrow \omega_{\text{in}})$ at x from all directions, and turns the incident illumination into outgoing radiance according to the local reflectance properties of the surface. Note that indeed this operator is *local* in the sense that it only involves the radiance function at one point x. Now, the *transport operator* is defined as the concatenation of these operators:

$$\mathcal{T} = \mathcal{R}\mathcal{G}. \tag{1.11}$$

This operator takes a function that describes the spatioangular distribution of outgoing radiance from the surfaces of the scene, propagates it along straight lines to yield an incident radiance distribution, and applies the BRDF in the local reflection step in order to produce another outgoing radiance distribution. In other words, the operator computes the effect of exactly one reflection on a lighting function. Finally, the operator version of the rendering equation is

$$L = E + \mathcal{T}\, L. \tag{1.12}$$

The Neumann series solution from Equation (1.3) simply states that the global illumination solution is the sum of emitted light E, direct (once-reflected) light $\mathcal{T}E$, twice-reflected light $\mathcal{T}^2E$, and so on.

The above derivation of the rendering equation uses the outgoing radiance function as the unknown. An equivalent formulation results if we write the equation for unknown *incident* radiance instead. The incident form of the rendering equation is

$$L_{\text{i}} = E_{\text{i}} + \mathcal{T}_{\text{i}} L_{\text{i}}, \tag{1.13}$$

where the incident transport operator $\mathcal{T}_{\text{i}} = \mathcal{G}\mathcal{R}$, L_{i} denotes $L(x \leftarrow \omega_{\text{in}})$, and $E_{\text{i}} = \mathcal{G}\, E$. The outgoing radiance solution that corresponds to L_{i} is obtained by applying $\mathcal{R}$ one final time.

Diffuse and Non-diffuse Reflection. A surface whose BRDF is constant with respect to the angular arguments, i.e., $f_r(x, \omega_{\text{in}} \rightarrow \omega_{\text{out}}) \equiv f_r(x)$, is called *diffuse* or *Lambertian*, and the light reflected by such a surface is called *radiosity*. Inspection of Equation (1.10) quickly reveals that when this is the case, the surface reflects light to all directions in equal proportions, i.e., the appearance depends only on x, not the viewing direction. This situation is considerably simpler than the case of arbitrary (*glossy* or *specular*) reflectance, since radiosity is a 2D function instead of a 4D function. Many global illumination techniques take advantage of this fact.

Linear Measurements and Adjoint Operators. What are the values that need to be assigned to pixels when rendering an image? Measurement devices such as pixels of a camera sensor are directly sensitive to radiance [24], and thus it suffices to sample the footprint of the pixel and compute a suitable average of the radiance sent by the surfaces towards the pixel.[4] Computing this average may be formulated as integrating the product of a

[4]The exact weights of this average are determined by a so-called pixel sampling filter.

certain weight function w and the global illumination solution. We denote this by[5]

$$\int_S \int_\Omega w(x,\omega)\, L(x,\omega)\, \cos\theta \, d\omega \, dA_x := \langle w, L\rangle = \langle w, \sum_{i=0}^{\infty} \mathcal{T}^i E\rangle, \quad (1.14)$$

where the last equality is due to the Neumann series that determines the global illumination solution L. Now, the *adjoint* $\mathcal{T}^*$ [57] of the transport operator $\mathcal{T}$ is the unique linear operator that satisfies $\langle w, \mathcal{T}E\rangle = \langle \mathcal{T}^* w, E\rangle$ for all w, E. Applying this definition to Equation (1.14) yields

$$\langle w, L\rangle = \langle w, \sum_{i=0}^{\infty} \mathcal{T}^i E\rangle = \langle \sum_{i=0}^{\infty} \mathcal{T}^{*i} w, E\rangle = \\ \langle w, E\rangle + \langle \mathcal{T}^* w, E\rangle + \langle \mathcal{T}^*\mathcal{T}^* w, E\rangle + \dots \quad (1.15)$$

Just as the lighting solution may be computed bounce by bounce by transporting the emission E by $\mathcal{T}$, the value of the linear measurement may be computed by transporting the measurement function w by $\mathcal{T}^*$ bounce by bounce and integrating against the emission. This is what many ray tracers do; by "following reverse light paths", they actually propagate the measurement using the adjoint. In addition to pixel values, the computation of projection coefficients in basis expansions may be expressed as a series of linear measurements.

Distant Illumination. Many applications make use of distant incident illumination. This means that light emitted onto the scene is taken to originate from a large source whose distance to the scene is so large that the only variation in the illumination is angular, i.e., $L(x \leftarrow \omega_{\text{in}}) \approx E(\omega_{\text{in}})$. This has the effect that the direct lighting term $\mathcal{T}E$ is computed differently from the other terms (cf. the next subsection); the computation of the subsequent terms is unaffected. The distant source is not considered to be a part of the scene in the sense that it is not a part of the simulation after the first surface interaction. The distant light source is often called a *lighting environment*. The use of lighting environments captured from the real world has become hugely popular after Debevec and Malik introduced a method for capturing high dynamic range images from multiple photographs [29] and Debevec applied the resulting "light probes" for rendering synthetic objects in real scenes [28].

Direct vs. Indirect Illumination, Shadows, and Ambient Occlusion. The term $\mathcal{T}E$ describes light that has reflected off a surface exactly once before hitting the camera. It is called *direct illumination*. The subsequent bounces of light, when summed together, make up indirect illumination. The purpose of *shadow algorithms* is the computation of direct illumination. Direct and indirect illumination usually differ considerably in what is called their frequency content. Direct illumination usually contains discernible

[5] Many authors include the cosine term in the measure $d\omega$, but it may just as well be incorporated into the weighting function.

shadow boundaries. While the subsequent bounces indeed also contain indirect shadows, they are usually very smooth.[6] The visual significance of the sharp features present in direct illumination necessitates their accurate computation, while the smooth nature of indirect illumination often allows its computation to be performed using lower sampling rates [95] and more aggressive approximations. Durand et al. [32] study these phenomena by spectral analysis.

In situations where the light sources are compact – consider, for instance, the lighting fixtures in an architectural model – the emission function E is only non-zero on small areas compared to the whole extent of the scene. Computing the direct, shadowed illumination $\mathcal{T}E$ by evaluating the reflectance integral in Equation (1.8) over the whole hemisphere at the receiving points is inefficient in such cases, as the integrand has zero value over most of the hemisphere. It is rather more efficient to integrate over the surfaces of the light sources instead by changing variables:

$$(\mathcal{T}E)(x \to \omega_{\text{out}}) = \int_{\text{lights}} E(y \to x)\, f_r(x, y \to \omega_{\text{out}})\, V(x,y)\, \frac{\cos\theta\ \cos\theta'}{r^2} dA_y. \quad (1.16)$$

Here θ' is the angle between the surface normal at y and the vector separating x from y, r is the distance between x and y, and $V(x,y)$ is the *visibility function* that takes a value of 1 when the points are mutually visible and 0 otherwise. In the area formulation the visibility function takes the role of the ray cast function: Instead of explicitly finding the single point visible to x in a given direction by a ray cast, here we integrate over all the light sources, and pick out the one that is visible to x in a given direction by V.

Computation of direct illumination due to distant illumination does not usually benefit from the above change-of-variables treatment, since some light is usually assumed to originate from each direction. Concretely, the direct illumination due to an environmental source is

$$(\mathcal{T}E)(x \to \omega_{\text{out}}) = \int_{\Omega(x)} E(\omega_{\text{in}})\, V(x, \omega_{\text{in}})\, f_r(x, \omega_{\text{in}} \to \omega_{\text{out}})\ \cos\theta\, d\omega_{\text{in}}. \quad (1.17)$$

Here E is the distant incident illumination, V takes value 1 if the ray from x towards ω_{in} is not blocked by the scene and 0 otherwise, and the rest of the terms are as before. The problem is thus reduced to solving for the hemispherical visibility function at each receiver point. Shadows from environmental illumination call for specialized algorithms because of this.

Ambient occlusion is a general term for lighting solutions computed using a homogenous distant lighting environment ($E(\omega_{\text{in}}) \equiv$ const.). Ambient occlusion causes smooth shadows such that the intensity of the shadow at a surface point is related to the overall occlusion of the point by the rest of the scene. While usually only direct illumination is considered, it is also possible to compute ambient occlusion with multiple bounces.

[6]Here we mean "smooth" as opposed to "sharp", not the mathematical notion of smoothness.

2 RELATED RESEARCH

This chapter presents an overview of related research in physically-based rendering. The main foci are on finite element global illumination techniques and physically-based shadow algorithms for area and environmental light sources.

Global illumination algorithms can be roughly divided into two categories. On one hand there are techniques based on ray tracing; they compute point estimations of the solution of the rendering equation, usually using stochastic techniques. These methods are especially well-suited for directly generating pictures of the solution. This is because image sensors (pixels) are sensitive to radiance [24], and this is exactly what ray tracers excel in computing. Finite element methods form the other main category. These algorithms represent the lighting solutions as expansions in a finite-dimensional approximating function space. The versatility of these techniques lies in the fact that generally the solutions are not dependent on a particular viewpoint. This allows, for instance, interactive visualization of precomputed illumination solutions in architectural models in so-called walkthrough applications. The division between ray tracers and finite element techniques is not strict; indeed, many hybrid algorithms exist (e.g., [90, 14]).

We describe the evolution of finite element techniques in some detail to highlight the connections to precomputed light transport. Indeed, many of the recent precomputed light transport methods are directly based on the finite element method, which becomes apparent through Publication [P1]. In a more abstract vein, precomputed transport may be seen as a mathematical way of constraining the otherwise unwieldy computations that arise from finite element discretizations when interactive editing of the incident illumination is desired. Much of the earlier work on finite elements has the same abstract goal of intelligently reducing computation. Although they have their uses in image relighting techniques, algorithms based on path tracing, density estimation and/or adjoints are left out to limit the scope; the reader may consult any of the excellent textbooks [33, 72, 94] for an introduction. A survey on adjoints and the related concept of *importance* is provided by Christensen [17]. We also do not treat techniques for capturing light transport operators from real scenes, as none of the particular techniques presented in Publications [P2]–[P4] have that purpose.

2.1 Early Techniques

Realistic image synthesis in its modern form is firmly rooted in the seminal work of Kajiya [49] on his rendering equation. His work was preceded by several techniques that resemble modern global illumination algorithms, but were not based on an explicit physically-based operator equation. In essence, Kajiya presented a general framework and showed how seemingly different algorithms are best seen as different numerical techniques for solving the same equation under different assumptions.

The two general categories of global illumination algorithms are already visible in the early techniques that predate the rendering equation. Whitted's recursive ray tracer [96] generates images with direct illumination (including shadows from point sources) and perfect mirror reflections and refractions by following "reverse" paths of light from the image pixels into the scene, recursively spawning reflected and refracted rays at surfaces. The early work of Kay and Greenberg [52] also treats refractions. On the other front, the finite element method was introduced to computer graphics by Goral et al. [36], who imported a technique from the radiative heat transfer literature for simulating interreflections. Their technique computes a view-independent approximation of the equilibrium radiosity solution under the assumption that all surfaces are perfect Lambertian (diffuse) reflectors.

Kajiya [49] introduced the rendering equation to the computer graphics community in a form that differs somewhat from that given previously in Chapter 1 – his formulation is not in terms of radiance but a related quantity. The essence of the equation is not new; it has been known for long as one of the possible boundary conditions for the Boltzmann integrodifferential equation that describes the propagation of radiation in an environment with a scattering medium and reflective boundaries [13, 86].[1] However, Kajiya shows that this equation is a good model for image synthesis, and observes that the previous ray tracing and radiosity techniques can in fact be seen as solving the same equation by different approximations and based on different assumptions of the scene properties. He also describes a *path tracing* algorithm for solving the rendering equation in cases where the reflections are limited to neither Lambertian nor perfect mirrors. Many subsequent techniques have built on the same principle.

2.2 Finite Element Methods

Finite element global illumination methods have a long history. Since the introduction of radiosity by Goral et al. [36], much research effort has been devoted to the construction of algorithms that improve on all aspects of the basic finite element method. Most early finite element work concentrated on diffuse scenes; these are called radiosity methods. Later, similar techniques were formulated for glossy, i.e. non-Lambertian, reflectors as well. Finite elements are currently reliving a burst of interest due to their new applications in precomputed light transport techniques, which will be treated later in Section 2.3. In this section we first present the general problems associated with finite element solutions, and then proceed to describe how these problems have been dealt with in different ways.

The original radiosity method of Goral et al. rests essentially on the Galerkin discretization [6, 7, 39] of the diffuse rendering equation using piecewise constant basis functions. This leads to a linear system of equations, where the matrix entries involve so-called *form factors* [36, 24] (more generally *coupling coefficients*) that describe the transport of radiosity from one basis function to all others, and the unknown vector determines the approximate solution in terms of the piecewise constant basis functions.

[1] This more general equation is usually known as the *volume rendering equation* in graphics parlance.

A notable feature of this equation is that the number of matrix entries is n^2 when the basis has n functions. Unfortunately, n may easily range in millions in practical applications [74], and consequently the matrix does not even remotely fit in memory all at once. Furthermore, when using simple basis sets such as piecewise constant functions, the matrix is *full*, which means that a significant fraction of its entries are non-zero. This is due to the non-local nature of the light transport operator. This is in stark contrast to finite elements in the case of the partial differential equations of computational physics, where each row of the matrix typically has a small, almost constant number of non-zero entries, owing to the local nature of differential operators. Finally, each entry in the matrix is a 4D integral that is costly to evaluate, not least because the integrand includes the visibility function. These observations together imply that the naive one-by-one computation of the matrix entries is an insurmountable task in all but the simplest of geometries and crudest of discretizations.

The problems arising from both the large number of form factors in the naive discretization have been attacked on three fronts. Perhaps the simplest way to combat the huge number of form factors is to use *iterative matrix-free techniques* for solving the system. These techniques compute the matrix entries on-the-fly during execution of an iterative linear solver, such as Jacobi or Gauss-Seidel iteration [35]. This general approach was pioneered by Cohen et al. [21], who introduced the concept to graphics, and furthermore described a "shooting" iteration whose visual convergence is faster than that of traditional Gauss-Seidel iteration. The shooting algorithm was later shown to be a combination of Southwell and Jacobi iterations [37]. Matrix-free iterations have remained popular to date. Another avenue of combating the number of form factors is to use higher-order basis functions to capture the variation in the solution using smaller number of more expressive basis functions. Zatz [98] and Troutman and Max [84] were among the first to take this approach.

Perhaps the most important general approach for reducing the complexity of finite element light transport simulations is the use of *hierarchical function bases*. This is best illustrated through an example: The light cast by an unoccluded light source onto a smooth surface some distance away from the light source is a smooth function, and it can thus, intuitively, be represented well by few "large", smooth basis functions.[2] On the other hand, where there are shadow boundaries due to the same light source, the solution varies fast, and hence large, smooth functions will not do a good job in representing the solution.

The basic idea in hierarchical methods based on multiresolution function spaces is to compute the interactions between parts of the scene only to the accuracy that is deemed necessary. This is facilitated by multiresolution function spaces, such as those composed of wavelets [27], where coarser spaces will contain fewer, "larger" functions, and the basis functions on each finer level encode *differences* to the approximation already obtained from the coarser levels. These differences can be intuitively thought of as "corrections". Using such bases leads to *numerical sparsity* in the dis-

[2]Mathematically rigorous treatment of such claims may be found in the books by Daubechies [27] and Mallat [64], for instance.

cretized operator, exactly because the light transport does not always need to be resolved to full resolution; theoretically, the number of matrix entries smaller than a threshold becomes $O(n)$ instead of the usual $O(n^2)$ for a large class of operators [9]. The hierarchical solution algorithm may proceed in the computation from coarse onto the finer levels, stopping where significant corrections are no longer deemed necessary. Cohen et al. [23] utilized a two-level hierarchy in the radiosity setting as early as 1986; they discretized the receivers more coarsely as the senders. However, the first truly hierarchical algorithm with multiple levels was the radiosity technique of Hanrahan et al. [40] who formulated their technique inspired by hierarchical N-body solvers (e.g. [8]). Hierarchies were successfully combined with higher-order basis functions in the wavelet radiosity algorithm by Gortler et al. [38], who imported the technique from the numerics community [9]. Work on higher-order wavelet radiosity methods has continued up to present times [44]. Willmott et al. [97] describe a hierarchical radiosity technique that builds piecewise constant hierarchical function spaces on top of hierarchical clusters of input polygons, rather than utilizing hierarchical 2D function spaces.

Finite element methods have also been applied to the case of glossy reflections. Seemingly the first application of a FEM-like method to glossy reflection was presented by Immel et al. [45], who used piecewise constant basis functions for both spatial and directional variation. Subsequent work includes the glossy radiosity technique of Sillion et al. [78], where the directional distributions are encoded in spherical harmonics, and the hierarchical wavelet techniques of Schröder et al. [77] and Christensen et al. [18, 19]. Unlike their simpler diffuse counterparts, these methods have not, to date, found their way to wide practical use. This stems from the increased need of storage and computation – obviously, solving for and storing a 4D radiance distribution is more costly than a diffuse 2D radiosity distribution. However, new interest in glossy finite element techniques has been sparked by precomputed light transport techniques.

2.3 Precomputed Light Transport

Precomputed light transport techniques aim at interactive relighting of images or scenes. At first, attempting to formulate an interactive global illumination algorithm in an interesting scene seems absurdly difficult; as we have seen, merely computing the solution that corresponds to a single emission entails a significant amount of work and requires sophisticated algorithms. However, it turns out that the following simple observation may be exploited for designing global illumination algorithms that allow changing the illumination interactively while the geometry remains static.

Consider what happens if the emission function E is constrained to lie in some lower-dimensional subspace, which we denote by $\mathbb{E}_h$:

$$E = \sum_{i=1}^{\dim \mathbb{E}_h} e_i \, C_i(x \to \omega_{\text{out}}). \tag{2.1}$$

Here the C_i are the basis of $\mathbb{E}_h$. It immediately follows from the linearity of the rendering equation that

$$\begin{aligned} L &= (\mathcal{I}-\mathcal{T})^{-1}E \\ &= (\mathcal{I}-\mathcal{T})^{-1}\sum_{i=1}^{\dim \mathbb{E}_h} e_i\, C_i = \sum_{i=1}^{\dim \mathbb{E}_h} e_i\, \left((\mathcal{I}-\mathcal{T})^{-1}C_i\right) \\ &= \sum_{i=1}^{\dim \mathbb{E}_h} e_i\, L_i, \quad \text{with } L_i := (\mathcal{I}-\mathcal{T})^{-1}C_i. \end{aligned} \tag{2.2}$$

In words, the lighting solution that results from using a linear combination of the C_i as the emission is just a corresponding linear combination of lighting solutions L_i that have been computed using each C_i separately as the emitter. Once the L_i have been precomputed, a novel lighting solution that corresponds to any emission in $\mathbb{E}_h$ is simple to compute as their combination. This simple observation is the basis of precomputed light transport techniques. It is also possible to formulate precomputed transport methods where the emissions are defined as distant illumination or on some other manifold that is not directly a part of the scene. A more thorough mathematical formalization of these and related techniques is one of the contributions in this thesis [P1]. It will be described in more detail in the next chapter; here we settle for briefly describing previous work. A more thorough review can be found in Publication [P1].

Image relighting techniques render a set of basis images from a fixed viewpoint so that each image corresponds to a single solution L_i from (2.2). Examples of this are the interactive lighting design system of Dorsey et al. [31], and several techniques that produce images of scenes under dynamic distant illumination, e.g., skylight [70, 30, 83, 68] or local light sources [41].

Many precomputed light transport techniques are ultimately based on the finite element method, although they are often not directly expressed as such. These methods represent the basis solutions L_i from Equation (2.2) in terms of a linear function basis. This allows free viewpoints, in contrast to image relighting techniques. In the very simplest form, the radiosity technique of Airey et al. [2] can be seen as a direct implementation of Equation (2.2). The methods described by Sloan et al. [80] for rendering both diffuse and glossy objects under "low-frequency" distant illumination sparked a significant amount of follow-up work that has not ceased to date. They represent a distant illumination function in terms of a few spherical harmonics (SH). Kautz et al. [51] describe an extension to arbitrary BRDFs. Haar wavelets are another popular choice for representing the incident illumination. This approach is taken by Ng et al. [68], Liu et al. [63] and Wang et al. [91, 93]. Spherical radial basis functions (SRBFs) have also been utilized for the task in recent work by Tsai and Shih [85]. Some methods utilize separable BRDF approximation [50] for tailoring an efficient basis for outgoing radiance in the glossy case [63, 91]. Sloan et al. [81] use bidirectional texture functions for representing the final surface interaction. This results in compelling small-scale surface detail. Local precomputed light transport techniques precompute the response of a small surface patch to incident lighting [66, 82], neglecting large-scale shadowing and interreflection effects. These are sometimes called texture relighting methods. Subsurface scattering is also a linear form of light transport, and thus translucent objects can also be relit using techniques that are similar to the above ones [92]. Lensch et al. [62] describe a precomputed transfer technique for rendering translucent meshes with local light sources.

The precomputed solutions L_i easily become a significant storage burden as the dimensionality of $\mathbb{E}_h$ grows. A relatively high dimension is unfortunately necessary, if high-frequency lighting effects, such as spot-like light sources, are required. Compression of the solutions is therefore a must. Two main avenues have been employed so far. First, by representing the illumination solutions in a hierarchical basis (in wavelets, say), the solutions may be compressed simply by thresholding the coefficients. This is known as non-linear approximation [64]. Several authors follow this strategy in their precomputed transport techniques [68, 63, 91, 92]. The second approach is to first precompute the solutions and subsequently run them through a machine learning algorithm whose aim is to reduce the dimensionality of the dataset. These methods range from simple linear principal component analysis (PCA) [83], [P2] to techniques based on clustering the geometry and approximating the solutions within the cluster using some linear method [79, 58]. The main problem associated with these "compute first, compress later" techniques is that the precomputation stage easily requires an unwieldy amount of time and memory.

Kontkanen et al. [55] recently described a precomputed transport method that employs hierarchical techniques for precomputing and representing the whole solution operator that turns direct illumination on the scene into indirect illumination. This differs from "usual" precomputed transport methods by dynamically rendering the direct illumination using traditional methods, e.g., shadow maps, on the graphics processing unit (GPU), and only utilizing precomputation for computing indirect illumination. The method has strong ties to previous hierarchical radiosity techniques. Methods based on this principle avoid many of the compression issues, since the precomputation method is already hierarchical and aims only to compute what is required in the first place. Furthermore, precomputation times are significantly shorter.

As popular as precomputed light transport techniques have recently become, they are not the only avenues in which interactive global illumination has been pursued. Another strategy is to use fast path tracing techniques running on a cluster of PCs, as pioneered by Wald and coworkers [89, 88]. However, the hardware requirements are substantial, and the solutions still exhibit some noise and temporal artifacts.

2.4 Physically-Based Shadow Algorithms and Ambient Occlusion

Visibility techniques that are used for determining the visibility of the light source to the point being shaded when rendering direct illumination are called shadow algorithms.[3] Physically-based shadow algorithms for area

[3]It should however be noted that similar techniques have their uses in other situations as well. For instance, the computation of coupling coefficients in finite element settings requires knowledge of visibility.

and environmental light sources can be roughly categorized into *analytic* methods and techniques based on *sampling*. Analytic techniques determine the actual visible portions of the light source by geometric computations, while sampling techniques discretize the reflectance integral by a number of point samples and evaluate visibility only at these points. Physically-based rendering of soft shadows is a difficult problem, particularly so in deforming scenes. Algorithms designed for area and environmental light sources and dynamic scenes are consequently in relatively short supply.

Nishita and Nakamae [71] describe a soft shadow technique that utilizes silhouette edges and an analytic irradiance formula. The occluders have to be decomposed into convex polyhedra which severely limits the applicability of their technique. Chin and Feiner [15] use two BSP trees to clip the occluded parts of the light source away and also utilize an analytic formula for evaluating the irradiance due to the visible portion. *Beam tracing* [42] constructs a polygonal beam between the point being shaded and the light source. The beam is clipped according to the occluding geometry within the beam, yielding an exact representation of the visible portion of the light source. The algorithm, while elegant, quickly becomes impractical in highly-tessellated scenes. The variant described by Ghazanfarpour and Hasenfratz [34] subdivides the beam in an axis-aligned way until parts of the beam are not blocked, or blocked by a single triangle. This avoids explicit clipping of the beam against blocker geometry, but unfortunately the beam is always subdivided along blocker edges even if they belong to a single, connected surface.

Distributed ray tracing, as proposed by Cook et al. [25], discretizes the integral in Equation (1.16) over the surface of the light source using a number of point samples, and estimates the integral stochastically by evaluating the integrand at these points only. The visibility function is evaluated by a ray tracer. This is, naturally, significantly easier than maintaining a geometric representation of the visible portions of the light source. However, the number of samples used has to be chosen high enough in order to avoid excessive noise. Many variants of this technique are still widely used today.

Distributed ray tracing may also trivially be used for rendering shadows from distant environmental illumination. This, however, requires a significant number of rays per shading event, when high frequencies are present in the environmental illumination. Moreover, highly glossy BRDFs have narrow lobes, meaning that much of the hemisphere contributes negligibly to the final outgoing radiance. Several techniques have been proposed for importance sampling the reflectance integral according to the incident illumination or the product of the incident illumination and the BRDF [1, 61, 12, 20]. The aim of these techniques is to guide the expenditure of the costly shadow rays to locations where they are likely to contribute significantly to the result. However, these techniques are still prone to noise in highly-occluded regions.

The hemicube [22], although originally introduced in the radiosity context for computing form factors, may be easily utilized as a shadow solver for environmental illumination. The hemicube is centered at the point being shaded and oriented according to the local surface normal. Then an image of the scene is rendered on each of the five sides of the hemicube. This results in a visibility map of the hemisphere above the point, which is easily used for rendering shadows from distant illumination through evaluation of Equation (1.17). The paraboloid mapping proposed by Heidrich and Seidel [43] may also be used for this purpose, although rendering on the GPU becomes problematic due to curved edges.

Ambient occlusion and related techniques [100, 60, 16] may be seen as rendering shadows from isotropic environmental illumination by a suitable visibility solver.

Moving and Deforming Objects.

The interactive rendering of physically-based soft shadows from area and distant environmental sources on deforming objects has received relatively little attention, no doubt due to the inherent difficulties involved. In order to give a realistic view of the current situation, we include in the following review also techniques that are strictly speaking not physically-based.

Kontkanen and Laine [54] and Malmer et al. [65] describe techniques for rendering ambient occlusion to the surroundings of moving objects. The techniques produce compelling results at a modest memory and computational cost; however, they are limited to rigid objects. Zhou et al. [99] describe a related technique for rendering shadows cast by moving objects on their surroundings due to distant illumination. Their technique is also limited to rigid objects. James and Fatahalian [46] parameterize a precomputed lighting solution in terms of reduced phase space coordinates that describe the precomputed deformation of the scene. In a somewhat similar vein, Kontkanen and Aila [53] exhaustively sample the pose space of an animated character, precompute an ambient occlusion solution for each pose, and subsequently fit an affine model that describes the ambient occlusion at mesh vertices in terms of the animation parameters in the least squares sense. James and Twigg [47] compress precomputed lighting solutions for fixed animation sequences by PCA. These techniques are purely data-driven, and the set of possible deformations must be known beforehand. Moreover, significant precomputation is required. Sattler et al. [76] present a simple, hardware-accelerated technique for rendering ambient occlusion on deformable models at near-interactive rates using depth maps and occlusion queries. Bunnell [11] describes a real-time method for rendering ambient occlusion on deforming objects. The model is represented by a hierarchy of disks for occlusion computations; however, occluder fusion, i.e., the combined effect of multiple occluders, is accounted for only approximately. Ren et al. [75] describe a related technique for rendering of shadows from low-frequency environmental illumination onto objects that may deform arbitrarily through skeletal subspace deformation (often referred to as *skinning*). They utilize a hierarchy of spheres to represent the model for shadow computations, and use mean value coordinates [48] for animating the sphere hierarchy.

Soft Shadow Volumes for Ray Tracing

Here we describe the physically-based soft shadow algorithm of Laine et al. [59] in slightly more detail, since Publication [P4] improves this technique. The algorithm renders shadows from planar area light sources by

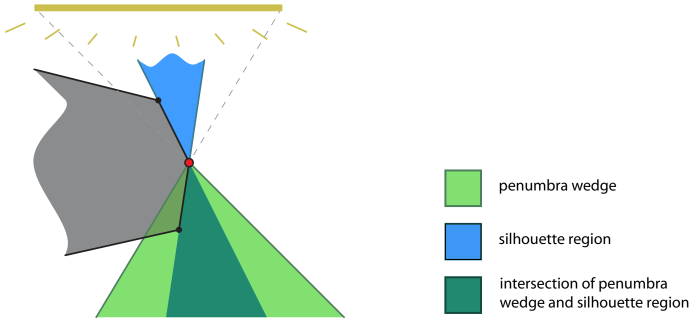


Figure 2.1: A 2D cross section illustrating penumbra wedges and silhouette regions. The penumbra wedge (green) is the volume of space from which the generating edge (red dot) projects onto the light source. When looked at from points in the silhouette region (blue), the generating edge is a silhouette. In order for an edge to be relevant for the integration of depth complexity for a given receiver point, the receiver must lie *both* in the penumbra wedge and the silhouette region.

evaluating Equation (1.16) for one receiver point x at a time through Monte Carlo integration. The result is exactly the same as a stochastic ray tracer would produce using the same sample set; only the method for determining the visibilities of the shadow rays is different. The technique achieves impressive speedups compared to a ray tracer in many situations. This is because the amount of work performed by the algorithm is dependent on the complexity of silhouettes, not the amount of blocker geometry.

Consider the computation of the visibility to a single, planar area light source from a fixed point x. Once the sample points $\{l_i\}_{i=1}^{n}$ on the light source (the *light samples*) have been chosen, the algorithm computes the visibility from x to each of the l_i utilizing coherence between the light samples. This is achieved by reinterpreting the visibility function through *depth complexity* $D(x, l)$, which is defined as the number of occluding surfaces a ray from x to l pierces on its way. Clearly, $V(x, l) = 1$ if and only if $D(x, l) = 0$. It is clear that the depth complexity only changes when there is a *silhouette edge* that belongs to an occluder somewhere inside the generalized cone defined by x and the light source. Hence the silhouette edges may be interpreted as the derivative of the depth complexity function. After all the relevant silhouette edges have been found, it is a simple matter to integrate the derivatives to yield a *relative depth complexity* for each of the light samples. Once this has been completed, a single reference shadow ray is sufficient to determine whether or not the samples that share the lowest relative depth complexity are visible or not. The depth complexity is indeed relative, since the light source may be completely blocked even when there are no silhouette edges between x and the light.

In order to be relevant for the integration of depth complexity for the light samples, an edge e of a blocker must fulfill two criteria: 1) It must reside at least partially within the generalized cone formed by x and the light source; and 2) e must be a silhouette as seen from x. *Penumbra wedges* [5] are semi-infinite convex volumes associated with each potential silhoutte edge. Figure 2.1 illustrates the geometric situation. They are constructed in such a way that whenever a receiver point lies within the wedge, the edge fulfills criterion 1. In order to avoid testing all penumbra wedges against each point being shaded, the algorithm uses a hemicube-like data structure that is wrapped around the scene to quickly determine a conservative set of potential wedges. The returned set is then further pruned by a geometric point-in-wedge test. Furthermore, a silhouette test must still be performed on the remaining edges to see if they satisfy criterion 2.

3 NOVEL THEORY AND TECHNIQUES

The chapter describes the contributions made in the publications [P1]–[P4] in Sections 3.1–3.4 and presents some concluding remarks in Section 3.5. The author's role in each of the publications is summarized in Chapter 4.

3.1 A Framework for Precomputed and Captured Light Transport

Publication [P1] presents a mathematical framework for methods that precompute light transport in virtual scenes; these include a substantial number of precomputed radiance transfer and image relighting methods. Furthermore, techniques that capture light transport operators from real scenes for relighting purposes may be described using the same formalism. Such techniques include (but are not limited to) bidirectional texture functions (BTFs) [26] and reflectance fields (e.g. [67]). The article builds on well-established mathematical tools and gives the framework in the form of an operator equation that extends the rendering equation. After the equation has been derived, a large body of earlier work is characterized in its terms.

The key idea of the formulation is as follows. Instead of having the emission function E belong to the same space of radiance functions as the sought-after lighting solution L, the emission lies in a function space defined on some other domain than that formed by the surfaces of the scene. This function space is called the *emission space* $\mathbb{E}$. Perhaps the simplest example is distant illumination that lies in the space of spherical functions. Then, a linear mapping $\mathcal{V}$ from E onto the space of incident radiance functions on the scene is introduced. This mapping describes how a function in $\mathbb{E}$ is turned into an incident lighting distribution on the scene. It is now a simple matter to write down the incident rendering equation with the incident direct illumination provided by $\mathcal{V}$ from the emission space:

$$(\mathcal{I} - \mathcal{T}_{\mathrm{i}})L_{\mathrm{i}} = \mathcal{V}E \;\Leftrightarrow\; L_{\mathrm{i}} = (\mathcal{I} - \mathcal{T}_{\mathrm{i}})^{-1}\mathcal{V}E, \quad E \in \mathbb{E}. \tag{3.1}$$

Here L_{i} is the incident global illumination solution defined on the surfaces of the scene that results from using $E \in \mathbb{E}$ as the emission. To get an outgoing radiance solution, the local reflection operator $\mathcal{R}$ is applied, followed by a set of linear measurements of the solution. The measurements are incorporated in the operator $\mathcal{M}$. This results in

$$\boldsymbol{m} = \mathcal{M}\mathcal{R}(\mathcal{I} - \mathcal{T}_{\mathrm{i}})^{-1}\mathcal{V}E, \quad E \in \mathbb{E}. \tag{3.2}$$

The result $\boldsymbol{m}$ is a vector of scalar measurements of the lighting solution. The measurements may include, for instance, computing suitable averages of the lighting at locations defined by the pixels in an image – this would be an image relighting method – or applying a projection operator for approximating the lighting solution in a function basis. Equation (3.2) is a linear operator equation much like the rendering equation, and hence similar numerical techniques may be employed for its solution. If $\mathbb{E}$ is finite-dimensional, the operator $\mathcal{M}\mathcal{R}(\mathcal{I} - \mathcal{T}_{\mathrm{i}})^{-1}\mathcal{V}$ is actually a finite matrix. The "transfer matrices" from precomputed radiance transfer techniques may be seen as subsets of this operator after a particular discretization has been applied.

3.2 Matrix Radiance Transfer

Publication [P2] treats the interactive rendering of non-deforming objects in dynamic low-frequency lighting environments, including smooth shadows and indirect illumination. The method is a derivative of the precomputed radiance transfer methods of Sloan et al. [80] and Kautz et al. [51], whose runtime efficiency is improved by utilizing a function basis for outgoing radiance that is more efficient to evaluate than the spherical harmonics.

In the original methods of Sloan et al. [80] and Kautz et al. [51], a significant computational cost is incurred by the multiplication of the spherical harmonic coefficients that describe the incident illumination by so-called *transfer matrices*. By definition, the resulting transferred radiance includes the effect of self-shadows and interreflections caused by the object on itself. Since transferred radiance is an incident quantity, it must be reflected once more to yield outgoing radiance that can be assigned to pixels. Both the above methods utilize the spherical harmonics for these computations. Since the spherical harmonics have global supports over the sphere, all the coefficients for transferred radiance need to be computed before the reflection step in both techniques.

The technique described in Publication [P2] casts the full pipeline from the distant illumination to outgoing radiance into a single, large matrix expression. Contrary to the previous techniques, outgoing radiance is projected into a basis of piecewise bilinear functions that have local supports over the hemisphere. As a result, only *four* coefficients are required per vertex to determine outgoing radiance, and hence, only four rows of the matrix-vector product that determines these coefficients from the incident illumination need to be computed per frame. This is a significant saving compared to the previous techniques that use spherical harmonics for the outgoing radiance computations. The improved efficiency comes at the cost of increased memory usage, since the dimension of the new basis has to be higher than the SH dimension to yield comparable quality. This, in turn, leads to larger transfer matrices.

In addition, a simple compression scheme based on principal component analysis (PCA) is described to reduce the data size of the transfer matrices. The compressed representation allows direct rendering without an intermediate decompression step. Admittedly, subsequent, more advanced compression schemes, such as the clustered principal components (CPCA) described by Sloan et al. [79] have significant advantages over the technique described in Publication [P2].

3.3 Self-Shadowing of Dynamic Objects

Publication [P3] describes an efficient software technique for rendering physically-based soft self-shadows on deforming objects due to distant, low-frequency environmental illumination. Both diffuse and mildly glossy

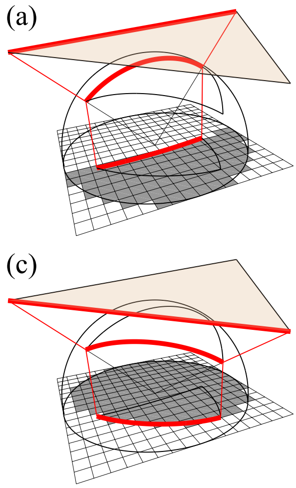


Figure 3.1: An illustration of hemispherical rasterization using bit masks. Each edge of the triangle defines a plane together with the center of projection. Bit masks that have a set bit for each pixel above the halfspace that corresponds to the plane are fetched for each edge (a)-(c). Their logical intersection (d) is the image of the triangle under the hemispherical projection.

BRDFs are supported, but indirect illumination is not accounted for. The most natural usage scenario is rendering shadows on a character moving about in a virtual environment, so that the lights and shadows cast by other objects onto the character are represented by the incident environmental illumination function. The shadowed illumination is computed at the vertices of the model on the CPU and interpolated to the pixels on the GPU. The technique does not require a priori knowledge of the animation, which is a major advantage compared to many other techniques for rendering soft shadows on deforming objects. Here indirect illumination has been forfeited in order to be able to animate the object freely.

The shadowed illumination is rendered by evaluating a discretized version of Equation (1.17), where the distant illumination $E(\omega_{\text{in}})$ is represented as a spherical harmonic expansion. The visibility term is first determined by rasterizing the object in its current pose into a 32×32 *hemispherical visibility buffer*. Once the visibility has been computed, it is downsampled in blocks of 4×4 pixels, and subsequently the reflectance integral is approximated by a sum over the downsampled buffer. When rasterizing the object, a two-level level-of-detail (LOD) scheme is used: Triangles nearby to the point being shaded are rendered from the full-resolution display mesh, while triangles further away are rendered from a simplified mesh.

The hemispherical projection used for rendering the blocker triangles is as follows: Each triangle is projected onto the unit hemisphere centered at the receiving point, and then flattened onto the unit disk by dropping the

z coordinate. This parameterization implements the "Nusselt analog" [24], and it is thus a perfect importance sampling of the hemisphere according to the local cosine term.

The projection and rasterization of blockers into a visibility buffer has obvious similarities to the hemicube [22]. However, the hemispherical method allows the rendering of the whole hemisphere by a single pass over the blocking geometry, in contrast to the five passes required by the hemicube. Furthermore, the distribution of pixels in the hemicube does not importance sample the hemisphere in a good manner, unlike the present parameterization. While the paraboloid parameterization of Heidrich and Seidel [43] also only requires one pass for rendering the whole hemisphere, the distribution of its pixels is also not ideal.

The edges of blocker triangles become curved under the hemispherical projection. To cope with this complication, a form of precomputation is employed. The set of points inside any spherical triangle may be defined as the union of three binary spherical functions, each of which has value 1 in the hemisphere that corresponds to the halfspace defined by the center-of-projection and one edge of the blocker triangle, and 0 in the other halfspace. The rasterized images of such binary functions are precomputed and stored as bit masks for a number of possible halfspaces, and the runtime image of a triangle is found simply by fetching these bit masks, and ANDing them together. This is illustrated in Figure 3.1. The result is finally ORed to the visibility buffer, yielding correct occluder fusion.

3.4 An Improved Soft Shadow Volume Algorithm

The physically-based soft shadow volume algorithm described by Laine et al. [59] is analyzed and improved in Publication [P4]. The original technique sometimes suffers from unpredictable performance due to the overly conservative spatial acceleration structure that is used for finding the penumbra wedges [5, 59] that may contain the point currently being shaded. At times, performance may degrade even below that of a reference ray tracer. These cases are analyzed, and an improved acceleration structure that alleviates the problems is presented.

Publication [P4] identifies several major shortcomings in the hemicube-like acceleration structure used in the original algorithm. These affect only the performance, not the correctness of the algorithm. First, the size of the hemicube is dependent on the orientation of the light source with respect to the scene. Increased hemicube size has the effect that the footprints of the wedges grow, with the result that the wedges will be reported as potential candidates for a larger fraction of receiver points. This adversely affects performance. Second, since the wedges are flattened onto the sides of the hemicube, all information on the 3D location of the wedge is lost (see Figure 3.2a), and again the wedge will be unnecessarily reported for many receiver points. Finally, the hemicube is unable to use the silhouette regions for culling the wedges due to its reliance on a 2D projection. This is illustrated in Figure 3.2b.

To remedy the above performance problems, a novel 3D hierarchical wedge query structure is introduced. In essence, the wedges are conserva-

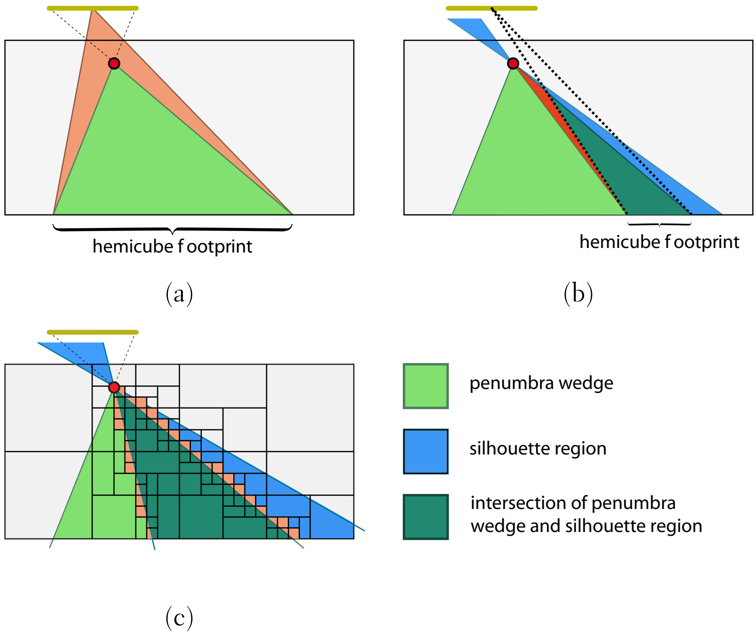


Figure 3.2: 2D cross sections that illustrate the hemicube and the 3D query structure. (a) In addition to the points that belong to the penumbra wedge (green), the wedge is reported by the hemicube structure for the points inside the orange polygon that do not belong to the penumbra wedge. (b) The points inside the red polygon lie inside both the penumbra wedge and the silhouette region (blue), but when projected onto the hemicube from the center of the light source, their image does not overlap the footprint of the combined wedge and silhouette region. Consequently, the silhouette region cannot be easily utilized by the original method. (c) An illustration of the 3D wedge query structure. The intersection of the penumbra wedge and the silhouette region is conservatively "rasterized" into a 3D hierarchical grid. The orange regions lie outside the intersection, but are reported nonetheless. Comparison to the corresponding orange areas in (a) reveals that the hemicube is much more conservative.

tively "rasterized" into a 3D hierarchical grid so that a single walk down the tree and gathering of wedges from the nodes passed during the traverse yields the potential wedge set. See Figure 3.2c. Since the structure is 3D and no projections are required, both the 2D-related problems of the hemicube are avoided. As another major improvement, a conservative silhouette test can now easily be incorporated into the spatial structure, which leads to significantly tighter potential wedge sets and hence better performance particularly in scenes with small silhouette regions, i.e., highly-tessellated scenes with smooth, curved geometry. The construction of the data structure is performed lazily only in places where queries are made in order to avoid a long startup time and unnecessary memory consumption.

The novel method outperformed the original technique in all the cases it was tested in during the preparation of Publication [P4], including all the test runs whose results were not included in the publication. The savings are significant particularly in cases where the above-mentioned problems are acute, while gains are marginal in cases where the above-mentioned problems do not manifest themselves. The new query structure achieves higher performance since it prunes the candidate wedge sets much more efficiently than the original technique. The improved algorithm is substantially more predictable, since issues such as the orientation of the light source do not affect performance significantly. The benefits come at the cost of higher memory consumption due to the increased dimensionality of the query structure.

3.5 Concluding Remarks

The techniques presented in this thesis approach the problem of realistic image synthesis from several different directions. Two techniques are aimed at interactive scenarios: Precomputed transfer [P2] fixes the scene in order to gain relighting ability, while the environmental shadow algorithm [P3] neglects indirect illumination in order to be able to animate the object freely. The off-line soft shadow technique [P4], on the other hand, requires pre-processing and knowledge of the whole scene in advance, but after preprocessing, it capitalizes on coherence between nearby shadow rays. These techniques illustrate the need for specialized algorithms for different usage scenarios.

It is the author's hope that the operator equation described in Publication [P1] ties precomputed transfer more tightly to traditional global illumination in the eyes of the research community, and thus facilitates the application of the whole prior global illumination arsenal for creating novel precomputed transport methods.

The recent work by Ren et al. [75] on soft shadow computation for animated characters has to be considered in many aspects more advanced than the method described in [P3]. They effectively use a deeper hierarchy for representing the blockers. Furthermore, since the visibility is accumulated in spherical harmonics, the shading integral reduces to a dot product, removing the need to iterate over the pixels. However, graphics hardware has undergone significant changes since Publication [P3] was published. New DirectX^{TM}10 [10] class hardware features integer instructions in shader units, and consequently it should be relatively straightforward to implement the algorithm on a GPU. This would be a worthwhile exercise, since the execution model of the hemispherical rasterizer is highly amenable to massively parallel implementation: No heavy serial per-frame operations are necessary before the rendering of illumination may begin, since no complex spatial hierarchies that need to be maintained by the host CPU are required.

Publication [P1]
This article presents a novel abstract operator equation that can be used for describing a large class of techniques that precompute light transport operators for relighting purposes or capture such operators from real scenes. The operator equation is an extension of the well-known rendering equation. The text describes how different discretizations of the operator equation lead to particular previous methods, and describes a large body of earlier work in terms of the framework. Furthermore, directions of future work are outlined based on insights drawn from the abstract formulation.

The author is the sole author of this publication.

Publication [P2]
The technique described in this paper enables rendering of global illumination effects from smooth, low-frequency lighting environments more efficiently than previous similar techniques. Added efficiency is achieved by applying a change of basis and rendering the outgoing radiance in a directional basis of functions with compact supports. This significantly reduces runtime complexity by removing the need for large matrix-vector multiplications. Furthermore, a simple compression scheme based on principal component analysis (PCA) is described.

The author invented the idea of using a directional basis with compact supports to gain runtime efficiency, formulated the associated mathematics, implemented the technique, and wrote 70% of the paper. The compression scheme was designed by Dr. Kautz and implemented by the author.

Publication [P3]
A software rasterizer capable of rendering triangles under a hemispherical projection is described. The complications arising from the curved edges produced by the projection are solved with simple look-ups and bit operations. The rasterizer is applied to interactive rendering of shadows due to distant, low-frequency environmental illumination on animated objects, enabling the rendering of difficult lighting effects without resorting to unreasonably cumbersome precomputation. The technique requires no a priori knowledge of the animation.

The research topic of rendering environmental shadows for animated objects originated from Dr. Kautz. The author invented the hemispherical rasterizer and the method of parameterizing spherical triangles by unions of half-spaces, enabling the rasterization of spherical triangles by simple look-ups and bit operations. The author implemented the hemispherical rasterizer and wrote 50% of the paper. The rest of the paper was written by Dr. Kautz and Dr. Aila.

Publication [P4]

This article analyzes the geometric causes of certain performance problems in a previous physically-based soft shadow volume algorithm. Once the problems are identified and analyzed, a novel spatial acceleration structure is introduced that largely alleviates the problems. Comprehensive measurements are reported to validate the superiority of the enhanced method in several cases.

The author and Dr. Laine formulated the novel acceleration structure and the related algorithms in equal proportions. The realization that silhouette edges may be exploited by the novel structure was made together by the author and Dr. Laine. The identification of the problems of the previous technique was carried out mainly by Dr. Laine, who also wrote 10% of the paper. The author implemented the technique, performed the measurements, and wrote 80% of the paper. The rest of the paper was written by Dr. Aila.

BIBLIOGRAPHY

[1] Sameer Agarwal, Ravi Ramamoorthi, Serge Belongie, and Henrik Wann Jensen. Structured importance sampling of environment maps. *ACM Trans. Graph.*, 22(3):605–612, July 2003.

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

[3] James Arvo. The Role of Functional Analysis in Global Illumination. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95*. Springer-Verlag, 1995.

[4] James Arvo, Kenneth Torrance, and Brian Smits. A framework for the analysis of error in global illumination algorithms. In *Proceedings of ACM SIGGRAPH 94*, pages 75–84. ACM Press, 1994.

[5] Ulf Assarsson and Tomas Akenine-Möller. A geometry-based soft shadow volume algorithm using graphics hardware. *ACM Trans. Graph.*, 22(3):511–520, July 2003.

[6] Kendall Atkinson. *The Numerical Solution of Integral Equations of the Second Kind*. Cambridge University Press, 1997.

[7] Kendall Atkinson and Weimin Han. *Theoretical Numerical Analysis*. Springer-Verlag, 2001.

[8] J. Barnes and P. Hut. A Hierarchical O(N log N) force-calculation algorithm. *Nature*, 324(4), December 1986.

[9] G. Beylkin, R. Coifman, and V. Rokhlin. Fast Wavelet Transforms and Numerical Algorithms I. *Commun. Pure Appl. Math.*, 44:141–183, 1991.

[10] David Blythe. The Direct3D 10 system. *ACM Trans. Graph.*, 25(3):724–734, July 2006.

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

[12] David Burke, Abhijeet Ghosh, and Wolfgang Heidrich. Bidirectional importance sampling for direct illumination. In *Rendering Techniques 2005 (Proc. 16th Eurographics Symposium on Rendering)*, pages 147–156, June 2005.

[13] Subrahmanyan Chandrasekhar. *Radiative Transfer*. Dover, 1960.

[14] Shenchang Eric Chen, Holly E. Rushmeier, Gavin Miller, and Douglass Turner. A progressive multi-pass method for global illumination. In *Computer Graphics (Proceedings of ACM SIGGRAPH 91)*, pages 165–174. ACM Press, 1991.

[15] Norman Chin and Steven Feiner. Fast Object-Precision Shadow Generation for Area Light Source using BSP Trees. In *Proc. Symposium on Interactive 3D Graphics*, pages 21–30, 1992.

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

[17] Per H. Christensen. Adjoints and importance in rendering: An overview. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):329–340, Jul-Sep 2003.

[18] Per H. Christensen, Eric J. Stollnitz, David H. Salesin, and Tony D. DeRose. Wavelet radiance. In *Photorealistic Rendering Techniques (Proc. Fifth Eurographics Workshop on Rendering)*. Springer-Verlag, 1994.

[19] Per H. Christensen, Eric J. Stollnitz, David H. Salesin, and Tony D. DeRose. Global illumination of glossy environments using wavelets and importance. *ACM Trans. Graph.*, 15(1):37–71, 1996.

[20] Petrik Clarberg, Wojciech Jarosz, Tomas Akenine-Möller, and Henrik Wann Jensen. Wavelet Importance Sampling: Efficiently Evaluating Products of Complex Functions. *ACM Trans. Graph.*, 24(3), 2005.

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

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

[23] Michael F. Cohen, Donald P. Greenberg, David S. Immel, and Philip J. Brock. An efficient radiosity approach for realistic image synthesis. *IEEE Computer Graphics and Applications*, 6(3), 1986.

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

[25] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. In *Computer Graphics (Proceedings of SIGGRAPH 84)*, pages 137–145, July 1984.

[26] Kristin J. Dana, Bram van Ginneken, Shree K. Nayar, and Jan J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Trans. Graph.*, 18(1):1–34, 1999.

[27] Ingrid Daubechies. *Ten Lectures on Wavelets*. SIAM, 1993.

[28] Paul Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of SIGGRAPH 98*, pages 189–198. ACM Press, 1998.

[29] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of ACM SIGGRAPH 97*, pages 369–378. ACM Press, 1997.

[30] Yoshinori Dobashi, Kazufumi Kaneda, Hideo Yamashita, and Tomoyuki Nishita. Method for calculation of sky light luminance aiming at an interactive architectural design. *Computer Graphics Forum*, 15(3):109–118, 1996.

[31] Julie O'B. Dorsey, James Arvo, and Donald Greenberg. Interactive Design of Complex Time-Dependent Lighting. *IEEE Computer Graphics and Applications*, 15(2):26–36, 1995.

[32] Frédo Durand, Nicolas Holzschuch, Cyril Soler, Eric Chan, and François Sillion. A frequency analysis of light transport. *ACM Trans. Graph.*, 24(3):1115–1126, 2005.

[33] Philip Dutré, Philippe Bekaert, and Kavita Bala. *Advanced Global Illumination*. AK Peters, 2003.

[34] Djamchid Ghazanfarpour and Jean-Marc Hasenfratz. A Beam Tracing with Precise Antialiasing for Polyhedral Scenes. *Computers & Graphics*, 22(1):103–115, 1998.

[35] Gene H. Golub and Charles F. Van Loan. *Matrix Computations, 3rd edition*. Johns Hopkins University Press, 1996.

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

[37] Steven Gortler, Michael F. Cohen, and Philipp Slusallek. Radiosity and relaxation methods. *IEEE Trans. Vis. Comput. Graph.*, 14(6):48–58, 1993.

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

[39] Wolfgang Hackbusch. *Integral Equations – Theory and Numerical Treatment*. Birkhäuser Verlag, 1995.

[40] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. In *Computer Graphics (Proceedings of ACM SIGGRAPH 91)*, pages 197–206. ACM Press, 1991.

[41] Miloš Hašan, Fabio Pellacini, and Kavita Bala. Direct-to-indirect transfer for cinematic relighting. *ACM Trans. Graph.*, 25(3):1089–1097, 2006.

[42] Paul S. Heckbert and Pat Hanrahan. Beam tracing polygonal objects. In *Computer Graphics (Proceedings of SIGGRAPH '84)*, pages 119–127. ACM Press, 1984.

[43] Wolfgang Heidrich and Hans-Peter Seidel. View-independent environment maps. In *Proc. Graphics Hardware '98*, pages 39–45, 1998.

[44] Nicolas Holzschuch and Laurent Alonso. Combining Higher-Order Wavelets and Discontinuity Meshing: a Compact Representation for Radiosity. In *Rendering Techniques 2004 (Proceedings of the Eurographics Symposium on Rendering 2004)*, pages 275–286. The Eurographics Association, 2004.

[45] David S. Immel, Michael F. Cohen, and Donald P. Greenberg. A radiosity method for non-diffuse environments. In *Computer Graphics (Proceedings of ACM SIGGRAPH 86)*, pages 133–142. ACM Press, 1986.

[46] Doug James and Kayvon Fatahalian. Precomputing Interactive Dynamic Deformable Scenes. *ACM Trans. Graph.*, 22(3):879–887, July 2003.

[47] Doug L. James and Christopher D. Twigg. Skinning mesh animations. *ACM Trans. Graph.*, pages 399–407, 2005.

[48] Tao Ju, Scott Schaefer, and Joe Warren. Mean value coordinates for closed triangular meshes. *ACM Trans. Graph.*, 24(3):561–566, 2005.

[49] James T. Kajiya. The Rendering Equation. In *Computer Graphics (Proceedings of ACM SIGGRAPH 86)*, pages 143–150. ACM Press, 1986.

[50] Jan Kautz and Michael D. McCool. Interactive Rendering with Arbitrary BRDFs using Separable Approximations. In *Proceedings of the 10th Eurographics Workshop on Rendering*, pages 281–292, June 1999.

[51] Jan Kautz, Peter-Pike Sloan, and John Snyder. Fast, Arbitrary BRDF Shading for Low-Frequency Lighting Using Spherical Harmonics. In *13th Eurographics Workshop on Rendering*, pages 301–308, June 2002.

[52] Douglas S. Kay and Donald P. Greenberg. Transparency for computer synthesized images. In *Computer Graphics (SIGGRAPH '79 Proceedings)*, pages 158–164, August 1979.

[53] Janne Kontkanen and Timo Aila. Ambient occlusion for animated characters. In *Rendering Techniques 2006 (Proc. Eurographics Symposium on Rendering)*. Eurographics Association, jun 2006.

[54] Janne Kontkanen and Samuli Laine. Ambient occlusion fields. In *Proceedings of ACM SIGGRAPH 2005 Symposium on Interactive 3D Graphics and Games*, pages 41–48. ACM Press, 2005.

[55] Janne Kontkanen, Emmanuel Turquin, Nicolas Holzschuch, and François Sillion. Wavelet radiance transport for interactive indirect lighting. In *Rendering Techniques 2006 (Proc. Eurographics Symposium on Rendering)*. Eurographics Association, jun 2006.

[56] Rainer Kress. *Linear integral equations, 2nd ed.* Springer, 1999.

[57] Erwin Kreyszig. *Introductory Functional Analysis*. John Wiley & Sons, 1989.

[58] Anders Wang Kristensen, Tomas Akenine-Möller, and Henrik Wann Jensen. Precomputed local radiance transfer for real-time lighting design. *ACM Trans. Graph.*, 24(3):1208–1215, 2005.

[59] Samuli Laine, Timo Aila, Ulf Assarsson, Jaakko Lehtinen, and Tomas Akenine-Möller. Soft shadow volumes for ray tracing. *ACM Trans. Graph.*, 24(3):1156–1165, 2005.

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

[61] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Adaptive numerical cumulative distribution functions for efficient importance sampling. In *Rendering Techniques 2005 (Proc. 16th Eurographics Workshop on Rendering)*, pages 11–20, June 2005.

[62] Hendrik P. A. Lensch, Michael Goesele, Philippe Bekaert, Jan Kautz, Marcus A. Magnor, Jochen Lang, and Hans-Peter Seidel. Interactive rendering of translucent objects. *Computer Graphics Forum*, 22(2):195–206, June 2003.

[63] Xinguo Liu, Peter-Pike Sloan, Heung-Yeung Shum, and John Snyder. All-Frequency Precomputed Radiance Transfer for Glossy Objects. In *Rendering Techniques 2004 (Proceedings of the Eurographics Symposium on Rendering 2004)*, pages 337–344. The Eurographics Association, 2004.

[64] Stéphane Mallat. *A Wavelet Tour of Signal Processing, 2nd ed.* Academic Press, 1999.

[65] Mattias Malmer, Fredrik Malmer, Ulf Assarson, and Nicolas Holzschuch. Fast precomputed ambient occlusion for proximity shadows. *Journal of Graphics Tools*, to appear.

[66] Tom Malzbender, Dan Gelb, and Hans Wolters. Polynomial texture maps. In *Proceedings of ACM SIGGRAPH 2001*, pages 519–528, August 2001.

[67] Wojciech Matusik, Hanspeter Pfister, Remo Ziegler, Addy Ngan, and Leonard McMillan. Acquisition and rendering of transparent and refractive objects. In *Rendering Techniques 2002 (Proc. 13th Eurographics Workshop on Rendering)*, pages 267–278, June 2002.

[68] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. All-Frequency Shadows Using Non-linear Wavelet Lighting Approximation. *ACM Trans. Graph.*, 22(3):376–381, 2003.

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

[70] Jeffry S. Nimeroff, Eero Simoncelli, and Julie Dorsey. Efficient Re-rendering of Naturally Illuminated Environments. In *Proceedings of the Fifth Eurographics Workshop on Rendering*, pages 359–373, Darmstadt, Germany, 1994.

[71] Tomoyuki Nishita and Eihachiro Nakamae. Half-Tone Representation of 3-D Objects Illuminated by Area Sources or Polyhedron Sources. In *IEEE Computer Software and Application Conference*, pages 237–242, 1983.

[72] Matt Pharr and Greg Humphreys. *Physically Based Rendering*. Morgan Kaufmann, 2004.

[73] Paul Rademacher, Jed Lengyel, Ed Cutrell, and Turner Whitted. Measuring the perception of visual realism in images. In *Rendering Techniques 2001 (Proc. 12th Eurographics Workshop on Rendering)*, pages 235–248, June 2001.

[74] Remedy Entertainment. Max Payne 2. Computer game. Rockstar Games, 2003.

[75] Zhong Ren, Rui Wang, John Snyder, Kun Zhou, Xinguo Liu, Bo Sun, Peter-Pike Sloan, Hujun Bao, Qunsheng Peng, and Baining Guo. Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. *ACM Trans. Graph.*, 25(3):977–986, 2006.

[76] Mirko Sattler, Ralf Sarlette, Gabriel Zachmann, and Reinhard Klein. Hardware-accelerated ambient occlusion computation. In *Proc. Vision, Modeling, and Visualization 2004*, pages 331–338. Akademische Verlagsgesellschaft Aka GmbH, Berlin, November 2004.

[77] Peter Schröder and Pat Hanrahan. Wavelet methods for radiance computations. In *Photorealistic Rendering Techniques (Proc. Fifth Eurographics Workshop on Rendering)*. Springer-Verlag, 1994.

[78] François X. Sillion, James R. Arvo, Stephen H. Westin, and Donald P. Greenberg. A global illumination solution for general reflectance distributions. In *Computer Graphics (Proceedings of ACM SIGGRAPH 91)*, pages 187–196. ACM Press, 1991.

[79] Peter-Pike Sloan, Jesse Hall, John Hart, and John Snyder. Clustered Principal Components for Precomputed Radiance Transfer. *ACM Trans. Graph.*, 22(3):382–391, 2003.

[80] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph.*, 21(3):527–536, 2002.

[81] Peter-Pike Sloan, Xinguo Liu, Heung-Yeung Shum, and John Snyder. Bi-Scale Radiance Transfer. *ACM Trans. Graph.*, 22(3):370–375, July 2003.

[82] Peter-Pike Sloan, Ben Luna, and John Snyder. Local, deformable precomputed radiance transfer. *ACM Trans. Graph.*, 24(3):1216–1224, 2005.

[83] Patrick Teo, Eero Simoncelli, and David Heeger. Efficient Linear Re-rendering for Interactive Lighting Design. Technical Report CS-TN-97-60, Stanford University, 1997.

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

[85] Yu-Ting Tsai and Zen-Chung Shih. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Trans. Graph.*, 25(3):967–976, 2006.

[86] Gennadi Vainikko. Integral equations. Lecture notes, Helsinki University of Technology, 2003.

[87] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997.

[88] Ingo Wald, Carsten Benthin, and Philipp Slusallek. Interactive global illumination in complex and highly occluded environments. In *Rendering Techniques 2003 (Proc. Eurographics Symposium on Rendering)*, pages 74–81. Eurographics Association, 2003.

[89] Ingo Wald, Thomas Kollig, Carsten Benthin, Alexander Keller, and Philipp Slusallek. Interactive global illumination using fast ray tracing. In *Rendering Techniques 2002 (Proc. Eurographics Workshop on Rendering)*, pages 15–24. Eurographics Association, 2002.

[90] John R. Wallace, Michael F. Cohen, and Donald P. Greenberg. A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. In *Computer Graphics (Proceedings of ACM SIGGRAPH 87)*, pages 311–320. ACM Press, 1987.

[91] Rui Wang, John Tran, and David Luebke. All-Frequency Relighting of Non-Diffuse Objects using Separable BRDF Approximation. In *Rendering Techniques 2004 (Proceedings of the Eurographics Symposium on Rendering 2004)*, pages 345–354. The Eurographics Association, 2004.

[92] Rui Wang, John Tran, and David Luebke. All-frequency interactive relighting of translucent objects with single and multiple scattering. *ACM Trans. Graph.*, 24(3):1202–1207, 2005.

[93] Rui Wang, John Tran, and David Luebke. All-frequency relighting of glossy objects. *ACM Trans. Graph.*, 25(2):293–318, 2006.

[94] Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping*. AK Peters, 2001.

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

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

[97] Andrew Willmott, Paul Heckbert, and Michael Garland. Face cluster radiosity. In *Rendering Techniques (Proc. 10th Eurographics Workshop on Rendering)*, 1999.

[98] 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.

[99] Kun Zhou, Yaohua Hu, Stephen Lin, Baining Guo, and Heung-Yeung Shum. Precomputed Shadow Fields for Dynamic Scenes. *ACM Trans. Graph.*, 24(3):1196–1201, 2005.

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