

Efficient Rendering of Translucent Objects Using the Diffusion Approximation

Proefschrift voorgelegd tot het behalen van de graad van
Doctor in de Wetenschappen, richting Informatica,
te verdedigen door

Tom MERTENS

Promotor : Prof. dr. P. Bekaert

2005

D O C T O R A A T S P R O E F S C H R I F T



Universiteit Maastricht



PARTNER IN DE UNIVERSITEIT LIMBURG

- 681.39

- Computer graphics

BIBLIOTHEEK UNIVERSITEIT HASSELT



03 04 0084045 7

051153



27 JUN 2005



681.39
MERT
2005

uhasselt

School voor Informatietechnologie
Kennistechnologie, Informatica, Wiskunde, ICT

Efficient Rendering of Translucent Objects Using the Diffusion Approximation

Proefschrift voorgelegd tot het behalen van de graad van
Doctor in de Wetenschappen, richting Informatica,
te verdedigen door

Tom MERTENS

Promotor : Prof. dr. P. Bekaert

2005



27 JUN 2005

According to the guidelines of the Limburgs Universitair Centrum, a copy of this publicatie has been filed in the Royal Library Albert I, Brussels, as publication D/2005/2451/3

transnationale Universiteit Limburg
School voor Informatietechnologie

Efficiënte weergave van translucente objecten
m.b.v. de diffusiebenadering

Proefschrift voorgelegd tot het behalen van de graad van
Doctor in de Wetenschappen, richting Informatica
aan de transnationale Universiteit Limburg
te verdedigen door

Tom Mertens

op 4 juli 2005

Jury

Prof. dr. Marc Geysens, voorzitter, transnationale Universiteit Limburg
Prof. dr. Philippe Bekaert, promotor, transnationale Universiteit Limburg
Prof. dr. Frank Van Reeth, co-promotor, transnationale Universiteit Limburg
Prof. dr. Eddy Flerackers, transnationale Universiteit Limburg
Prof. dr. Frédo Durand, Massachusetts Institute of Technology
Prof. dr. Philip Dutré, Katholieke Universiteit Leuven
Prof. dr. Hans-Peter Seidel, Max Planck Institut Für Informatik

transnational University of Limburg
School of Information Technology

Efficient Rendering of Translucent Objects
Using the Diffusion Approximation

Dissertation submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science
at the transnational University of Limburg,
to be defended by

Tom Mertens

on July 4, 2005

Committee

Prof. dr. Marc Geysens, chairman, transnationale Universiteit Limburg
Prof. dr. Philippe Bekaert, advisor, transnationale Universiteit Limburg
Prof. dr. Frank Van Reeth, co-advisor, transnationale Universiteit Limburg
Prof. dr. Eddy Flerackers, transnationale Universiteit Limburg
Prof. dr. Frédo Durand, Massachusetts Institute of Technology
Prof. dr. Philip Dutré, Katholieke Universiteit Leuven
Prof. dr. Hans-Peter Seidel, Max Planck Institut Für Informatik

Abstract

In translucent materials such as fruit and marble, light diffuses beneath the surface, creating a distinct appearance. It is impossible to capture this using traditional reflection models, and requires computation of subsurface light transport. Full simulation has proven intractable, fueling the development of more efficient models, capable of retaining the most salient visual qualities. In this dissertation, we will explore the direction of approximating light transport as a diffusion process, in order to achieve efficient and practical simulation of subsurface light transport.

In seminal work by Jensen et al. [JMLH01a], a simplified diffusion model was introduced: the *fast dipole approximation*. It reduces the simulation of subsurface light transport to the computation of a surface integral over the object of interest. We show that the dipole approximation is simple enough to achieve the goal of interactive rendering. Two novel integration schemes are developed, which are fast and flexible enough to interactively account for varying viewpoint, illumination and geometry. In this context, we contribute to the dipole model in several ways: an efficient integration procedure over polygons, an importance sampling scheme for Monte Carlo quadrature, and an adjustment for ensuring reciprocity of light transport. The dipole approximation sacrifices accuracy to increase performance. We will analyze the underlying assumptions, and assess the impact for computer graphics. Most importantly, we find that conservation of energy is not guaranteed, and certain geometry-dependent visual cues are missing.

When interactivity is not the main concern, a more accurate simulation is possible. We introduce a volumetric diffusion method for arbitrarily shaped objects, based on the multigrid method [Hac85, Sta95]. Two important issues are solved. First, accurate representation of interactions near the object's surface, which is realized by applying the so-called embedded boundary discretization [DCL⁺98, JC98]. Second, the solution adaptively refines where required, in order to improve efficiency and keep memory requirements fea-

sible [BO84, DeZ93, BBSW94]. Although this approach may be slower than dipole-based methods, it is capable of dealing with both homogeneous and heterogeneous materials, and more accurately reproduces the general appearance of translucent objects. Contrary to previous methods for dealing with similar cases, computation time is only a few minutes.

The dipole approximation makes no allowance for radiometric accuracy, but enjoys the attractive property of not requiring a volumetric representation. Inspired by the traditional boundary element method [HP02, HASS03], we show that a general exact boundary diffusion model can be formulated. We point out the relationship to the well-known radiosity problem [CW93] and our novel polygon integration technique. As a practical application, the boundary diffusion model is employed for experimental validation of the dipole approximation, which confirms the findings of our analysis.

Samenvatting (Dutch)

In translucente materialen zoals fruit en marmer, wordt licht verstrooid onder het oppervlak, waardoor een uniek uitzicht ontstaat. Deze effecten zijn niet te reproduceren met de traditionele belichtingsmodellen uit het domein van computergrafiek. Een volledige simulatie van wat er zich onder het oppervlak afspeelt, vereist een onredelijke berekeningstijd. Daarom zoeken we naar nieuwe belichtingsmodellen, die de berekeningen simplificeren, maar niet ten koste gaan van de belangrijkste visuele effecten. In dit proefschrift gaan we verder in op het idee om lichttransport te benaderen als een diffusieproces.

In baanbrekend werk door Jensen et al. [JMLH01a], werd een gesimplificeerd diffusiemodel voorgesteld: de snelle dipoolbenadering. Het reduceert de berekeningen van sub-oppervlak lichttransport tot een oppervlakte-integraal over het betreffende object. We tonen aan dat dit model simpel genoeg is om interactieve weergave van translucente materialen toe te laten. Hiervoor ontwikkelen we twee nieuwe integratieprocedures, die snel en flexibel genoeg zijn om zowel kijkpunt, belichting als geometrie interactief aan te passen. In deze context dragen we bij aan de dipoolbenadering met een efficiënte integratieprocedure over polygonen, een voorkeursbemonsteringmethode voor Monte Carlo-integratie, en een aanpassing die reciprociteit van lichttransport garandeert. Alhoewel de dipoolbenadering zeer plausibele resultaten voortbrengt, komt correctheid in het gedrang vanwege de onderliggende aannames. We zullen deze analyseren, en hun impact voor computergrafiek inschatten. In het bijzonder, merken we op dat behoud van energie niet gegarandeerd is en bepaalde geometrie-afhankelijke visuele kenmerken ontbreken.

Wanneer interactiviteit niet primeert, kunnen we een accuratere simulatie realiseren. We introduceren een volumetrische diffusiemethode voor complex gevormde objecten, gebaseerd op het multirooster algoritme [Hac85, Sta95]. Twee belangrijke problemen worden opgelost. Ten eerste, accurate representatie van interacties aan het oppervlak, hetgeen we aangepakken d.m.v. de zogenaamde ingebedde randdiscretizatie [DCL⁺98, JC98]. Ten tweede ge-

bruiken we adaptieve verfijning, teneinde performantie op te krikken en het geheugenverbruik haalbaar te houden [BO84, DeZ93, BBSW94]. Alhoewel volumetrische diffusie trager kan zijn dan methoden gebaseerd op de dipoolbenadering, laat het toe om zowel homo- als heterogene materialen weer te geven, en de kenmerken van translucentie beter te reproduceren. In tegenstelling tot vorige methoden om gelijkaardige situaties aan te pakken, is onze berekeningstijd slechts enkele minuten.

Het ontbreekt de dipoolbenadering aan radiometrische accuraatheid, maar daarentegenstaande komt er geen dure volumetrische representatie aan te pas, omdat de berekeningen enkel aan de rand plaatsvinden. Geïnspireerd door de randelementmethode [HP02, HASS03], tonen we aan dat een algemeen exact randdiffusiemodel kan worden geformuleerd. We observeren overeenkomsten met het bekende radiositeitsprobleem [CW93], en onze nieuwe polygoonintegratieprocedure. Als praktische toepassing zullen we randdiffusie aanwenden om het dipoolmodel experimenteel te valideren, waarmee we de bevindingen van onze analyse bevestigen.

Acknowledgments

First of all, I am grateful for the privilege of working with my advisor, Philippe Bekaert. His ideas, support and encouragement have been instrumental to complete this dissertation. Also, I'd like to thank my co-advisor Frank Van Reeth, for sparking my interest in computer graphics, his confidence in my ability, and for making sure I didn't lack any resources.

A PhD is hardly ever the work of a single person. I'm indebted to those who contributed significantly to the work and ideas presented in this dissertation: Jan Kautz and Tom Haber. Jan's invaluable input is testified by the fact that he co-authored three of my papers. Discussions with Tom have been instrumental, and he deserves most of the credit for chapter 4. If co-authors were to be put on dissertations, these names should be next to mine.

Thanks to the members of my reading committee, for making themselves available, despite their busy schedule.

In 2002, I had the privilege of visiting the graphics lab at the Max Planck Institut für Informatik, thanks to Hans-Peter Seidel and Jan Kautz. Also, I'd like to acknowledge assistance from Jens Vorsatz and Hitoshi Yamauchi during our project. I want to express my gratitude to Frédo Durand and Jan Kautz for realizing my visit to the Massachusetts Institute of Technology, in 2003. While working on our weathering project, input from Jiawen Chen, Tiffany Wang, Paul Green and Eugene Hsu was greatly appreciated. Thanks to Tom Annen, for being an excellent roommate, and sharing a "distinct" sense of humor.

Of course, I am indebted to my colleagues at the Expertise Centre for Digital Media who contributed to this dissertation in one way or another, and I am grateful for all the good times we shared. In particular, I'd like to mention the following people: Fabian Di Fiore, Jan Fransens, Erik Hubo, Pieter Jorissen, Kris Luyten, Tom Van Laerhoven and William Van Haevre. Erik Hubo and Koen Beets deserve all the credit for the cool photographs in figure 1.1. Thanks to Kris Luyten and Fabian Di Fiore for helping out with L^AT_EX is-

sues. Peter Vandoren and Ingrid Konings provided valuable administrative assistance. Finally, thanks to those who reviewed this dissertation.

I'm thankful for the insightful discussions with Simon Premože, and his help in finding some important literature. I'd also like to express my gratitude toward Jensen et al. [JMLH01a] for publishing their material measurements, Paul Debevec for his illumination data [Deb], and the graphics group at Stanford University for sharing their 3D model repository [su3]. The head model in figure 3.13 and 3.14 is courtesy of nVIDIA [nvi]. The "sculpture" model in figure 4.5 is courtesy of FarField technology [far].

Thanks to the people of "Broodjesbar Woody's", whose excellent sandwiches now constitute an important part of my nutrition.

I gratefully acknowledge my sources of funding: the **European Regional Development Fund (ERDF)**, the **Flemish Government** and the **Flemish Interdisciplinary Institute for Broadband Technology (IBBT)**. Also, the **Marie Curie Doctoral Fellowship** for funding my visit to the Max Planck Institut für Informatik in 2002.

Finally, I'd like to thank my family and friends, in particular my parents, Kristof, Laurien, Nancy and Sanna, for their indispensable support.

Diepenbeek — May 2005

*Dedicated to my parents,
for their unconditional support*

Contents

1	Introduction	1
1.1	Scope and Contributions	4
1.2	Applications	5
1.3	Overview of Related Work	5
1.4	Overview of this Dissertation	8
2	Theory of Light Transport	11
2.1	Radiometry	11
2.2	Light Transport in Participating Media	12
2.2.1	Interaction of Light with a Surface	13
2.2.2	Radiative Transfer	14
2.2.3	Scattering Order	17
2.3	Diffusion Approximation	18
2.3.1	Derivation of the Diffusion Equation	18
2.3.2	Boundary Conditions	20
2.3.3	Assumptions	22
2.3.4	Solutions	22
2.4	Bidirectional Distribution Functions	25
2.5	Conclusion	27
3	Fast Dipole Approximation	29
3.1	Dipole Model	30
3.1.1	Diffusion in a Semi-Infinite Medium	30
3.1.2	Dipole BSSRDF	31
3.1.3	Diffusion for Arbitrary Geometry	32
3.2	Efficient Rendering	34
3.3	Hierarchical Boundary Element Method	36
3.3.1	Form Factor Computation	37
3.3.2	Hierarchical Evaluation	45

3.3.3	Results	50
3.4	Local Diffusion	54
3.4.1	Importance Sampling of the BSSRDF	55
3.4.2	Integration over the Surface	57
3.4.3	Implementation	61
3.4.4	Results	63
3.5	Comparison of Interactive Rendering Methods	65
3.5.1	Categorization of Common Techniques	67
3.5.2	Comparison Table	68
3.5.3	Discussion	69
3.6	Analysis of the Fast Dipole Approximation	70
3.6.1	Physical Soundness	70
3.6.2	Phenomenological Soundness	71
3.7	Conclusion	73
4	Volumetric Simulation	75
4.1	Overview	75
4.2	Volumetric Simulation	76
4.2.1	Embedded Boundary Discretization	76
4.2.2	Solution by Means of the Multigrid Method	79
4.2.3	Adaptive Refinement	80
4.2.4	Putting Everything Together	81
4.3	Results	81
4.4	Conclusion	83
5	Boundary Diffusion Model	87
5.1	Boundary Integral Equation	87
5.1.1	Discussion	88
5.2	Solving the Boundary Integral Equation	89
5.2.1	Discretization	89
5.2.2	Form Factor Computation	90
5.3	Experimental Validation of the SIMA	91
5.3.1	Implementation	91
5.3.2	Observations	92
5.3.3	Discussion	95
5.3.4	Conclusion	96
5.4	Conclusion	96

<i>CONTENTS</i>	xi
6 Conclusion	97
6.1 Summary	97
6.2 Future Work	99
A Derivation of Modified Helmholtz BIE	101
B Computing Form Factor F_{ii}^b	103
C Publications	105

Chapter 1

Introduction

Materials like marble, cheese, wax, human skin, jade, etc... interact with light in an unusual way. Instead of being directly reflected off the surface, light is diffused inside, thereby creating a unique appearance (fig. 1.1). We call these materials *translucent*. Practically all objects found in nature exhibit some degree of translucency, with the exception of metals. When observing a translucent object, one notices the soft and diffuse character, concealing small geometric features. Also, there are noticeable chromatic effects, sometimes dubbed “color shifts”, similar to the sky’s color variation from blue to red. These salient effects are important to reproduce, as they add significantly to the photo-realism of a scene.

In this dissertation, we describe methods to efficiently synthesize images of translucent objects. The underlying responsible phenomenon is called *subsurface scattering*: light enters the material, scatters one or more times inside, and exits at a location that is very likely not the same as the entry point. Canonical reflection models in computer graphics only account for local scattering at the surface, thus assuming that light is always reflected directly at the entry point (fig. 1.2). Typically, such renderings look unrealistic and hard when applied to translucent materials. We therefore need novel reflection models and rendering techniques.

One goal in this dissertation, is to render images at interactive rates (1 frame per second or faster). However, simulation of subsurface scattering requires intensive computation. Light transport inside a translucent material can be modeled accurately by the *radiative transfer equation*, a complex integro-differential equation. Direct simulation approaches require many hours to generate a single frame, yet, the appearance caused by subsurface scattering is qualitatively very simple. This observation argues for approximating

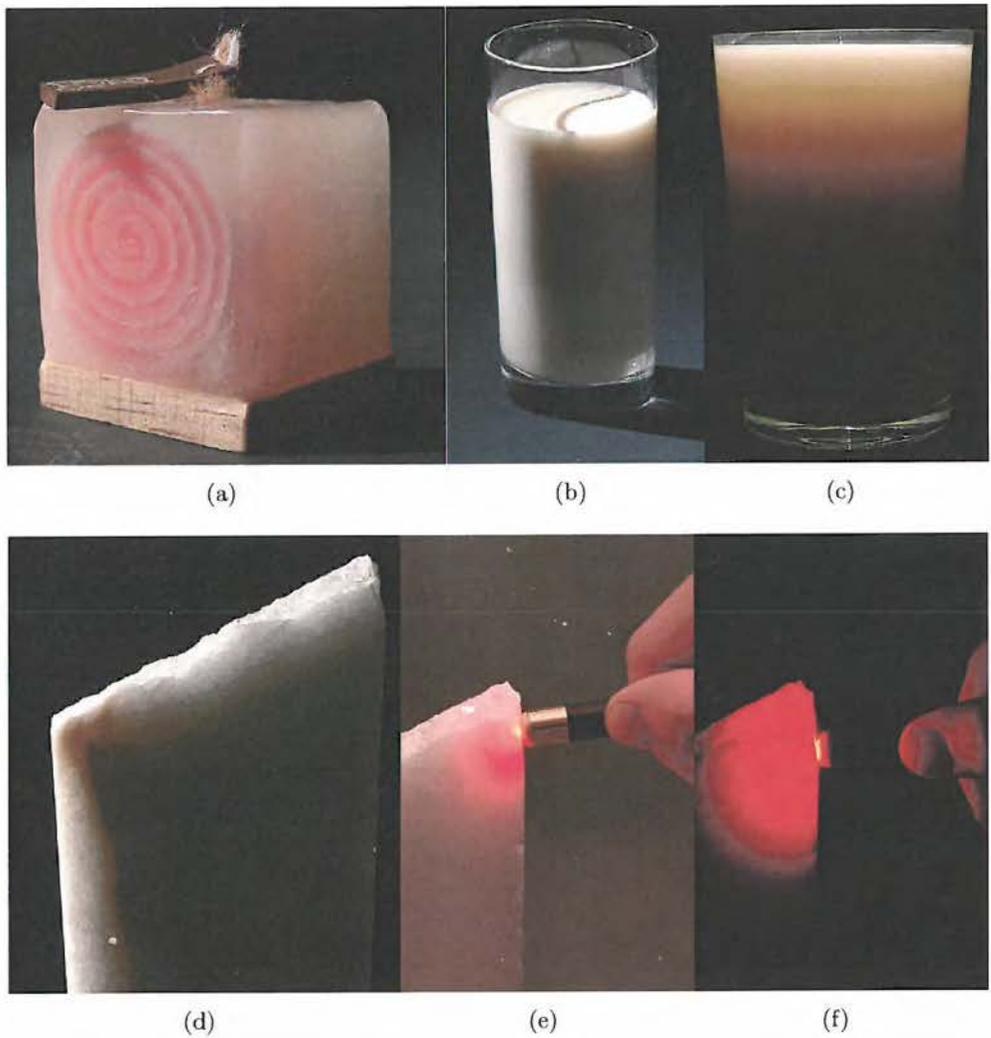


Figure 1.1: Photographs of translucent materials. (a) Candle made of wax. (b) A glass of milk. (c) Same glass of milk lit from above using a flashlight. Note the chromatic effects due to scattering. (d) Marble slab lit from the back. (e-f) Marble slab lit by a red laser. Note the diffusive character of scattered light.

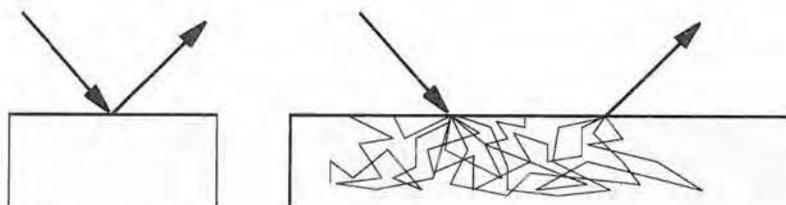


Figure 1.2: **Left:** Traditional reflection model. Light is only reflected at the surface. **Right:** Subsurface scattering. Light penetrates the material, scatters inside and exits at a different location.

the exact solution with various simplifications of the underlying model, in order to achieve the required computational speedup. It turns out that, under certain conditions, light transport can be approximated as a diffusion process, thereby greatly reducing the computational complexity of the problem [Sta95, Kv01, JMLH01a]. In order to treat transport as diffusion, the translucent material should be highly scattering, e.g. like marble, milk and human skin. In this case, light will internally “bounce around” many times, referred to as multiple scattering. Contrary to single scattering (one bounce), multiple scattering causes the typical diffuse appearance (fig. 1.2), and shows less dependence on the direction of incoming light.

Aside from interactive rendering, we will discuss how diffusion can be used to render more complex situations, like heterogeneous materials. Although these situations cannot be dealt with at interactive rates, we present a novel rendering method which is more efficient than techniques which directly solve the radiative transfer equation.

Terminology

In order to avoid possible confusion throughout this dissertation, it is appropriate to define some important terminology:

- *Participating Medium*: matter which “influences” passing light by scattering or absorbing it. It can be gaseous (e.g. fog), liquid (e.g. milk) or solid (e.g. skin).
- *Transparent Medium*: a medium which allows for light passing through, having a clear and sharp character (e.g. glass).
- *Translucent Medium*: a medium which allows for penetration of light in a diffusive fashion, like paper and marble, for instance. It blurs out most

detail, and does not convey form.

- *Subsurface Scattering*: the underlying phenomenon causing translucency.
- *Subsurface Diffusion*: the special case of subsurface scattering, when the effect of scattering can be observed as a diffusion process, due to a high amount of scattering events. In particular, the dependence on the incoming light direction can be neglected.

1.1 Scope and Contributions

In this dissertation, we will present methods to efficiently render objects, consisting of highly scattering materials. Single scattering will not be treated, as its contribution is small compared to multiple scattering in this situation.

Note that scattering or diffusion may also occur directly at a (rough) surface, even in “clear” materials, where subsurface scattering is negligible (for instance, matte glass). Such scattering may appear similar to subsurface scattering, yet the underlying process is completely different. We will not treat this case here.

Our practical contributions can be summarized as follows:

- We propose efficient **interactive** rendering algorithms using the fast dipole approximation, as introduced by Jensen et al. [JMLH01a]. In particular, we develop a local diffusion and hierarchical boundary method. In this context, we make contributions to the dipole model: an efficient integration procedure over polygons, an alternative dipole placement scheme which ensures reciprocity for arbitrary meshes, and an importance sampling scheme.
- We introduce an efficient and flexible rendering algorithm, which is able to accurately **simulate** diffusion in arbitrarily shaped objects. In particular, we apply a suitable discretization scheme, which solves stability issues near the boundary, and solve the diffusion problem using a fast adaptive multigrid method.
- We introduce a boundary diffusion model, inspired by the traditional boundary element method. It will mainly be used to carry out an *experimental validation* of the fast dipole approximation.

Most of these contributions have appeared in academic literature (see appendix C for an overview of the published work). The analysis and validation

of the fast dipole approximation (section 3.6 and 5.3) are original contributions of this dissertation.

1.2 Applications

We see the application of our work in these areas:

Real-Time Photo-Realistic Rendering One of the main goals of computer graphics is to generate photo-realistic images, as fast as possible. Many applications like computer games, virtual reality, simulations and so on, require that these images are computed at interactive rates. In this context, we present interactive techniques for rendering translucent objects, e.g. marble, food, A particularly important example for translucent materials is human skin, which is essential for displaying virtual characters.

Efficient Predictive Rendering We are interested in simulating reality, in order to predict what materials will look like under certain illumination conditions. We strive to achieve not only plausible results, but also (to some degree) radiometric accuracy. This can be important for architectural applications, or studying perception of materials [FDA03, FJB04]. Predictive rendering (forward problem) may be employed to solve the inverse problem, i.e. estimation of material parameters or acquiring internal structure. Similar approaches have been developed for tomographic imaging in biological tissue [Hie97].

1.3 Overview of Related Work

Here, we present an overview of relevant published work.

Light Transport Simulation in Participating Media

In theory, subsurface scattering effects can be rendered using methods which deal with participating media. A lot of related work has been presented over the past 2 decades, and we refer to Perez et al.'s survey and Dutré et al.'s book for the less recent work [PCPS97, DBB03]. In the last few years, Monte Carlo rendering in participating media has made significant progress, e.g. with Metropolis light transport [PKK00], photon mapping [Jen96, DEJ⁺99], and the scattering framework by Pharr et al. [PH00]. These methods require a

significant amount of time to compute a solution, and do not show much promise toward reaching interactivity.

More recently, Premože et al. [PAS03] presented a mathematical framework for path integration, which led to an efficient (offline) rendering algorithm [PAT⁺04] capable of accurately rendering multiple scattering effects. Hegeman et al. [HAP05] implemented a simplified version on graphics hardware, yielding interactive results for varying illumination. Related to this, is the work by Kniss et al. [KPHE02], who proposed an empirical model for rendering translucent volume data.

To conclude our discussion, we note that most of these approaches are too expensive to generate images within a reasonable amount of time. Hegeman et al. and Kniss et al. obtain interactive rendering rates at the cost of significant simplifications of the transport model [HAP05, KPHE02]. Moreover, we observe that most methods are geared toward phenomena like fog and smoke, and require a full volumetric description of the medium. In our case, the setting is different:

- Translucent objects are usually denser and scatter more. As a result, multiple scattering becomes the most important contribution.
- The medium is bounded by a “hard” interface, and has a different index of refraction than air. Care has to be taken to include the effects of refraction and internal reflection.
- The viewer and light sources are always outside of the medium. Moreover, we are only interested in outgoing radiance at the interface. Therefore, it is desirable to restrict computations to the surface of the object, and avoid (or at least reduce) costly volumetric computations typically required in the previously mentioned approaches.

These observations argue for specific models and rendering algorithms for dealing with translucent objects.

Precomputation and Data-Driven Techniques

The precomputed radiance transfer framework [SKS02] is capable of rendering arbitrary (linear) light transport under varying distant, band-limited illumination. In particular, it has been shown to work for participating media [SKS02] and subsurface diffusion [SHHS03, JMLH01a]. Wang et al. [WTL05] extended this technique to deal with high-frequency illumination and single scattering. Rendering rates are interactive. Other precomputation techniques based on

diffusion [JMLH01a] have been proposed [LGB⁺02, CHH03, HV04] in order to achieve interactive rendering. They will be discussed in detail, in section 3.5.

Chen et al. [CTW⁺04] proposed to precompute light transport in a local heterogeneous volume and replicate this information over an object's surface using texture synthesis [TZL⁺02]. The remaining part (inner core) of the object is treated using Jensen's diffusion model [JMLH01a]. After precomputation, rendering an image requires several seconds, up to a minute.

Gösele et al. [GLL⁺04] proposed a technique to acquire the appearance of real-world heterogeneous translucent objects. They measure the response to incident laser light at every location of the surface using high-dynamic range imaging. The captured objects are displayed interactively under novel illumination using Lensch et al.'s approach [LGB⁺02].

Conceptually, precomputation and data-driven techniques are the same. Both approaches rely on a large collection of data, from which renderings are "inferred" under novel illumination conditions. The difference lies in how the data is acquired (synthesized or measured), and its representation.

For pure precomputation, the simulation step usually takes many hours, and only allows for relighting. As a result, they can only be used for static scenarios. Real-world acquisition yields plausible renderings, but cannot deal with synthetic geometry and material parameters, and does not allow for editing. Compact representation is the most important issue, as it is directly related to the efficiency of run-time rendering. Basically, the rendering (relighting) problem is reformulated as a compression problem. In our work, we aim at reducing the complexity of the underlying simulation, instead of accelerating relighting. In fact, our work is complementary to these techniques, as it may be used to speed up the precomputation phase.

Subsurface Scattering

We can distinguish two trends in methods for rendering subsurface scattering: one-dimensional (1D) and full 3D scattering.

- **1D** One-dimensional techniques approximate scattering along the depth only (w.r.t. the object's surface), ignoring lateral transport. The Kubelka Munk model [KM31, Pha01] approximates diffuse (independent of incoming light's orientation) transport layered media. Hanrahan and Krüger [HK93] analyzed the 1D radiative transfer equation for layered media. An analytical solution for single scattering was developed, and formulated as a BRDF. Pharr and Hanrahan [PH00] proposed an efficient

Monte Carlo-based rendering framework based on scattering equations. 1D transport is simple, but yields non-plausible results as light may exit at a location different than the point of entry.

- **3D** Scattering equations can be extended to three dimensions [PH00]. Dorsey et al. [DEL⁺99] simulated subsurface scattering in stone using photon mapping. Both these methods are very flexible and accurate, yet very slow, in particular for highly scattering materials. Premože et al.'s algorithm [PAS03] is less dependent on scattering order, but still requires a significant amount of computation time.

Diffusion

Diffusion [Sta95, Kv01, JMLH01a] is a sound approximation to general light transport in the case of highly scattering media, i.e. where direct simulation of scattering breaks down. It significantly reduces the computational complexity of light transport. Stam was the first to suggest the diffusion approximation for rendering translucent materials [Sta95]. He applied the multigrid method to solve the diffusion equation, but only demonstrated it on a simple, illustrative example. In chapter 4, we will extend his work to deal with arbitrarily shaped objects, and even heterogeneities.

Jensen et al. [JMLH01a] introduced the fast dipole approximation (FDA), which does not require numerical procedures. Instead, an approximate analytical solution to the diffusion equation for a semi-infinite homogeneous slab is used directly on arbitrary 3D objects. This crude approximation yields highly plausible renderings and serves as the basis for several interactive methods [LGB⁺02, DS03, CHH03, HV04]. In the same spirit, we introduce novel algorithms which obtain interactive performance; see section 3.3 and 3.4. Also, an analysis and experimental validation will be presented in sections 3.6 and 5.3, respectively.

1.4 Overview of this Dissertation

In chapter 2, we review the theory of light transport and diffusion. The popular fast dipole approximation (FDA) [JMLH01a] will be the main topic of chapter 3. In this context, we introduce novel rendering algorithms which enable us to interactively render translucent objects. A detailed comparison to other interactive methods is presented as well. We conclude with an analysis of the fast dipole approximation. In chapter 4, a novel simulation technique

is introduced to accurately solve the diffusion problem in arbitrary objects and materials. Chapter 5 introduces a generalized boundary diffusion model, which will mainly be used to experimentally validate the FDA. Finally, chapter 6 summarizes our main contributions and findings, and lays out directions for future work.

Chapter 2

Theory of Light Transport

Before discussing practical solutions to rendering translucent objects, a brief treatment of mathematical and physical preliminaries is appropriate.

Our problem is set in the more general framework of radiative transfer [Cha60], which describes the transport of light in a medium (in our case the translucent material). We will also recapitulate how radiative transfer can be approximated efficiently by diffusion.

We start off by discussing the relevant physical quantities, borrowed from the field of radiometry.

In the remainder of this dissertation, we will denote the volume occupied by a translucent object as Ω , and its boundary (surface) as Γ .

2.1 Radiometry

Light is a form of electromagnetic radiation, which can be considered as waves or particles. For simplicity, we will explain radiometric quantities in terms of “light particles”, or photons. Each photon carries a specific quantity of energy, hence the total energy is proportional to the number of photons. To avoid an unnecessarily complex treatment of physical quantities, we will just consider energy, the basic unit in radiometry, as the number of photons. Table 2.1 summarizes the radiometric quantities relevant to image synthesis.

Radiance is particularly important for image synthesis, as it is the same quantity recorded by a camera or human eye. For a ray $(\mathbf{x}, \vec{\omega})$, it is proportional photon hit rate within a small differential area dA at \mathbf{x} , with photons coming from small differential solid angle $d\omega$ around direction $\vec{\omega}$. To compensate for the relative orientation θ of dA w.r.t. $\vec{\omega}$, we weigh the solid angle by

Quantity	Notation	Definition	Units
Energy	Q	\sim #photons	Joule
Flux (power)	Φ	dQ/dt	Watt
Irradiance	$E(\mathbf{x})$	$d\Phi/dA$	Watt/ m^2
Radiosity	$M(\mathbf{x})$	$d\Phi/dA$	Watt/ m^2
Radiance	$L(\mathbf{x}, \vec{\omega})$	$d^2\Phi/d_{\perp}\omega dA$	Watt/ m^2sr

Table 2.1: Overview of important radiometric quantities.

a cosine factor: $d_{\perp}\omega = \cos\theta d\omega$.

Flux density is also important, as it can be considered as an angle-independent (diffuse) variant of radiance. It comes in two flavors: irradiance and radiosity. The former measures incoming radiance L , cosine-integrated over the sphere or hemisphere, depending on whether we are measuring on a surface or in a volume:

$$E(\mathbf{x}) = \int L(\mathbf{x}, \vec{\omega}) d_{\perp}\omega$$

Radiosity (a.k.a. radiant exitance) is similar, but equals the cosine-integrated outgoing radiance L , i.e. radiance after reflection/scattering, in the case of surfaces/volumes, respectively:

$$M(\mathbf{x}) = \int L(\mathbf{x}, \vec{\omega}) d_{\perp}\omega$$

To convert radiosity to (diffuse) radiance, we normalize the above integral, i.e. $L(\mathbf{x}) = \frac{1}{\pi}M(\mathbf{x})$.

2.2 Light Transport in Participating Media

The theory of light transport is our starting point for realistic image synthesis. This section is devoted to introducing the underlying model for transport in participating media: *radiative transfer*. In our setting, the participating medium refers to a translucent object.

Note that not all optical phenomena will be accounted for. Geometric optics are mostly relevant for realistic image synthesis, i.e. reflection, transmission and volumetric scattering. Effects like diffraction, interference, polarization, fluorescence, phosphorescence are ignored. Also note that we are only interested in a steady-state solution of transport; time-dependency is ignored

as well. Including the aforementioned phenomena would significantly complicate transport modeling, while their effects are perceivable at most in a few pathological cases.

2.2.1 Interaction of Light with a Surface

Before explaining what goes on inside, we need to discuss how light enters the object. When a ray arrives at the interface which separates the vacuum from the medium, it is “split” into a reflected and transmitted part (fig. 2.1). The fraction of reflection at a surface is called reflectance. Formally, we define it as the ratio of reflected and incident flux:

$$\rho = \frac{\Phi_{reflected}}{\Phi_{incident}}$$

Similarly, transmittance can be defined by $\frac{\Phi_{transmitted}}{\Phi_{incident}}$.

For a perfectly smooth planar surface, the reflected outgoing angle w.r.t. the surface normal θ_r equals the incoming angle θ_i (ideal reflection, like a mirror). The transmitted ray’s angle θ_t is given by Snell’s law of refraction: $n_1 \sin \theta_i = n_2 \sin \theta_t$ [BW59], where the respective indices of refraction are $n_1 = 1$ for vacuum, and n_2 for the material bounded by the surface. We denote $\eta = n_2/n_1$ as the relative refraction index.

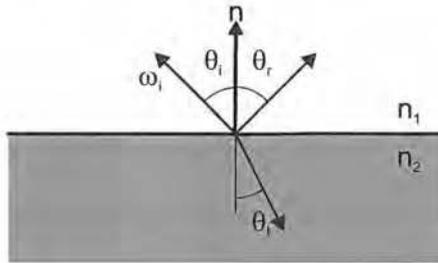


Figure 2.1: Reflection and transmission of an incoming ray of light at a perfectly smooth surface.

Reflectance is described by the Fresnel formulae [BW59],

$$F_r^{\parallel} = \frac{n_2 \cos \theta_i - n_1 \cos \theta_t}{n_2 \cos \theta_i + n_1 \cos \theta_t}$$

$$F_r^{\perp} = \frac{n_1 \cos \theta_i - n_2 \cos \theta_t}{n_1 \cos \theta_i + n_2 \cos \theta_t}$$

where we have F_r^{\parallel} and F_r^{\perp} as reflectance for light polarized parallel and perpendicularly to the plane of incidence, respectively. These terms can be combined to obtain the reflectance for the unpolarized case:

$$F_r(\eta, \theta_i) = \frac{1}{2}(F_r^{\parallel 2} + F_r^{\perp 2})$$

The corresponding transmittance equals $F_t(\eta, \theta_i) = 1 - F_r(\eta, \theta_i)$.

Dielectric (or non-conducting) media have a real index of refraction. For angles $\theta_i \rightarrow \pi/2$, Fresnel reflectance becomes dominant w.r.t. transmittance. This effect can be observed in reality, as surfaces appear to be ideal reflectors when viewing from a grazing angle. For perpendicularly impinging directions, more light passes through the interface, into the medium. Conductors (complex refraction index) exhibit a negligible amount of transmission, regardless of the angle of incidence. As a result, subsurface scattering becomes less important here.

2.2.2 Radiative Transfer

Following Chandrasekhar [Cha60], we will introduce the radiative transfer equation (RTE), which describes light transport in participating media.

A participating medium is modeled as a set of particles, for instance molecules, or water droplets (fog). Contrary to transport in vacuum, the presence of particles causes radiance to change along a ray. These interactions can be classified into 3 different events: *absorption*, *out-scattering*, and *in-scattering*. Also, photons can be added to the equation, as a result of *emission* from a light source. We will discuss each event, and show how they can be combined to form the radiative transfer equation. See figure 2.2 for an illustration.

Let us consider a ray $R = (\mathbf{x}, \vec{\omega})$ in 3D space, and observe what can happen over a small differential distance ds .

Absorption When a photon hits a particle, it may be absorbed and turned into a different type of energy, e.g. heat. The amount of absorption is defined by the coefficient $\sigma_a(\mathbf{x})$, which is described as the absorption probability per unit length.

$$\begin{aligned} L(\mathbf{x} + \Delta s \vec{\omega}, \vec{\omega}) &= L(\mathbf{x}, \vec{\omega}) - L(\mathbf{x}, \vec{\omega})\sigma_a(\mathbf{x})\Delta s \\ \Leftrightarrow \frac{L(\mathbf{x} + \Delta s \vec{\omega}, \vec{\omega}) - L(\mathbf{x}, \vec{\omega})}{\Delta s} &= -L(\mathbf{x}, \vec{\omega})\sigma_a(\mathbf{x}) \end{aligned}$$

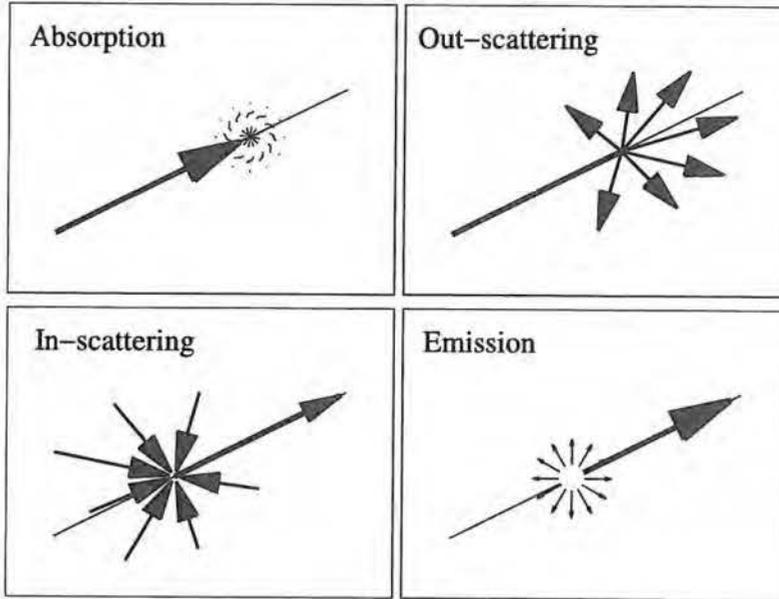


Figure 2.2: Possible interactions of light with a participating medium.

For $\Delta s \rightarrow 0$:

$$\frac{dL(\mathbf{x}, \vec{\omega})}{ds} = -L(\mathbf{x}, \vec{\omega})\sigma_a(\mathbf{x})$$

Out-scattering A particle may also alter a photon's direction due to reflection or refraction. This is the phenomenon we call scattering. Analogously to the absorption coefficient, we define $\sigma_s(\mathbf{x})$ as the probability per unit length that a photon will be scattered.

The combined effect of absorption and out-scattering is described by the extinction coefficient $\sigma_t = \sigma_a + \sigma_s$. Mathematically, we have:

$$\frac{dL(\mathbf{x}, \vec{\omega})}{ds} = -L(\mathbf{x}, \vec{\omega})\sigma_t(\mathbf{x})$$

The (scattering) albedo is given by

$$\alpha = \frac{\sigma_s}{\sigma_a + \sigma_s}$$

interpreted as the "survival" probability of a photon after scattering.

A photon's mean free path $l_{mfp} = 1/\sigma_t$, measures the average distance a photon can travel without being absorbed or scattered.

In-scattering A photon may be scattered from another ray onto R . We account for these additional photons with the following term:

$$L^{in}(\mathbf{x}, \vec{\omega}) = \sigma_s(\mathbf{x}) \int_{4\pi} p(\mathbf{x}, \vec{\omega}', \vec{\omega}) L(\mathbf{x}, \vec{\omega}') d\omega' \quad (2.1)$$

The phase function p describes the distribution of new photon directions after scattering, given an incoming direction. p is normalized: $\int_{4\pi} p(\mathbf{x}, \vec{\omega}', \vec{\omega}) d\omega' = 1$. L^{in} is expressed in $\frac{W}{m^3 sr}$, i.e. volume radiance. In the remainder of this text we assume the phase function only depends on the angle or its cosine: $p(\cos \theta = \vec{\omega}' \cdot \vec{\omega})$. This is valid for isotropic media, i.e. the phase functions are independent of the orientation of the medium.

Commonly, p is modeled empirically by the Henyey-Greenstein function [HG41]:

$$p(\theta) = \frac{1 - g^2}{4\pi(1 + g^2 - 2g \cos \theta)^{\frac{3}{2}}} \quad (2.2)$$

parameter g acts as an asymmetry parameter. Near -1 and 1, the phase function exhibits strong backward and forward scattering, respectively. For $g = 0$, the phase function is isotropic.

The change of radiance due to in-scattering is:

$$\frac{dL(\mathbf{x}, \vec{\omega})}{ds} = L^{in}(\mathbf{x}, \vec{\omega})$$

Emission A light source in the medium may add photons to R . We denote it by $Q(\mathbf{x}, \vec{\omega})$, with units $\frac{W}{m^3 sr}$. Along our differential distance, the increase is defined by:

$$\frac{dL(\mathbf{x}, \vec{\omega})}{ds} = Q(\mathbf{x}, \vec{\omega})$$

Radiative Transfer Equation Combining the previous 4 terms, we obtain the radiative transfer equation (RTE):

$$\frac{dL(\mathbf{x}, \vec{\omega})}{ds} = Q(\mathbf{x}, \vec{\omega}) - \sigma_t(\mathbf{x})L(\mathbf{x}, \vec{\omega}) + L^{in}(\mathbf{x}, \vec{\omega})$$

The accompanying boundary condition states that no scattered radiance should enter from outside the medium (we will return to this in the following section).

Another form of the RTE is obtained by integrating along the ray throughout the medium. Consider a segment starting at point \mathbf{x} , ending at the boundary of the medium: $(\mathbf{x}, \mathbf{x} + s\vec{\omega})$. Integration yields¹:

$$\begin{aligned} L(\mathbf{x}, \vec{\omega}) &= \int_0^s e^{-\tau(\mathbf{x}, \mathbf{x} + t\vec{\omega})} Q(\mathbf{x} + t\vec{\omega}) dt && \text{emission} \\ &+ e^{-\tau(\mathbf{x}, \mathbf{x} + s\vec{\omega})} L(\mathbf{x} + s\vec{\omega}, \vec{\omega}) && \text{absorption and out-scattering} \\ &+ \int_0^s e^{-\tau(\mathbf{x}, \mathbf{x} + t\vec{\omega})} L^{in}(\mathbf{x} + t\vec{\omega}, \vec{\omega}) dt && \text{in-scattering} \end{aligned}$$

$\tau(\mathbf{x}, \mathbf{x} + s\vec{\omega}) = \int_0^s \sigma_t(\mathbf{x} + t\vec{\omega}) dt$ is called the optical depth, and measures opacity. If for a typical path through the medium $\tau < 1$, the medium is considered “optically thin”, meaning that most photons will go right through the material (i.e. transparent, like glass). For the mean free path length l_u , $\tau = 1$. For optically thick materials, photons will not pass through easily, and are likely to be absorbed or scattered.

2.2.3 Scattering Order

The scattering order refers to the number of scattering events of a photon. We typically distinguish 3 types of scattering orders: no scattering, single scattering and multiple scattering. Each of these has special characteristics, and is therefore treated differently in practice.

No Scattering For transparent materials, photons are likely to pass through without causing a scattering event. Also, in optically thick materials which mostly absorb light, we can neglect scattering.

Single Scattering For thick materials with only a small optical depth, most photons will undergo a single scattering event.

Multiple Scattering For high albedos, a photon may undergo several hundreds of scattering events before leaving the object or being absorbed. As a result, the contribution from multiple scattering becomes very important. During each of those events, the dependence on light’s incoming orientation decreases, resulting in diffuse outgoing radiance at the surface [Sta95]. This is the main motivation for using the diffusion approximation (see next section).

¹This equation is also known as the volume rendering equation.

2.3 Diffusion Approximation

Our goal is to efficiently solve the RTE, by approximating light transport as a diffusion process. This is accomplished by sacrificing angle-dependence. As a result, the transport equation boils down to a simple linear partial differential equation (PDE). There are 2 good reasons for making this simplification:

- Multiply scattered light has an isotropic nature. A sufficient condition is that the material has a constant (or isotropic) phase function, i.e. $p = \frac{1}{4\pi}$. When the phase function is not constant, in highly scattering materials (high albedo), angular dependence is lost due to the sheer number of scattering events [Sta95]. For the particular case of visible subsurface scattering, these conditions are usually satisfied, due to the optical thickness of materials (e.g. marble).
- Diffusion can be used as a partial solution. For instance, it can provide a good estimate for contribution of multiple scattering, while single scattering might be treated by another method [JMLH01a]. Also note that in optically thick materials, the contribution of multiple scattering is dominant, and therefore single scattering is negligible.

In the following section, we will derive the diffusion equation.

2.3.1 Derivation of the Diffusion Equation

We consider the radiance to be composed of the reduced incident radiance L_{ri} , which is the attenuated radiance due to absorption and scattering, and the diffuse radiance L_d , representing light that has been scattered at least once (and therefore likely being diffuse):

$$L = L_{ri} + L_d$$

The first term obeys:

$$(\vec{\omega} \cdot \vec{\nabla})L_{ri}(\mathbf{x}, \vec{\omega}) = -\sigma_t(\mathbf{x})L_{ri}(\mathbf{x}, \vec{\omega})$$

Hence, we compute it by:

$$L_{ri}(\mathbf{x}, \vec{\omega}) = e^{-\tau(\mathbf{x}, \mathbf{x}_b)} L(\mathbf{x}_b, \vec{\omega}) \quad (2.3)$$

where \mathbf{x}_b is the point of entry of ray $(\mathbf{x}, \vec{\omega})$ on the medium's boundary.

Let L_{ri}^{in} and L_d^{in} be defined analogously to equation (2.1). The remaining part of the RTE is expressed in terms of L_d , and becomes:

$$(\vec{\omega} \cdot \vec{\nabla})L_d(\mathbf{x}, \vec{\omega}) = Q(\mathbf{x}, \vec{\omega}) + L_{ri}^{in}(\mathbf{x}, \vec{\omega}) - \sigma_t(\mathbf{x})L_d(\mathbf{x}, \vec{\omega}) + L_d^{in}(\mathbf{x}, \vec{\omega}) \quad (2.4)$$

Term L_{ri}^{in} represents the first scattering event of incoming radiance, and can be seen as an additional source. Let us merge the two source terms into $Q_+ := Q + L_{ri}^{in}$ for simplicity.

Under the assumption of L_d being practically diffuse, it can be approximated by the following linear expansion:

$$L_d(\mathbf{x}, \vec{\omega}) \approx \frac{1}{4\pi}\phi(\mathbf{x}) + \frac{3}{4\pi}\vec{\omega} \cdot \vec{E}(\mathbf{x}) \quad (2.5)$$

Here, we employed the first 2 moments of L_d , defined by

$$\phi(\mathbf{x}) := L_d^0(\mathbf{x}) = \int_{4\pi} L_d(\mathbf{x}, \vec{\omega})d\omega \quad \text{and} \quad \vec{E}(\mathbf{x}) := \vec{L}_d^1(\mathbf{x}) = \int_{4\pi} L_d(\mathbf{x}, \vec{\omega})\vec{\omega}d\omega \quad (2.6)$$

Scalar function ϕ is dubbed “fluence”, and can be interpreted as the photon density or concentration of light. Vector function \vec{E} is referred to as the “flux vector”, and represents the net flow vector along which photons travel. The expansion (2.5) can be derived by projecting L_d into the spherical harmonics basis, or from the Taylor expansion of the cosine around $\vec{\omega}$. As in equation (2.6), let us also define Q_+^0 as the first, and \vec{Q}_+^1 as the second moment of Q_+ .

Deriving the diffusion equation now boils down to filling in L_d 's expansion in the RTE. In the following, we briefly highlight the most important steps in this process. More details can be found in the book by Ishimaru [Ish78]. To avoid clutter, we implicitly assume dependence on \mathbf{x} where obvious. For reference, the relevant variables are summarized in table 2.2.

Integrating equation (2.4) over the sphere on both sides yields

$$\vec{\nabla} \cdot \vec{E} = Q_+^0 - \sigma_a\phi \quad (2.7)$$

Returning to the RTE (2.4), we insert the diffuse radiance approximation:

$$\frac{1}{4\pi}(\vec{\omega} \cdot \vec{\nabla})\left[\phi + 3\vec{E} \cdot \vec{\omega}\right] = Q_+(\mathbf{x}, \vec{\omega}) - \frac{1}{4\pi}\sigma_a\phi - \frac{3}{4\pi}\sigma'_t\vec{E} \cdot \vec{\omega}$$

where we used the *reduced* coefficients $\sigma'_t = \sigma_a + \sigma'_s$ and $\sigma'_s = (1 - g)\sigma_s$, which in turn are derived from the phase functions anisotropy factor g (corresponds to anisotropy parameter in equation (2.2)):

$$g = \int_{4\pi} p(\vec{\omega}' \cdot \vec{\omega})(\vec{\omega}' \cdot \vec{\omega})d\omega' \quad (2.8)$$

We continue by multiplying equation (2.7) on each side by $\vec{\omega}$, and integrating over the unit sphere, yielding:

$$\frac{1}{3}\vec{\nabla}\phi = \vec{Q}_+^1 - \sigma'_t\vec{E} \quad (2.9)$$

We now eliminate \vec{E} from (2.7). Rearranging equation (2.9), and substituting \vec{E} , leads us to the diffusion equation:

$$-\vec{\nabla} \cdot [D\vec{\nabla}\phi] + \sigma_a\phi = Q_+^0 + 3\vec{\nabla} \cdot [D\vec{Q}_+^1] \quad (2.10)$$

with the diffusion coefficient $D = \frac{1}{3\sigma'_t}$. We have now derived the steady state (time-invariant) diffusion equation. Alternatively, in case the medium is homogeneous (the coefficients σ_a and σ_s do not vary spatially), we write it as:

$$-(\nabla^2 - \sigma_e^2)\phi = \frac{1}{D}Q_+^0 + 3\vec{\nabla} \cdot \vec{Q}_+^1 \quad (2.11)$$

with $\sigma_e = \sqrt{3\sigma'_t\sigma_a}$ the effective transport coefficient. In the remainder of this dissertation, we denote the right hand side of equations 2.10 and 2.11 as $S(\mathbf{x})$.

2.3.2 Boundary Conditions

The diffusion equation (2.10) is a differential equation, hence a boundary condition is required to obtain a unique solution. Again, we follow Ishimaru to derive this condition [Ish78].

For the diffuse radiance equation (2.4), we observe that no diffuse radiance should enter the medium. Mathematically, $L_d(\mathbf{x}, \vec{\omega}) = 0$ for x at the boundary and $\vec{\omega}$ pointing towards the medium w.r.t. the boundary's normal.

Since diffuse radiance is essentially low pass filtered in the angular domain (eq. 2.5), this condition cannot be stated exactly. We therefore weaken it, by stating that total inward flux must be zero². In other words:

$$\int_{2\pi^-} L_d(\mathbf{x}, \vec{\omega})(\vec{\omega} \cdot \vec{n})d\omega = 0 \quad (2.12)$$

We fill in the two term expansion into (2.12):

$$\frac{1}{4\pi}\phi(\mathbf{x}) \int_{2\pi^-} (\vec{\omega} \cdot \vec{n})d\omega + \frac{3}{4\pi} \int_{2\pi^-} (\vec{E}(\mathbf{x}) \cdot \vec{\omega})(\vec{\omega} \cdot \vec{n})d\omega = 0$$

²Contrary to Ishimaru [Ish78], in our case the normal points away from the medium.

The first term reduces to $\frac{1}{4}\phi(\mathbf{x})$. In the second term, \vec{E} can be decomposed into $E_n\vec{n} + E_t\vec{t}$ where \vec{t} is a tangent vector at the surface. Using spherical coordinates for the system defined by \vec{t} and \vec{n} , we can rewrite the second term as:

$$\frac{3}{4\pi}E_t(\mathbf{x}) \underbrace{\int_{2\pi^-} (\vec{\omega} \cdot \vec{t})(\vec{\omega} \cdot \vec{n})d\omega}_0 + \frac{3}{4\pi}E_n(\mathbf{x}) \underbrace{\int_{2\pi^-} (\vec{\omega} \cdot \vec{n})^2d\omega}_{-\frac{2\pi}{3}} \quad (2.13)$$

The boundary condition becomes:

$$\phi(\mathbf{x}) - 2E_n(\mathbf{x}) = 0 \quad (2.14)$$

It can be rewritten using only ϕ by plugging in equation (2.9) for $\vec{E} \cdot \vec{n} = E_n$:

$$\phi + 2D \frac{\partial \phi}{\partial n} = -\frac{2}{\sigma'_t} \vec{n} \cdot \vec{Q}_+^1 \quad (2.15)$$

So far we have assumed the refraction indices were equal. For realistic materials, we need to account for internal reflection in the medium due to a possible refraction index mismatch. Instead of specifying that the total inward diffuse radiance should be nil, we equalize it to the internally reflected diffuse radiance [MSS88]:

$$\int_{2\pi^-} L_d(\mathbf{x}, \vec{\omega})(\vec{\omega} \cdot \vec{n})d\omega = \int_{2\pi^+} F_r(\eta^{-1}, \vec{\omega} \cdot \vec{n})L_d(\mathbf{x}, \vec{\omega})(\vec{\omega} \cdot \vec{n})d\omega \quad (2.16)$$

The relative index is inverted, as reflection is measured on the inside of the object. Applying the same derivation again (2.15), we obtain:

$$\phi + 2AD \frac{\partial \phi}{\partial n} = -A \frac{2}{\sigma'_t} \vec{n} \cdot \vec{Q}_+^1 \quad (2.17)$$

with $A = \frac{1+F_{dr}}{1-F_{dr}}$ and the diffuse Fresnel reflectance:

$$F_{dr}(\eta^{-1}) = \int_{2\pi} F_r(\eta^{-1}, \vec{n} \cdot \vec{\omega})(\vec{n} \cdot \vec{\omega})d\omega$$

Groenhuis et al. proposed a simple approximation [GFB83]:

$$F_{dr}(\eta^{-1}) \approx -\frac{1.44}{\eta^2} + \frac{0.71}{\eta} + 0.668 + 0.0636\eta$$

If Q_+ is isotropic, we have a simpler boundary condition:

$$\phi + 2AD \frac{\partial \phi}{\partial n} = 0 \quad (2.18)$$

Since fluence should always be positive, this equation tells us that the concentration of light decreases along the normal direction. In other words, light will always “leak out” at the boundary. This makes sense, as there is less “medium” for light to scatter in.

2.3.3 Assumptions

Care must be taken when applying the diffusion approximation. We will review the most relevant issues here.

- The reduced scattering albedo must be $\alpha' \approx 1$, or $\sigma'_s \gg \sigma_a$. In other words, **multiple scattering should be dominant**.
- The “reduced” mean free path $l'_{mfp} = 1/\sigma'_t$ should be sufficiently less than the geometric size [Rip00], in order for diffusion to take place. In other words, **the medium should be optically thick**.
- In the diffusion equation, the phase function’s behavior is modeled using the anisotropy parameter g . Isotropic scattering ($g = 0$) is an ideal case for diffusion. But even for $g > 0$, diffusion is still valid, as abundant forward scattering will still diffuse radiance. Ripoll [Rip00] notes that the diffusion approximation is **valid for g ’s up to .85**.
- It should be noted that the diffusion approximation’s accuracy near the boundary is low. The weak formulation of the boundary condition (2.12) is a symptom of this. Nevertheless, for computer graphics, diffusion is accurate enough to generate plausible renderings.

2.3.4 Solutions

Infinite Homogeneous Medium

The differential operator $L = \nabla^2 - \sigma_e^2$ in the diffusion equation (2.11) can be identified as the 3D modified Helmholtz operator. The corresponding Green’s function G , is defined as the solution of L w.r.t. to an impulse source term:

$$L\{G(\mathbf{x}, \mathbf{y})\} = -\delta(\mathbf{x} - \mathbf{y})$$

For this particular operator, the Green’s function is analytically known as [Hab03]:

$$G(\mathbf{x}, \mathbf{y}) = \frac{1}{4\pi} \frac{e^{-\sigma_e \|\mathbf{x} - \mathbf{y}\|}}{\|\mathbf{x} - \mathbf{y}\|} \quad (2.19)$$

f^0		first angular moment of scalar function $f(\vec{\omega})$
f^1		second angular moment of scalar function $f(\vec{\omega})$
L^{in}		total inscattered radiance
L_d		diffuse radiance
L_{ri}		reduced intensity
ϕ	L_d^0	scalar irradiance, radiant fluence
\vec{E}	\vec{L}_d^1	vector irradiance, radiant flux vector
Q		volumetric source
Q_+	$Q + L_{ri}^{in}$	total source term
p		phase function
g		mean cosine of the average scattering angle
σ_a		absorption coefficient
σ_s		scattering coefficient
σ_t	$\sigma_a + \sigma_s$	extinction coefficient
α	σ_s/σ_t	scattering albedo
σ'_s	$(1 - g)\sigma_s$	reduced scattering coefficient
σ'_t	$\sigma_a + \sigma'_s$	reduced extinction coefficient
α'	σ'_s/σ'_t	reduced scattering albedo
σ_e	$\sqrt{3\sigma_a\sigma'_t}$	effective transport coefficient
l_{mfp}	$1/\sigma_t$	mean free path length
l'_{mfp}	$1/\sigma'_t$	reduced mean free path length
l_d	$1/\sigma_e$	diffusion length
D	$1/(3\sigma'_t)$	diffusion coefficient
A	$(1 + F_{dr})/(1 - F_{dr})$	index mismatch factor
F_{dr}		diffuse Fresnel reflectance
$S(\mathbf{x})$		source term; RHS of diffusion equation

Table 2.2: Notation summary for the diffusion approximation.

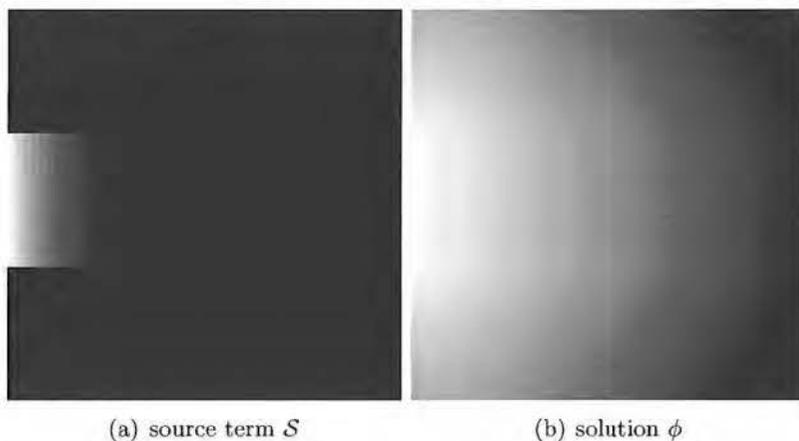


Figure 2.3: Simple example of light diffusion in a 2D square. The left image (a) shows the source term due to external illumination (collimated beam). The right image (b) shows the corresponding diffusion result.

Intuitively, G represents the “scattering response” to an isotropic light source (i.e. the impulse) in an infinite homogeneous medium. The magnitude of this response is measured by the diffusion length $l_d := 1/\sigma_e$, and indicates how far diffusion reaches.

Due to the linearity of L , we can rapidly find the solution for an arbitrary source term $\mathcal{S}(\mathbf{x})$:

$$\phi(\mathbf{x}) = \int_{\Omega} \mathcal{S}(\mathbf{y})G(\mathbf{x}, \mathbf{y})dV_{\mathbf{y}}, \quad (2.20)$$

which can be interpreted as a convolution on \mathcal{S} with kernel G .

The Green’s function plays an important role the construction of the fast dipole approximation (chapter 3), and the boundary diffusion model (chapter 5).

Accounting for Boundaries and Heterogeneities

One can rapidly establish a solution in an infinite homogeneous medium by simply convolving the source term. However, in practice, the difficulty lies in accounting for the presence of boundaries. For very simple geometry, like an infinite plane, analytical solutions are available (see chapter 3). For arbitrarily shaped objects, or when material coefficients vary spatially, one has to resort to numerical simulation. Solving PDEs like the diffusion equation (2.10) is

a problem frequently encountered in engineering and science [PTVF92]. The canonical approach is discretization. In particular, finite differencing, finite element methods and boundary element methods are all applicable [HP02]. After discretization, the problem boils down to solving a linear system of equations. In chapter 4, we will propose a solution based on finite differencing.

For illustration purposes, figure 2.3 shows a simple 2D example of light diffusion.

Final Display

Observe a differential patch dA with normal n . We see that one can express radiosity in terms of the net flux through this patch:

$$M(\mathbf{x}) = \int_{4\pi} L_d(\mathbf{x}, \vec{\omega}) d_{\perp}\omega = \vec{n} \cdot \vec{E}(\mathbf{x}) \quad (2.21)$$

If a solution is available in terms of ϕ , we can reconstruct radiosity at the surface using equation (2.7), or more easily by employing the boundary condition (2.14):

$$M = \frac{1}{2A}\phi \quad (2.22)$$

2.4 Bidirectional Distribution Functions

Traditionally in computer graphics, reflection is described by the bidirectional reflectance distribution function (BRDF) [NRH⁺77]:

$$f_r(\vec{\omega}_i, \vec{\omega}_o) = \frac{dL(\mathbf{x}, \vec{\omega}_o)}{dE(\mathbf{x})} = \frac{dL(\mathbf{x}, \vec{\omega}_o)}{L(\mathbf{x}, \vec{\omega}_i) d_{\perp}\omega_i}, \quad (2.23)$$

Rearranging equation (2.23) and integrating over the hemisphere, yields the expression for outgoing radiance at a surface:

$$L(\mathbf{x}, \vec{\omega}_o) = \int_{\Omega} f_r(\vec{\omega}_i, \vec{\omega}_o) L(\mathbf{x}, \vec{\omega}_i) d_{\perp}\omega_i$$

The BRDF relates incident to outgoing light at a surface location \mathbf{x} . It has two important properties [DBB03]:

1. Conservation of energy: $\int_{2\pi} f_r(\vec{\omega}_i, \vec{\omega}_o) d_{\perp}\omega_i \leq 1$. In other words, the total amount of reflected flux should be less than the total incoming flux.
2. Reciprocity: $f_r(\vec{\omega}_i, \vec{\omega}_o) = f_r(\vec{\omega}_o, \vec{\omega}_i)$ (a.k.a. Helmholtz reciprocity). Intuitively, f_r is not affected by reversing the direction of light.

The BRDF serves as an abstraction mechanism for reflection at arbitrary surfaces. In practice, it can be expressed analytically by various models (see Dutré et al. [DBB03] for an overview). Alternatively, it can be measured and tabulated (see Matusik et al. [MPBM03] and references therein).

In transparent materials like glass and water, the effect of subsurface scattering is negligible. Transmission can then be treated using conventional techniques (i.e. ray tracing along the refracted direction [Whi80]). For non-transparent media, if the scattering response remains very local, contribution from subsurface scattering can be modeled as a BRDF (see literature review about 1D subsurface scattering models; section 1.3). For the remaining cases, subsurface scattering is not negligible, and should be described by a more comprehensive model.

The bidirectional surface scattering reflectance distribution function (BSSRDF) generalizes the BRDF by introducing dependency on incoming and outgoing surface location [NRH⁺77]:

$$S(\mathbf{x}_i, \vec{\omega}_i, \mathbf{x}_o, \vec{\omega}_o) = \frac{dL(\mathbf{x}_o, \vec{\omega}_o)}{d\Phi(\mathbf{x}_i)} = \frac{d^2L(\mathbf{x}_o, \vec{\omega}_o)}{L(\mathbf{x}, \vec{\omega}_i) d_{\perp} \omega_i dA_i} \quad (2.24)$$

Note that the dimensionality is increased by 4, w.r.t. the BRDF.

Rearranging equation (2.24) and integrating over the hemisphere and the surface Γ yields:

$$L(\mathbf{x}_o, \vec{\omega}_o) = \int_{\Gamma} \int_{2\pi^+} S(\mathbf{x}_i, \vec{\omega}_i, \mathbf{x}_o, \vec{\omega}_o) L(\mathbf{x}_i, \vec{\omega}_i) d_{\perp} \omega_i dA_i \quad (2.25)$$

The BSSRDF abstracts the effect of radiative transfer inside a translucent object. Similarly to BRDFs, it obeys:

1. Conservation of energy:

$$\int_{\Gamma} \int_{2\pi} S(\mathbf{x}_i, \vec{\omega}_i, \mathbf{x}_o, \vec{\omega}_o) d_{\perp} \omega_i dA_i \leq 1 \quad (2.26)$$

2. Reciprocity:

$$S(\mathbf{x}_i, \vec{\omega}_i, \mathbf{x}_o, \vec{\omega}_o) = S(\mathbf{x}_o, \vec{\omega}_o, \mathbf{x}_i, \vec{\omega}_i) \quad (2.27)$$

The second property follows from the fact that radiative transfer is reciprocal w.r.t. $(\mathbf{x}_i, \vec{\omega}_i)$ and $(\mathbf{x}_o, \vec{\omega}_o)$, if the index of refraction at x_i and x_o is the same [Aro97].

Compared to the literature on BRDFs, little has been done on BSSRDF modeling. So far, Jensen et al. [JMLH01a] are the only to have introduced

an analytical BSSRDF model, which consists of a single scattering term and a diffusion term (see chapter 3). Gösele proposed a data-driven model, based on the work by Lensch et al. [LGB⁺02].

Once a BSSRDF model is available, rendering simply boils down to evaluating the integration in equation (2.24).

2.5 Conclusion

In this chapter, mathematical models were introduced for the purpose of simulating light transport in translucent objects. Most importantly, for highly scattering translucent materials, light propagation can be considered as a diffusion process, and described mathematically by a simple linear PDE (2.10).

In the remainder of this dissertation, we will introduce novel efficient algorithms for solving the diffusion equation.

Chapter 3

Fast Dipole Approximation

One of the goals stated in this dissertation, is rendering of translucent objects at interactive rates. The radiative transfer equation (RTE) is vastly complex (section 2.2), and therefore less suitable as an underlying rendering model for translucent objects. The diffusion equation is more appropriate. However, direct solutions using discretization are still far from interactive, as they inherently require costly matrix inversion [Sta95].

The fast dipole approximation (FDA), introduced by Jensen et al. [JMLH01a], makes further simplifications to subsurface diffusion (fig. 3.1), resulting in a simple linear model free of matrix inversion. It can be expressed analytically as in the form of a Bidirectional Surface Scattering Reflectance Distribution Function (BSSRDF, see section 2.4).

As will be shown in this chapter, the FDA is simple enough to reach the goal of interactivity. Using this model, outgoing radiance is expressed simply as an integral over the object's surface. We introduce novel algorithms which can efficiently compute this integral at interactive rates, allowing for dynamic adjustment of viewpoint, illumination, geometry and material properties. A thorough comparison between our methods and related work will be presented later in this chapter. To conclude, we also discuss the accuracy of the fast dipole model.

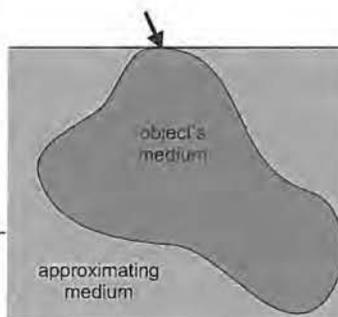


Figure 3.1: The fast dipole approximation assumes the geometry is locally infinitely planar where light impinges (indicated by arrow).

3.1 Dipole Model

In a semi-infinite medium, diffusion can be solved analytically, albeit mild simplifications which will be discussed shortly [EVNT78, FPW92]. Jensen et al. [JMLH01a] proposed to apply this solution to arbitrary geometry, thereby avoiding costly computational procedures such as Monte Carlo simulation and matrix inversion.

We will briefly outline the FDA derivation and the associated BSSRDF model, and discuss how it can be used for arbitrary geometry. See table 2.2 for notation concerning the diffusion approximation.

3.1.1 Diffusion in a Semi-Infinite Medium

Consider a collimated beam of light with flux Φ , perpendicularly impinging on the semi-infinite medium. We can accurately construct the corresponding source term \mathcal{S} by placing an infinite number of exponentially decaying isotropic sources along the normal in the medium. In order to simplify computations, Farrell et al. [FPW92] proposed to only account for the first scattering event using a single isotropic source. It carries a power Φ , and is placed at a distance z_r beneath the surface, equal to the reduced mean free path length l'_{mfp} . Employing a single isotropic source is justified by observing that in optically thick media, reduced intensity decreases rapidly, and the effect of diffusion destroys most directional dependence. The resulting source function is [FPW92]:

$$\mathcal{S}(\mathbf{x}) = \alpha' \Phi \delta_{z_r}(\mathbf{x})$$

with $\alpha' = \sigma'_s / \sigma'_t$, and the corresponding Dirac impulse δ_{z_r} . In an infinite medium, the solution is simply the Green's function (2.19). In a semi-infinite medium, the diffusion boundary condition (eq. 2.18) needs to be satisfied over the infinite planar interface. It can be satisfied approximately by forcing ϕ to zero at an imaginary plane at a distance $2AD$ above the interface, called the *extrapolated boundary* [FPW92, MSS88] (fig. 3.2). We can now apply the method of images [Hab03]: a second source is added as a mirrored version of the first, w.r.t. the extrapolated boundary [EVNT78, FPW92]. The second source has power $-\Phi$, thereby forcing the fluence to zero at the extrapolated boundary. The combination of the scattering source and virtual source (dipole), expresses the solution simply as:

$$\phi(x) = \alpha' \Phi \frac{1}{4\pi D} \left[\frac{e^{-\sigma_e d_r}}{d_r} - \frac{e^{-\sigma_e d_v}}{d_v} \right]$$

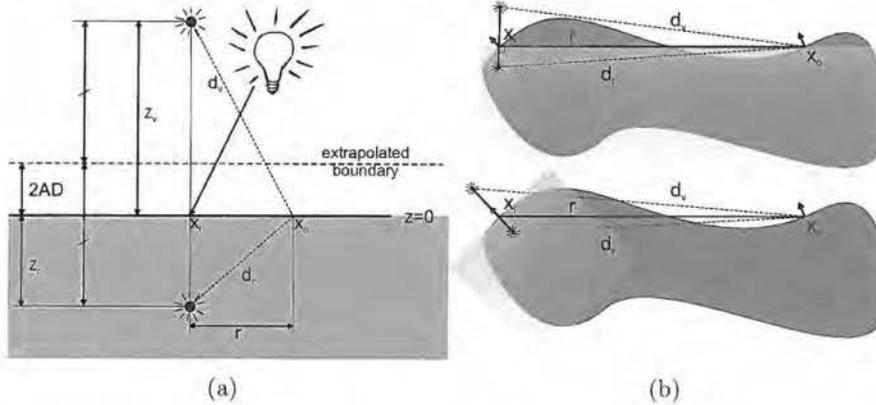


Figure 3.2: (a) Illustration of fast dipole approximation. See text for detailed explanation. (b) Dipole placement. Upper image: “rotated dipoles”; the dipole axis is oriented perpendicularly w.r.t. to vector $\mathbf{x}_o - \mathbf{x}_i$. Lower image: normal-aligned dipole axis; the dipoles are placed along the surface normal at x_i . The gray overlay indicates the semi-infinite approximating medium (also see figure 3.1)

with d_r and d_v the distance to the real and virtual source, respectively. Note that the latter is placed at height $z_v = z_r + 4AD$ above the surface. See figure 3.2 for an illustration.

3.1.2 Dipole BSSRDF

Now we are ready to derive the actual BSSRDF. As was shown in the previous section, we will represent external illumination by an isotropic source in the medium, near the surface. For each incoming ray of light, we can place a source along the corresponding refracted direction. However, in order to force reciprocity for the BSSRDF, Jensen et al. [JMLH01a] proposed to fix the source directly below the surface, along the normal direction.

We rewrite (2.9) and drop the source term, since incident illumination is represented by an isotropic source:

$$\vec{E}(\mathbf{x}) = -D\vec{\nabla}\phi(\mathbf{x})$$

This identity is known as Fick’s law [Fic55], and enables us to express \vec{E} as the fluence’s gradient. Intuitively, it states that light flows away from higher photon densities. For a differential contribution on the surface at location \mathbf{x}_i ,

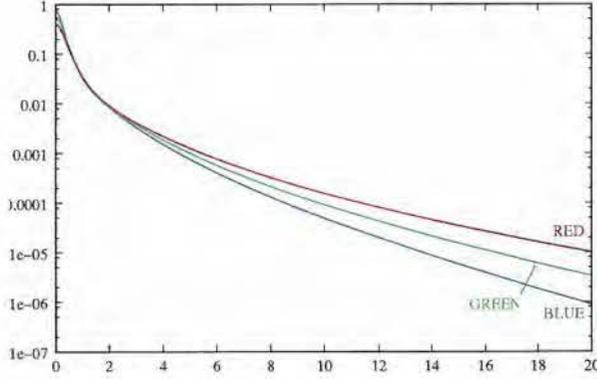


Figure 3.3: Radial plot of the dipole BSSRDF (R_d) for marble. Observe the steep exponential falloff and wavelength dependence.

the outgoing radiance at \mathbf{x}_o is measured by (using eq. 2.21):

$$dM(\mathbf{x}_o) = -D \frac{\partial \phi(\mathbf{x}_i)}{\partial n_o} \quad (3.1)$$

$$= -\frac{1}{4\pi} d\Phi(\mathbf{x}_i) \frac{\partial}{\partial n_o} \left[\frac{e^{-\sigma_e d_r}}{d_r} - \frac{e^{-\sigma_e d_v}}{d_v} \right] \quad (3.2)$$

Rearranging and differentiating gives the diffuse BSSRDF $R_d = \frac{dM(\mathbf{x}_o)}{d\Phi(\mathbf{x}_i)}$:

$$R_d(\mathbf{x}_i, \mathbf{x}_o) = \frac{\alpha'}{4\pi} \left[z_r (1 + \sigma_e d_r) \frac{e^{-\sigma_e d_r}}{d_r^3} + z_v (1 + \sigma_e d_v) \frac{e^{-\sigma_e d_v}}{d_v^3} \right]$$

We construct the general BSSRDF by adding Fresnel coefficients, under the assumption of having a perfectly smooth interface:

$$S(\mathbf{x}_i, \vec{\omega}_i, \mathbf{x}_o, \vec{\omega}_o) = \frac{1}{\pi} F(\eta, \vec{\omega}_i) R_d(\|\mathbf{x}_i - \mathbf{x}_o\|) F(\eta, \vec{\omega}_o) \quad (3.3)$$

A plot of R_d is shown in figure 3.1.2.

Jensen et al. [JMLH01a] also add a single scattering term to the BSSRDF. We will ignore it, as we are only interested in subsurface diffusion.

3.1.3 Diffusion for Arbitrary Geometry

The main approximation behind the FDA, is to apply the previously presented solution for arbitrary geometry (fig. 3.1). One simply employs the 3D distance $\|\mathbf{x}_i - \mathbf{x}_o\|$ to evaluate R_d in equation (3.3). Under the assumption

$$\begin{aligned}
S(\mathbf{x}_i, \vec{\omega}_i, \mathbf{x}_o, \vec{\omega}_o) &= \frac{1}{\pi} F_t(\eta, \vec{\omega}_i) R_d(r) F_t(\eta, \vec{\omega}_o) \\
R_d(r) &= \frac{\alpha'}{4\pi} \left[z_r (1 + \sigma_e d_r) \frac{e^{-\sigma_e d_r}}{d_r^3} + z_v (1 + \sigma_e d_v) \frac{e^{-\sigma_e d_v}}{d_v^3} \right] \\
z_r &= 1/\sigma'_t \\
z_v &= z_r + 4AD \\
d_r &= \sqrt{r^2 + z_r^2} \\
d_v &= \sqrt{r^2 + z_v^2} \\
r &= \|\mathbf{x}_i - \mathbf{x}_o\|
\end{aligned}$$

Table 3.1: Summary of the dipole BSSRDF model.

that geometry remains near-planar w.r.t. the scattering response, this is a sound approximation (e.g. human skin). Accuracy reduces as materials scatter more and further. Yet, the FDA is able to produce plausible renderings even for these cases [JB02]. In section 3.6, we will discuss the impact of this assumption.

The dipole BSSRDF loses its reciprocity property (eq. 2.27) when applied to arbitrary geometry. However, reciprocity can still be enforced by “rotating” the dipole axis perpendicular to vector $\mathbf{x}_o - \mathbf{x}_i$ (see figure 3.2). This scheme has two additional advantages. First, it can be computed easily using Pythagoras’ rule:

$$d_r = \sqrt{r^2 + z_r^2} \quad \text{and} \quad d_v = \sqrt{r^2 + z_v^2}, \quad (3.4)$$

thereby reducing the dimensionality of R_d to a single geometric parameter, namely $r = \|\mathbf{x}_o - \mathbf{x}_i\|$. Also, it avoids spurious high responses in corners (e.g. near edges in a cube), which poses a problem for normal-aligned dipoles [JMLH01a].

Rotating the dipoles may seem like a crude approximation. However, applying the solution for semi-infinite media is already incorrect in the first place, so it might be argued that the dipole axis’ orientation may be chosen arbitrarily. An additional argument in favor of rotated dipoles, is that the virtual geometry of the semi-infinite medium (i.e. the infinite plane) contains both x_i and \mathbf{x}_o , instead of only \mathbf{x}_i .

Later, we will exploit the reciprocity property to construct an efficient integration scheme (see section 3.3).

Table 3.1 summarizes the analytical expression for the dipole BSSRDF.

3.2 Efficient Rendering

The problem of evaluating the fast dipole model boils down to solving a surface integral (eq. 2.25). Filling in the dipole BSSRDF, we can express it as:

$$L(\mathbf{x}_o, \vec{\omega}_o) = \frac{1}{\pi} F_t(\eta, \vec{\omega}_o) M(\mathbf{x}_o) \quad (3.5)$$

with

$$M(\mathbf{x}_o) = \frac{1}{\pi} \int_{\Gamma} R_d(\|\mathbf{x}_i - \mathbf{x}_o\|) E_{\eta}(\mathbf{x}_i) d\mathbf{A}_i \quad (3.6)$$

where we have the transmitted irradiance (shortened to “irradiance” in the remainder of this chapter):

$$E_{\eta}(\mathbf{x}_i) = \int_{2\pi^+} F_t(\eta, \vec{\omega}) L(\mathbf{x}_i, \vec{\omega}) d_{\perp}\omega \quad (3.7)$$

In the next section, we will describe the most straightforward approach to solve our integration problem: uniform point sampling.

Point Sampling

Jensen et al. [JB02] proposed a simple Monte Carlo integration scheme [KW86] to efficiently solve equation (3.6). A set of uniform samples $\mathbf{p}_1, \dots, \mathbf{p}_N$ is distributed across Γ , and the irradiance E_{η} is computed at each sample. The uniform distribution can be obtained via the point relaxation algorithm introduced by Turk et al. [Tur91]. Next, $M(\mathbf{x}_o)$ is estimated by simply summing contributions from every point sample:

$$M(\mathbf{x}_o) = \frac{A_{\Gamma}}{N} \sum_{j=1}^N R_d(\|\mathbf{x}_o - \mathbf{p}_j\|) E_{\eta}(\mathbf{p}_j)$$

Motivated by the exponential decaying character of R_d , the accumulation can be implemented efficiently using a hierarchical scheme, in the same spirit as many-body simulations [App85]. All point samples are organized in an octree, so that distant points can be clustered into a single sample. The method is applicable to different types of surface representation (e.g. polygonal, implicit), due to its meshless nature, and is capable of generating high quality renderings.

In practice however, this integration scheme turns out to be costly. The point samples should be spaced at a distance proportional to $1/\sigma'_t$, the mean

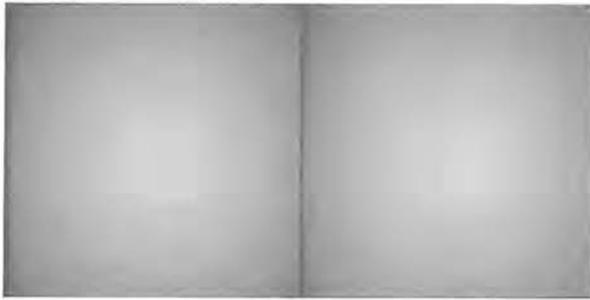


Figure 3.4: Renderings of a marble square slab of size $80 \times 80 \text{ mm}^2$ and $160 \times 160 \text{ mm}^2$ (84K and 337K samples respectively) using point sampling. Samples were placed using Turk’s relaxation algorithm [Tur91]. Low frequency noise appears in the reconstruction, since the points can only be distributed approximately uniform.

free path length of a photon [JB02]. Therefore, the total number of required samples is:

$$N \propto \frac{1}{\pi} A \sigma_t'^2 \quad (3.8)$$

As the object increases linearly in size, the number of required samples increases quadratically. For instance, a marble cube of with edges of length 1 cm , 10 cm and 1 m would require a multiple of 600, 60K and 6M samples, respectively. This behavior is likely to compromise performance in interactive settings.

The density criterion (eq. 3.8) guarantees that the relative error due to the discretization of equation (3.6) is constant w.r.t. scale. Reconstruction should therefore always be smooth. In practice however, samples can only be placed approximately uniform over the surface. As can be observed from figure 3.2, point sampling is very sensitive to deviations in the distribution, resulting in objectionable low frequency noise.

From a practical point of view, point sampling becomes less suitable in an interactive setting, in particular when material parameters and geometry are dynamic. In this case, the sample set should be re-distributed over the surface, and the accompanying hierarchy should be reconstructed (or updated), which is a significant overhead. Note that the relaxation for sample placement per se is already a costly operation [Tur91]. Doing the integration on-the-fly would be more efficient.

Improved Integration Strategy

The main improvement to achieve interactivity, is the introduction of alternative integration techniques which avoid costly uniform point sampling, and the overhead associated with it. We apply two different approaches:

- **Boundary Discretization** We discretize the boundary into elements (triangles), and assume irradiance is constant over each element. As such, the surface integral can be split up into integrals over each element, having only R_d in the integrand. By applying a semi-analytical quadrature, these integrals can be solved at run-time. We also introduce a hierarchy of elements in order to speed up computation significantly.
- **Importance Sampling** We derive an exact importance sampling scheme for R_d , in order to efficiently apply Monte Carlo quadrature. Under the assumption that the influence of subsurface diffusion remains local (e.g. for human skin), the quadrature can be implemented efficiently on graphics hardware. In particular, we represent irradiance in image space, and apply the importance sampling here, essentially acting as a filter.

After having detailed these two approaches, we compare them to methods described in related work.

3.3 Hierarchical Boundary Element Method

In this section, we outline a hierarchical boundary element method to solve equation (3.6) efficiently, in spirit of hierarchical radiosity with clustering [HSA91, SAG94, Sil95, WHG99].

If we break the surfaces of our object to be rendered into regions $\Gamma = \bigcup_k \Gamma_k$, and if we assume the irradiance $E_k = E_\eta(\mathbf{x}_k)$ is constant over Γ_k , equation (3.6) is reduced to a weighted sum:

$$M(x_o) = \sum_k E_k F(\Gamma_k, \mathbf{x}_o) \quad (3.9)$$

$$F(\Gamma_k, \mathbf{x}_o) = \int_{\Gamma_k} R_d(\|\mathbf{x}_o - \mathbf{x}_i\|) dA_i \quad (3.10)$$

We compute the weights $F(\Gamma_k, \mathbf{x}_o)$ for the midpoint of each element w.r.t. all other elements. The irradiance E_k is also evaluated at the midpoints. The sum (3.9) can then be used to evaluate radiosity at each element's midpoint and assumed constant over the whole element. In our implementation, we

average the per-element radiosity values at the vertices. To obtain a smooth solution, these per-vertex values are interpolated over the elements.

The factors $F(\Gamma_k, \mathbf{x}_o)$ are similar to point-to-patch *form factors* in the radiosity method [CW93]. There, the form factor usually has a purely geometric meaning, whereas our form factor also encodes the material properties, and does not include visibility. It essentially converts per-element irradiance to the corresponding radiosity increment at a point on the surface, as a result of subsurface scattering.

In the next section, we will propose an efficient quadrature to compute the form factor $F(\mathbf{A}, \mathbf{d})$. In section 3.3.2, we introduce an efficient hierarchical algorithm to solve equation (3.9).

3.3.1 Form Factor Computation

Using the rotated dipole placement scheme, we can apply the reciprocity principle for computing the integral in equation (3.10): instead of using dipoles in element \mathbf{A} , we place a single dipole at the evaluation point d . Let h be the perpendicular distance of d to the supporting plane τ of \mathbf{A} (see figure 3.5). Let z be the “dipole offset” which equals either z_r or z_v (see BSSRDF model, table 3.1), and d_z the location of the dipole source associated with d and z . From h and using equation (3.4), we can compute the distance h_z of the projected evaluation point p' onto τ , to d_z : $h_z^2 = h^2 + z^2$.

As both source terms in R_d are almost identical, we will focus on the integral of the contribution of a single source point d_z of the dipole:

$$I = \int_{\mathbf{A}} z(1 + \sigma s) \frac{e^{-\sigma s}}{s^3} dA$$

s denotes the distance between the integration point \mathbf{x} on the polygon \mathbf{A} to the considered dipole source d_z . We first assume that \mathbf{A} is a triangle. We shall see that our result straightforwardly extends to the case of arbitrary planar polygons. Figure 3.5 and table 3.2 illustrate and explain the symbols used in our derivation.

We solve the integral in polar coordinates in \mathbf{A} 's supporting plane, using the orthogonal projection of d onto this plane, called d' , as the pole:

$$I = \int_{\theta_{min}}^{\theta_{max}} \int_{r_{min}(\theta)}^{r_{max}(\theta)} z(1 + \sigma s) \frac{e^{-\sigma s}}{s^3} r dr d\theta \quad (3.11)$$

Consider, to start with, the case that one of \mathbf{A} 's vertices coincides with \mathbf{d}' so that $r_{min}(\theta) = 0$ for all polar angles θ . Change of integration variable r to

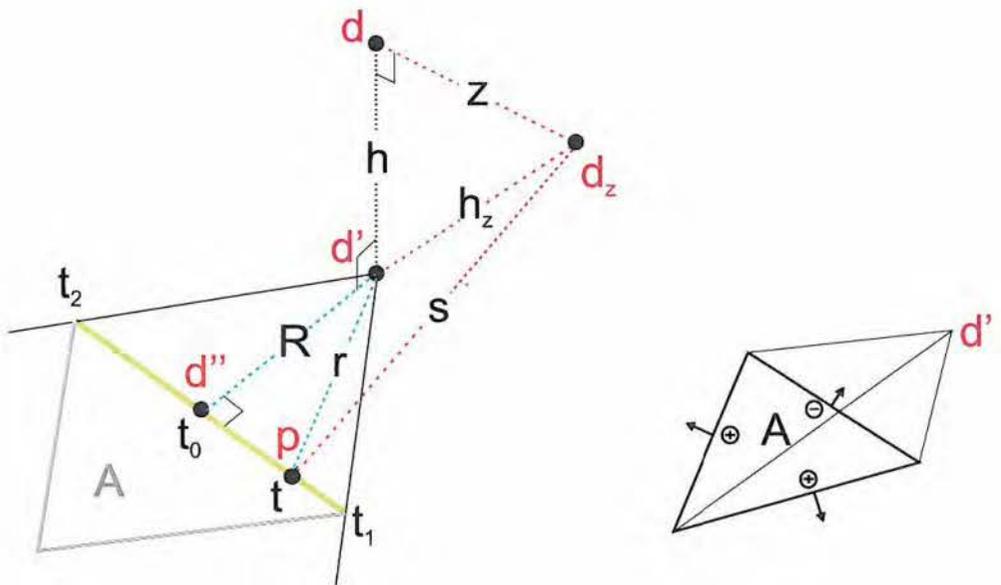


Figure 3.5: **Left:** depiction of geometric variables for deriving the form factor (situation for integrating over one edge, shown in green). Position vectors are indicated in red. **Right:** integration over the triangle is performed by integrating along the edges. The contribution of front facing edges is subtracted from the contribution of back facing edges.

τ	supporting plane of triangle
\mathbf{d}	evaluation point
z	dipole offset from \mathbf{d} (equals either z_r or z_v)
\mathbf{d}_z	location of dipole associated with d
\mathbf{d}'	\mathbf{d} orthogonally projected on τ
\mathbf{d}''	\mathbf{d}' orthogonally projected on edge
R	$\ \mathbf{d}' - \mathbf{d}''\ $
S	$\ \mathbf{d} - \mathbf{d}''\ $
h	$\ \mathbf{d} - \mathbf{d}'\ $, height of d w.r.t. τ
h_z^2	$h^2 + z^2 = \ \mathbf{d}' - \mathbf{d}_z\ ^2$
S^2	$R^2 + h_z^2$
t	parameter on edge, $t = 0$ corresponds to \mathbf{d}''
\mathbf{p}	point on edge for parameter t
t_0, t_1, t_2	midpoint, start and end of edge, respectively
Δt	$t = t_0 + \Delta t$
r	distance from \mathbf{d}' to \mathbf{p} ; $r^2 = R^2 + t^2$
s	distance from \mathbf{d}_z to \mathbf{p} ; $s^2 = r^2 + h_z^2$
L	edge length, $L = t_2 - t_1$

Table 3.2: Symbols used in the form factor derivation.

the distance $s = \sqrt{r^2 + h_z^2}$, and substitution of $u = \sigma s$, then yields:

$$\begin{aligned}
 I &= \int_{\theta_1}^{\theta_2} \int_{h_z}^{s(\theta)} z(1 + \sigma s) \frac{e^{-\sigma s}}{s^3} s \, ds \, d\theta \\
 &= \int_{\theta_1}^{\theta_2} \left[-\frac{z\sigma}{u} e^{-u} \right]_{\sigma h_z}^{\sigma s(\theta)} d\theta \\
 &= \frac{z}{h_z} e^{-\sigma h_z} (\theta_2 - \theta_1) - z \int_{\theta_1}^{\theta_2} \frac{e^{-\sigma s(\theta)}}{s(\theta)} d\theta. \tag{3.12}
 \end{aligned}$$

$s(\theta)$ denotes the distance from \mathbf{d}_z to points on the triangle edge opposite to \mathbf{d}' . Our area integral thus has been reduced to an integral over one triangle edge.

For a general triangle, the sum of three such expressions is obtained: one for each edge of the triangle. The contribution of back facing edges (furthest away from \mathbf{d}') is counted positive, while the contribution of front facing edges is subtracted (see figure 3.5).

For any closed polygon, the first terms in equation (3.12), called I_1 from now on, cancel if \mathbf{d}' is outside the polygon. If \mathbf{d}' is inside, their sum equals

$2\pi \frac{z}{h} e^{-\sigma h}$.

The second terms, which we call I_2 , are more complicated. Change of integration variable from θ to t , the distance along the edge segment measured w.r.t. the orthogonal projection \mathbf{d}'' of the dipole source point \mathbf{d} onto the edges supporting line, yields:

$$t(\theta) = R \tan \theta \quad \text{and} \quad d\theta = \frac{R dt}{R^2 + t^2} \quad (3.13)$$

$$I_2 = z \int_{\theta_1}^{\theta_2} \frac{e^{-\sigma s}}{s} \frac{R dt}{R^2 + t^2} \quad (3.14)$$

$$= z R \int_{t_1}^{t_2} \frac{1}{\sqrt{S^2 + t^2}} \frac{1}{R^2 + t^2} e^{-\sigma \sqrt{S^2 + t^2}} dt. \quad (3.15)$$

$t_1 = R \tan(\theta_1)$ and $t_2 = R \tan(\theta_2)$ denote the distances from the edge end-points to d'' . We did not find an analytic solution to this integral as such, but Taylor series expansion w.r.t. the midpoint of the edge (parameter t_0) of the three main factors in the integrand allows to obtain a good approximation:

$$\frac{1}{\sqrt{S^2 + t^2}} = \frac{1}{s_0} \left(1 - \frac{1}{2} \left[\frac{2t_0 \Delta t}{s_0^2} + \left(\frac{\Delta t}{s_0} \right)^2 \right] + \dots \right) \quad (3.16)$$

$$\frac{1}{R^2 + t^2} = \frac{1}{r_0^2} \left(1 - \left[\frac{2t_0 \Delta t}{r_0^2} + \left(\frac{\Delta t}{r_0} \right)^2 \right] + \dots \right) \quad (3.17)$$

$$e^{-\sigma s} = e^{-\sigma s_0} \left(1 - \sigma s_0 \left[\frac{2t_0 \Delta t}{s_0^2} + \left(\frac{\Delta t}{s_0} \right)^2 \right] + \dots \right) \quad (3.18)$$

Here, $s_0 := s(t_0)$ and $r_0 := r(t_0)$.

The product of the Taylor approximations results in a simple polynomial and its integration is straightforward:

$$I_2 = \frac{z R e^{-\sigma s_0}}{r_0^2 s_0} L \left(1 + \dots \right). \quad (3.19)$$

The higher order terms can be safely ignored by imposing constraints on the edge length $L = t_2 - t_1$, as will be discussed shortly.

Finally, the contribution of a single edge e due to one of the dipole source points is $I_e = I_1 - I_2$. The form factor of a triangle \mathbf{A} with three edges (a, b, c) and the dipole is:

$$\int_{\mathbf{A}} R_d(\|\mathbf{x}_i - \mathbf{x}_o\|) dA_i = \frac{\alpha'}{4\pi} \left[\pm (I_a^r + I_a^v) \pm (I_b^r + I_b^v) \pm (I_c^r + I_c^v) \right], \quad (3.20)$$

where I^r and I^v denote the contribution from each dipole source point. The sign for a term in the sum is positive if it is a back facing edge for \mathbf{x}_o , otherwise it is negative.

Error Analysis

The error due to ignoring second and higher order terms in the Taylor series expansions (3.16) to (3.18) is straightforward to analyze if one makes sure that the first order term is sufficiently smaller than 1. The series expansions converge rapidly in that case.

Suppose, we take $\frac{2t_0}{s_0} \frac{\Delta t}{s_0} + \left(\frac{\Delta t}{s_0}\right)^2 < \frac{1}{2}$ for the first factor. Inspecting the higher order terms, we then find that the relative error is about 15% maximally. This condition will be satisfied if we ensure that $\frac{|\Delta t|}{s_0} < \frac{1}{5}$.

A sufficient condition for fast convergence of the second factor is $\frac{|\Delta t|}{r_0} < \frac{1}{5}$. This condition is stronger than the previous one since $r_0 \leq s_0 = \sqrt{r_0^2 + h^2}$.

For the last factor we need to take $|\Delta t|\sigma < \frac{1}{5}$. With $L = 2|\Delta t|$, we have the following conditions:

$$\frac{L}{s_0} < \frac{L}{r_0} < \frac{2}{5} \quad \text{and} \quad L < \frac{2}{5\sigma}. \quad (3.21)$$

If these conditions are satisfied, the relative error on the integrand of I_2 is maximally roughly 50%. This apparently high error can only occur near the end-points of the edges. On most part of the edges, the error is much lower and so is the error on the integrals.

Edges not satisfying the above constraints are recursively subdivided until the conditions hold. The error decreases exponentially with subdivision.

Even when ignoring second and higher order terms, first order terms still appear in the result (3.19) for I_2 . Taking into account the same conditions above, it turns out that also the contribution from the first order terms are bounded and can be safely ignored.

The condition $L < \frac{2}{5\sigma}$ is a global one, and can be satisfied by edge subdivision as well.

Figure 3.6 compares the results of our method with Monte Carlo integration. It shows that our method indeed produces accurate results with the above conditions. Using only 12 Monte Carlo samples as done in [JB02], produces rather low accuracy for this setting.

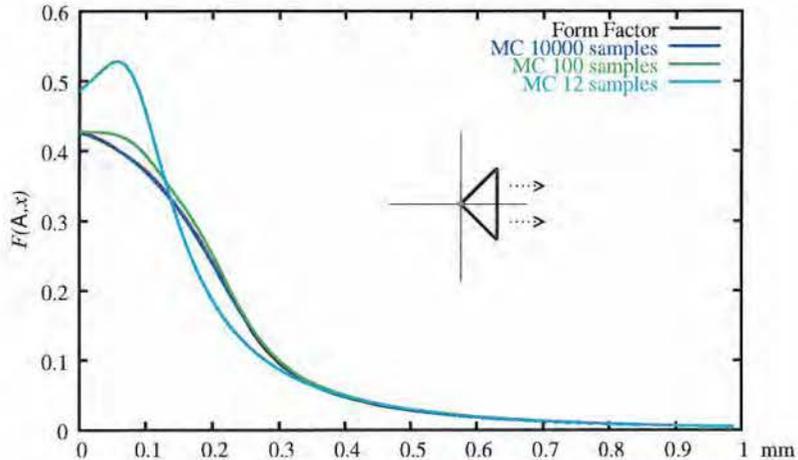


Figure 3.6: This graph shows $F(A, \mathbf{x})$ of a standard triangle A with the tip at $x = (0.1\text{cm}, 0\text{cm})$. The base edge is $L = 1\text{cm}$ and the height is 0.5cm . The x -axis of the graph shows the distance of the triangle's tip to O . The sample point x_o lies also in the origin (planar setting). Skim milk was used as the material. The black curve is evaluated using our method, the colored curves show Monte Carlo evaluations. Our form factor and 10000 Monte Carlo samples result in virtually the same curve.

Discussion

Our integration strategy has several distinct properties. First, it turns out that many of the edge contributions in a polygon mesh will cancel out. Indeed, when integrating over neighboring planar triangles, shared edges are iterated over twice, once per sharing triangle, and in opposite direction. If the triangles receive the same irradiance, the terms $\Upsilon := I_e^r + I_e^v$ for each such edge cancel out. Only the contribution of outer edges remains. For the same reason, the form factor for an arbitrary planar polygon can be obtained by simply summing the appropriately signed contributions Υ for all polygon edges.

Our integration strategy is partly based on analytic integration, and partly on numerical approximation through Taylor series expansion. The level of accuracy can be controlled by loosening or tightening the refinement conditions (Equation 3.21).

In figure 3.7, we show the number of samples that we need to take for different triangles. In this case, a sample corresponds to computing I_2 for a line segment (there can be multiple line segments for an edge, due to subdivision

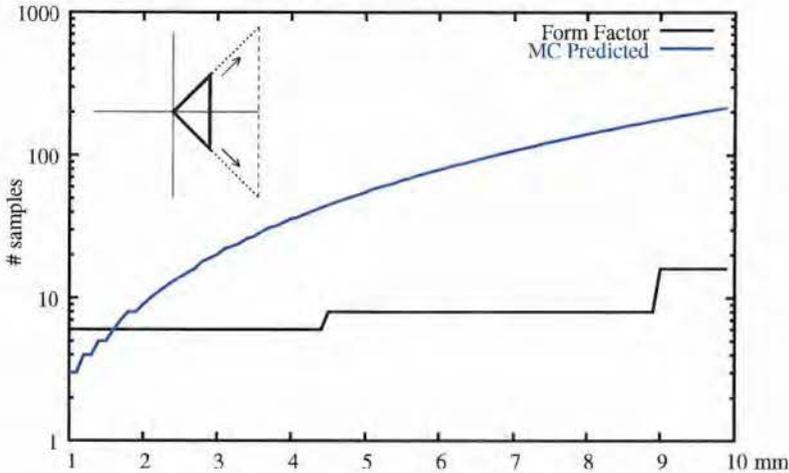


Figure 3.7: This graph compares the number of samples we have to take vs. Jensen and Buhler’s method [JB02] to evaluate the BSSRDF over an incrementally scaled standard triangle. We chose $\mathbf{x}_o = O$. The tip of the triangles lies at the origin as well. The x -axis of the graph shows the distance from the triangle’s base edge to origin. In this example we chose the material coefficients for marble. Although Jensen and Buhler’s method needs less samples for small triangles, their predicted number of samples is too small to achieve acceptable accuracy, see figure 3.6.

in order to meet the conditions). We compare this to the number of samples needed by Jensen and Buhler [JB02]. The efficiency of our integration strategy is evident.

For distant clusters, the form factor will be sufficiently small to be computed with a single point sample at the midpoint. As figure 3.6 indicates, there is nothing to gain by using the form factor in these cases. Due to the steeply descending nature of R_d as distance increases, we see that only for nearby clusters the form factor is appropriate.

The form factor procedure is fast enough to be executed at interactive speeds, and does not require any extra overhead. This enables us to render dynamic geometry (e.g. for animations) and allow for changing the material. Furthermore, our integration method does not suffer from high-frequency noise like Monte Carlo methods [JMLH01b], or low-frequency noise due to under-sampling as seen with point sampling [JB02].

Concerning the latter, we compared an obvious alternative to our form factor algorithm, i.e. a form factor using simple uniform sampling. The amount

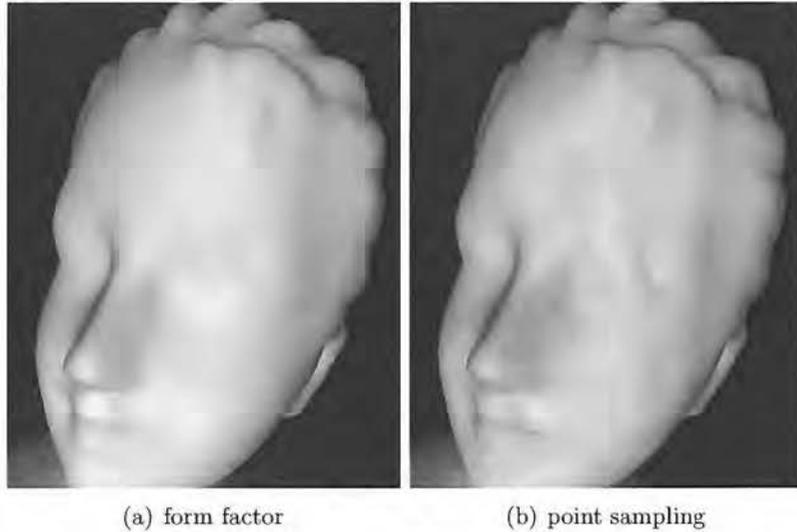


Figure 3.8: Comparison of visual quality with the form factor procedure and point sampling for marble at scale 20cm. Low frequency noise is clearly visible with point sampling.

of samples is chosen according to the point sample distribution criteria used by Jensen et al. [JB02]. It turns out that our form factor procedure is superior for both quality and performance. For moderately translucent materials, R_d is quite steep and more samples are needed in the integration. While making a material less translucent, we noticed a low frequency noise and a serious frame rate drop. See table 3.5 and figure 3.8 for details.

Our form factor algorithm assumes polygons with sufficiently constant irradiance. For sampling from environment maps and point/spot lights, this assumption is acceptable since lighting does not tend to change dramatically. Care must be taken in situations where sharp shadow borders occur, although for highly scattering media this becomes less of a problem due to the diffusing nature of multiple scattering.

It turns out that for all but the nearest elements, a single-sample estimate of the form factor is sufficiently accurate. We decide whether or not to apply this approximation based on the solid angle measure [JB02].

3.3.2 Hierarchical Evaluation

Computing subsurface scattered radiance in this simple way is unfortunately a quadratic procedure in terms of the number of elements. This is too costly for interactive image synthesis, except for the simplest models. A log-linear procedure is obtained by grouping distant elements hierarchically in so called clusters, in a similar way as in hierarchical radiosity with clustering [HSA91, SAG94, Sil95, WHG99].

Overview

Let us first discuss a general overview of our approach.

All elements are organized in a tree hierarchy, such that each node (cluster) represents a collection of elements as a single element. The leaf nodes represent the original elements of which the object is composed.

Rather than computing the form factor between each pair of elements, we first create a candidate *link* between the top level cluster containing all the elements of the object and itself. Next, this link is subjected to a refinement oracle. If the oracle decides that the link would not allow sufficiently accurate integration, the candidate link is refined by opening the cluster at the receiver or emitter side. As such, new candidate links result, which are tested in turn.

After linking, we compute the irradiance at all leaf nodes. The leaf node irradiance is then *pulled up* to the top of the cluster hierarchy by averaging the irradiance on child clusters. We then recursively traverse the cluster hierarchy and *gather* the contributions at each cluster element from over its links. This involves multiplying the irradiance at the emitter element with the form factor (3.10) corresponding to the pair of linked elements and adding the product at the receiver cluster. Clusters always receive contributions at their midpoint.

The contributions gathered at all clusters are then pushed down the hierarchy, i.e. energy received at higher levels in the hierarchy is distributed to the leaf clusters. The accumulated results at the leaf nodes (the elements of the object) are averaged at the vertices, in order to obtain a smooth solution by interpolating the solution using Gouraud shading.

As before, a single sample estimate suffices for most form factors. Form factors with nearby element or cluster elements, which would require more than one sample according to the point density criterion by Jensen et al. [JB02], are integrated using the semi-analytical technique of Section 3.3.1.

Figure 3.9 shows the hierarchy of cluster elements, relevant for calculating subsurface scattered light at the indicated spot. A significant part of the illumination is gathered at a higher level than at the leaf node.

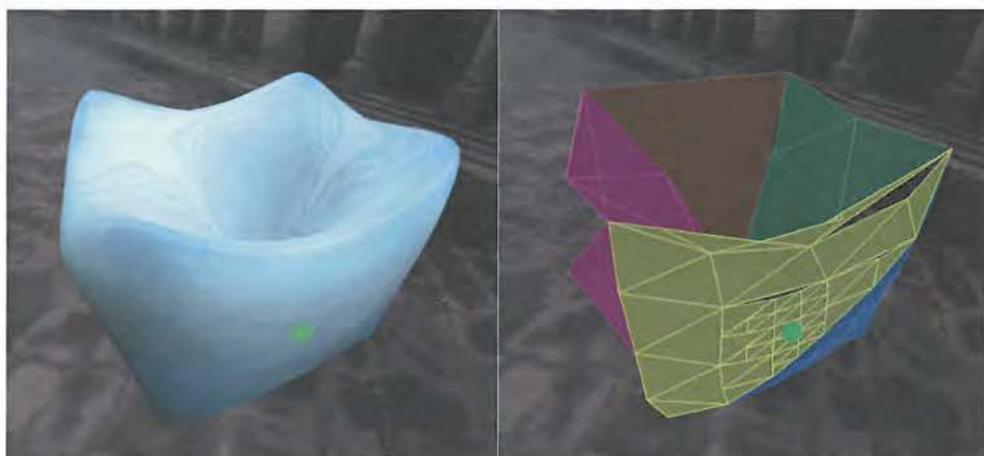


Figure 3.9: Part of the cluster hierarchy relevant for calculating subsurface scattered light at the dot. Each cluster element level is depicted with a different shade. The surrounding triangles are directly linked to the leaf containing the dot, whereas the other, larger triangles are connected through one of the leaf's parents. We see that interactions with distant geometry are handled at higher levels. The cracks between the different levels do not cause visible artifacts as long as the refinement criteria are met.

Implementation

After having presented a high-level overview of our rendering algorithm, we discuss the most important aspects regarding its implementation.

Cluster Hierarchy Our algorithm is given a hierarchical structure such as a multi-resolution mesh, a face cluster hierarchy or a subdivision surface. It is required that finer clusters are fully embedded in coarser levels, in order to consistently integrate over the surface. For instance, approximating subdivision schemes and progressive meshes cannot be applied here.

- **Face Clustering** Face cluster hierarchies have been used in different applications, but they are best known from face cluster radiosity [WHG99]. Such a hierarchy is built by merging neighboring clusters until a single cluster is reached containing all original faces.

We build our hierarchy bottom-up with an algorithm similar to Garland et al. [GWH01]. Unlike their work, our measure only tries to cluster faces as compactly as possible, ignoring surface curvature. Instead we make sure during linking, i.e. at run-time where geometry and curvature may change, that only near-planar clusters are chosen, since our form factor is correct for planar clusters only. To this end we take a simple curvature measure into account during linking in order to reduce the error for the assumption of planar clusters: $\frac{A'}{A}$, where A' and A are the total projected area and total area of the cluster triangles, respectively. The form factor from a cluster is then approximated by projecting all its triangles onto the plane defined by the cluster's midpoint and averaged normal, and working with the resulting polygon.

- **Subdivision** Another method to obtain a hierarchy (and mesh) with the desired properties is surface subdivision. Since the finer clusters have to be fully embedded in the next coarser level, we chose to use subdivision based on 4-to-1 splits. Essentially, subdivision techniques re-mesh the original object such that a suitable hierarchy can be built.

Subdivision surfaces: When only a coarse mesh is given, we can efficiently generate a sufficiently dense mesh, along with a suitable hierarchy, using subdivision surfaces. Any subdivision scheme to be used must be interpolatory to fulfill our hierarchy criteria. For example, the Butterfly algorithm [DLG90, ZSS96] can be used to generate hierarchies fulfilling this criteria. An example can be seen in Figure 3.9.

Shrink-Wrapped: Using a similar algorithm to [KVLS99], we create hierarchies for arbitrary closed meshes. We first project the mesh from its center onto a sphere, and relax it using the “umbrella” operator. We then take a coarse base mesh, whose vertices are shared with the original mesh, and recursively split. Whenever a new vertex is introduced, we project it onto the sphere, find in which triangle from the original mesh it lies, and use the interpolated coordinates from the original mesh as the new coordinates for the new vertex.

The hierarchy generated by subdivision is more efficient than the face clustering method, since every cluster is represented by exactly one triangle. On the other hand, it also requires the object to be re-meshed completely. The butterfly method can only produce simple shapes from a low polygon count base mesh. Although the shrink-wrapping can work on arbitrary meshes, the approach often leads to unevenly tessellated meshes. Overall, it is a compromise between efficiency and quality.

Hierarchical Refinement We use Jensen et al.’s maximum solid angle criteria [JB02] to decide whether or not to link 2 hierarchical elements. It is cheap to evaluate, and has proved to work well in practice. Let E and R be the emitting and receiving element, respectively. The refinement strategy can be summarized as:

- When the maximum deviation of the solid angle ω from R ’s children’s midpoints to E ’s collection of faces exceeds a certain threshold ϵ_1 , split and move the link down the hierarchy at R . Repeat, until the deviation is below ϵ_1 .
- Before connecting a link to an emitter cluster E , check ω from R ’s midpoint to E ’s faces. If it is above a threshold ϵ_2 , split and move the link down at E . Repeat, until it is below ϵ_2 .

In our experience, the solid angle criterion performed well enough for interactive rendering. More advanced linking criteria which reflect the magnitude of the actual form factor better, may be devised in order to reduce link count.

The refinement procedure may be executed in three different “modes”:

1. **Static Mode.** Only illumination changes are accounted. All links and form factors are precomputed. During rendering, the algorithm iterates over each link to gather contributions.

2. **Incremental Mode.** Lighting, material and geometry can be altered. The algorithm incrementally updates form factors of “tagged” links. In case of geometry or material changes, the corresponding links will be tagged. We use a simple demotion/promotion procedure [FYT94] to maintain consistency in the hierarchy: links which do not meet the conditions are moved one level up or down. Interactivity can be controlled by bounding the number of form factor computations per frame.
3. **On-the-fly Mode.** In each frame we traverse the hierarchy to perform a full evaluation from scratch. This mode avoids the overhead of keeping track of links, resulting in quicker updates. Memory requirements are also reduced significantly. However, the overall frame rate is lower than in interactive mode.

The static and incremental mode resembles the algorithm by Carr et al.[CHH03], except that we can immediately account for geometry/material changes, due to the efficient form factor integration.

Irradiance Sampling Irradiance is computed at each leaf cluster (triangle) at run-time for environment maps and point light sources. In the latter case, shadows can be computed by employing a variant of shadow mapping [Wil78]. In the former case, we sample irradiance from an environment based on its spherical harmonics coefficients. Nine coefficient suffice since the incoming radiance undergoes a cosine-weighted integration over the hemisphere and is thus bandlimited [RH01]. Shadows however cannot be taken into account, unless we assume rigid objects [SKS02]. We also ignore the Fresnel coefficient in equation (3.7) for simplicity. The resulting irradiance value is obtained with a simple dot product. Prefiltered environment [Gre86] maps can also be used, but we did not implement this alternative.

Note that irradiance is only sampled at leaf clusters; higher clusters *pull* the averaged irradiance from their children.

Final Rendering Our rendering algorithm computes radiosity at each vertex, which is passed to a set of hardware shaders. The final rendering includes reflection mapping or phong shading. We also accounted for the Fresnel coefficient in equation 3.5, when rendering reflections from environment maps.

Everything is computed in high dynamic range at floating point precision. A simple gamma curve per pixel is applied to the final shading for the purpose of tone mapping. This enhances the global response which is typically much lower than the local response, and adds a lot to the realism.

3.3.3 Results

We implemented our algorithm in C++ on a dual Intel Xeon 2.4Ghz 2Gb RAM configuration with an ATI Radeon 9700 graphics board.

Table 3.3 shows that our algorithm behaves roughly log-linear in the number of triangles.

T	L	$\frac{L}{T}$	on-the-fly	incr.	static
2016	89K	44.1	20	8	5
8160	388K	47.5	47	23	13
32736	1744K	53.3	188	68	43
131736	6905K	52.7	722	351	199

Table 3.3: Model complexity versus link count and running time (ms). Looking at the links-per-triangle ratio, we notice that there is roughly a log-linear correlation between the number of triangles T and the number of links L . The experiment was done for a marble butterfly subdivision model whose bounding box was scaled to 2.5cm.

In table 3.4 we illustrate the performance of our approach with different models. We see that the number of links is mesh-dependent. In this experiment we choose very conservative thresholds for the refinement oracle such that the quality of the solution is guaranteed for every situation.

model	type	T	L	on-the-fly
horse	FC	8764	992K	97
bust	FC	10518	1258K	102
elk	FC	11384	1743K	165
candle	S	4474	435K	41
cube	S	16380	1572K	154

Table 3.4: Overview of performance with different models. Timings for *on-the-fly* mode are in ms. Material was marble scaled at 10cm. The hierarchy types are indicated by: (F)ace (C)lustering or (S)ubdivision.

Table 3.5 illustrates the efficiency of the form factor compared to point sampling. We see that it behaves more robustly in rendering time when the local response gains importance.

We will now discuss some results depicted in figure 3.10. All screenshots were taken interactively in *on-the-fly* mode, at a frame rate ranging from 3 to 15fps, roughly. Except for figure 3.10.5, which is rendered in *incremental*

scale (m)	form factor (ms)	point sampling (ms)
.05	160	162
.1	160	168
.2	161	222
.4	169	501
.8	340	1778

Table 3.5: Comparison of the form factor procedure and point sampling on marble. The object's bounding box is scaled at different sizes. The duration of a full *on-the-fly* evaluation is measured.

mode at 7.5fps.

In figure 3.10.1-2-3 a horse is rendered with three different materials: skim milk, ketchup and candle wax, respectively. Notice the color shifts across the model in figure 3.10.1. The scattering of light is very obvious for the thin geometric features in figure 3.10.3 when the model is lit from behind.

In figure 3.10.4, the scale of whole milk changes from 1.0m, 10cm, 5cm, 2cm, 1cm, to 5mm, respectively. The hierarchy for the tweety model here was generated using shrink wrapping. In the leftmost image, the shading is practically the same as with simple diffuse reflection. The awkward appearance in regions where normals are orthogonal to the direction of incoming light, is due to (gamma curve) tone mapping.

We interactively added a bump to a subdivided cube in figure figure 3.10.5. The resulting model is lit with a spot light, and enhanced with shadow mapping. Notice how the shadow cast by the bump is 'leaking' further on the adjacent plane. For that same cube we applied a Perlin noise [Per85] distortion (figure 3.10.5): 3D noise sampled at each vertex and perturbs its position along the normal. Each frame we increment the offset to the noise's sampling position, resulting in a full deformation on 16K triangles. This situation is rendered interactively at roughly 4-5 frames per second, including irradiance sampling (spot light and shadows).

Figure 3.10.7a depicts how a simple subdivision shape is used to simulate the appearance of a candle. It is lit from inside by a moving and flickering point light source. When we apply a simple 'twist' deformation, the model gets thinner, resulting in more light passing through from the inside (figure 3.10.7b). As this is a very simple model of nearly 5K triangles, this experiment runs in real-time (15fps).

We show the toy elk model to demonstrate that models are not restricted to be of genus 0. The spherical wheel has been dented interactively, resulting

in more scattered light passing through.

Figure 3.10.9a shows a visualization of the clusters used for shading a triangle on the ear on the left. Interactions at the different levels in the hierarchy are colored uniquely. Right next to it (3.10.9b) we show the actual rendering.

The appearance of a marble bust in figure 3.10.10 has been enriched with Fresnel reflection from an environment map.

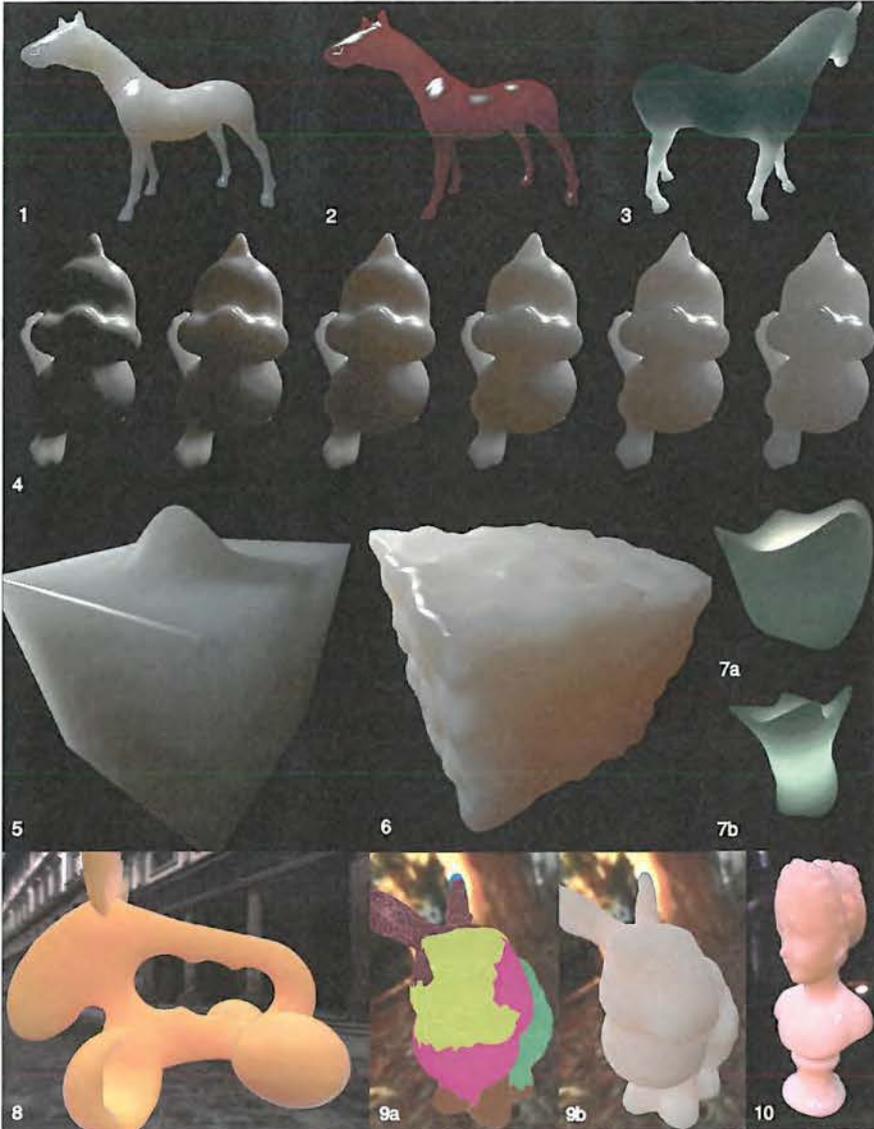


Figure 3.10: Interactive rendering results. **1-2-3**: Horse model with different materials. **4**: Changing the scale of subsurface scattering for whole milk. **5**: Tessellated cube with bump casting shadow. **6**: Deformation on skim milk cube using Perlin noise. **7**: Twist-deformation on candle. **8**: Example of a genus 1 model. **9**: Visualization of the traversed clusters in the hierarchy for shading the bunny's ear. **10**: Marble bust rendered in a high dynamic range environment.

3.4 Local Diffusion

In the previous section, we showed how discretization of the solution can lead to efficient integration of the FDA surface integral (eq. 3.6). It is particularly suitable for very translucent objects, i.e. when the diffusion length l_d (table 2.2) is large compared to geometry. On the other hand, if scattering remains local (small l_d), contributions from distant surface locations are negligible due to the strong exponential decay of the dipole BSSRDF. Performing a global gathering step therefore becomes less efficient. Another issue with small-scale diffusion, is that visual effects associated with subsurface diffusion become less pronounced, more subtle. As a result, the granularity of the solution should be dense enough to capture such effects. The boundary element method will perform less well, as the solution is computed only per element.

These arguments motivated us to develop a method that focuses on computing diffusion locally, in a more detailed fashion. A naive approach to this, would be applying uniform point sampled integration [JB02] in a local area around the point of interest. However, such a scheme is inefficient, as many samples will be required, and is prone to undersampling, causing objectionable low frequency noise.

A more efficient integration can be performed using Monte Carlo integration with importance sampling [KW86]. Importance sampling distributes the samples according to the function to be integrated, in our case R_d . To this end, we propose a method for generating such importance samples in section 3.4.1. We restrict the samples to lie within a certain radius w.r.t. evaluation point.

In order to further speed up the computation, we do not integrate directly on the surface of the object, but instead integrate the irradiance values in image-space (see section 3.4.2). The basic algorithm is depicted in figure 3.11. Importance samples are generated on the tangent plane at x_o . These are then projected on the object (in case of a perspective viewer with a perspective projection), and the corresponding irradiance values are accumulated there. This sampling strategy is amenable for implementation on graphics hardware. As an alternative for the case where a suitable texture parametrization is available, we discuss the possibility of combining this image space sampling with texture space sampling.

The granularity of the solution is very high, as we perform computations per-pixel. This will be important to capture subtle perceptual cues associated with subsurface diffusion, such as blurring of hard shadow borders and concealment of small-scale geometric features.

We will demonstrate our algorithm mainly on the problem of skin render-

ing, as it was specifically designed with this application in mind. In order to show its generality, examples of other materials are briefly demonstrated as well. It is conceivable that, in order to deal with various diffusion lengths, one may construct a hybrid method, consisting of local importance sampling, and the hierarchical boundary element discussed in section 3.3. We leave this as future work.

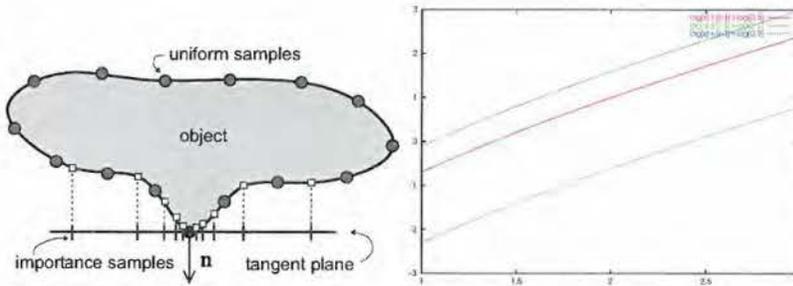


Figure 3.11: **Left:** Overview of the proposed technique: we locally sample incoming lighting in the tangent plane, according to the importance of the BSSRDF function. **Right:** Envelope of function for root finding procedure (section 3.4.1).

3.4.1 Importance Sampling of the BSSRDF

In this section we derive an exact importance sampling scheme for the dipole BSSRDF (see table 3.1) on an infinite plane. In the next section, we will discuss its use for sampling on arbitrary geometry. We refer the readers unfamiliar with importance sampling to Kalos and Whitlock [KW86] or Dutré et al. [DBB03].

First, we compute the total reflectance of the BSSRDF over an infinite plane. Consider the surface integral in equation (3.6). Over an infinite plane in polar coordinates, the integral can be reformulated as:

$$\int_0^{2\pi} \int_0^\infty E_\eta(r, \theta) R_d(r) r dr d\theta$$

First, we split up R_d into 2 terms:

$$\begin{aligned} R_d(r) &= \frac{\alpha'}{4\pi} [R_d^r(r) + R_d^v(r)] \\ R_d^r(r) &= z_r(1 + \sigma_e d_r) \frac{e^{-\sigma_e d_r}}{d_r^3}, \\ R_d^v(r) &= z_v(1 + \sigma_e d_v) \frac{e^{-\sigma_e d_v}}{d_v^3} \end{aligned}$$

Observe that it is easy to analytically integrate these terms over the 2D plane. By substituting d with $\sqrt{r^2 + z^2}$ we obtain:

$$\int_0^\infty z(1 + \sigma_e d) \frac{e^{-\sigma_e d}}{d^3} r dr = z\sigma \left[-\frac{1}{u} e^{-u} \right]_{u=\sigma_e z}^{u=\infty} \quad (3.22)$$

$$= e^{-\sigma_e z}, \quad (3.23)$$

with z either z_r or z_v and d either d_r or d_v . It follows that the total reflectance of the infinite plane is [JMLH01b]:

$$\rho = \frac{\alpha'}{2} [e^{-\sigma_e z_r} + e^{-\sigma_e z_v}] \quad (3.24)$$

We are now ready to derive the importance sampling procedure. Let us find samples r_i , distributed according the second part of the integrand: $R_d(r)r$. As it only depends on distance, the angle θ will be sampled simply using a uniform density $1/2\pi$.

Draw a uniform random number ξ from the unit interval. We must find a distance $r(\xi)$ such that the probability of finding $r(\xi)$ in an interval $[a, b]$ equals $\frac{2\pi}{\rho} \int_a^b R_d(r)r dr$, i.e. the probability density function is defined as: $p(r) := \frac{2\pi}{\rho} R_d(r)r dr$. Here, $\frac{2\pi}{\rho}$ is a normalization factor.

We start by selecting one of the two terms R_d^r or R_d^v w.r.t. their respective strengths. For this purpose, a suitable threshold T is derived from the integrated terms (integration is analogous to equation 3.22):

$$T = \frac{e^{-\sigma_e z_r}}{e^{-\sigma_e z_r} + e^{-\sigma_e z_v}}$$

If $T < \xi$, pick R_d^r and normalize ξ : let $\xi \leftarrow \frac{\xi}{T}$. Otherwise pick R_d^v and let $\xi \leftarrow \frac{\xi - T}{1 - T}$.

The cumulative distribution function corresponding to the chosen term (subsequently dubbed R_d^*), is inverted:

$$\int_0^{r(\xi)} R_d^*(r) r dr = \xi \int_0^\infty R_d^*(r) r dr.$$

Again, performing integration as in equation (3.22), we obtain:

$$(1 - \xi)e^{-\sigma z} = \frac{1}{\sqrt{1 + (\frac{r}{z})^2}} e^{-\sigma_e z \sqrt{1 + (\frac{r}{z})^2}}$$

Substituting $u = \sqrt{1 + (\frac{r}{z})^2}$ yields:

$$\sigma_e z(u - 1) + \log u + \log(1 - \xi) = 0 \quad (3.25)$$

This equation can be solved numerically for u . Note that this function is a smooth, monotonically increasing function for which we can find its root rapidly with Newton's method (fig. 3.11). Let $f(u)$ be the left hand side of 3.25. We initialize $u = 1$ and iterate as follows:

$$u \leftarrow u - f(u)/f'(u) \quad \text{with} \quad f'(u) = z\sigma_e + 1/u$$

Finally, we obtain a sampling distance r with:

$$r = z\sqrt{u^2 - 1}.$$

3.4.2 Integration over the Surface

In this section we describe how we employ the importance sampling scheme.

Two problems arise when extending the previously discuss sampling scheme to arbitrary surfaces. Firstly, for a given importance sampling distance to a point on the surface, it is not trivial to construct a location at this distance directly on the surface. This would require access to the local geometry, as well as a complex search routine. Therefore, we simply construct the samples in the tangent plane of that point, and project them onto the underlying surface.

The second problem is the acquisition of irradiance information over the surface. Two options are possible:

- We can render the irradiance once for one (or possibly more) reference views into a 2D image or texture map. The light view, similar to shadow mapping [Wil78, DS03], is a possible candidate. However, it may cause

artifacts at locations which are oriented perpendicularly to the light due to severe undersampling. This is a common problem for regular shadow mapping, which will likely worsen in our case. We opt for the observer's view (camera view), which already contains most relevant irradiance information. Also, it enables us to efficiently handle more complex lighting (e.g. from an environment map), without much extra cost.

- When a suitable texture parametrization is given, irradiance can be computed in the parametrized space and stored as a texture map.

Both options have particular advantages and disadvantages, which have to be traded off against each other. The first approach is not able to fully represent all relevant irradiance (e.g. from back facing geometry). The second approach does not have this problem, but requires the geometry to have a suitable parametrization.

In the following sections, we will explain each approach, and discuss the trade-offs in more detail.

Image-Space Sampling

In a nutshell, we will generate importance samples in the tangent plane associated with the pixel of interest, and project them onto the surface. Next, integration will consist of looking up and accumulating irradiance at each sample, using the irradiance map (see figure 3.14). Let us go over this process in more detail.

Consider the local scattering integral (equation 3.6):

$$\begin{aligned} I &= \int_A R_d(r) E_\eta(\mathbf{p}) dA_p \\ &= \int_{A'} R_d(r) E_\eta(\mathbf{p}') \left| \frac{dA_p}{dA_{p'}} \right| dA_{p'} \end{aligned} \quad (3.26)$$

The primed variables are associated with the tangent plane (see figure 3.12). Before integration, we compute the irradiance map, which will contain function E_η . Then, for each pixel to shade, our objective is to solve integral (3.26). To this end, we construct a set of samples in the tangent plane and project them on the surface w.r.t. the center of projection of our irradiance map (see figure 3.12). Basically, the integral is solved in *image space*.

For each sample point \mathbf{p}' we can then easily look up the irradiance $E_\eta(\mathbf{p}')$ in the irradiance texture. Since we assume local scattering, we take A to be a

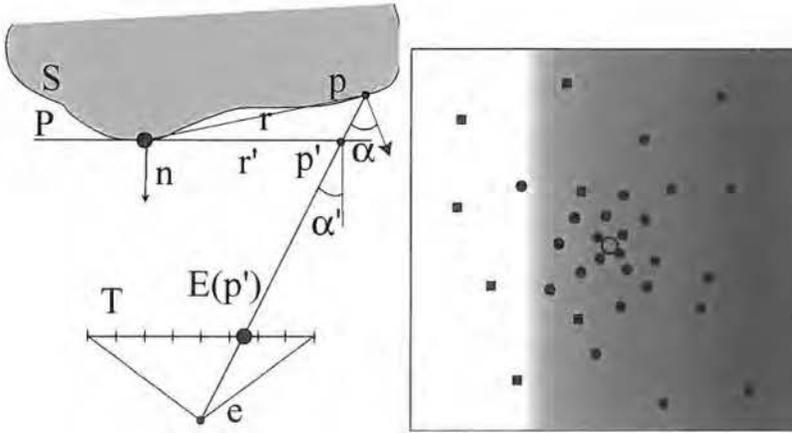


Figure 3.12: **Left:** geometry for sampling the irradiance in the tangent plane. The irradiance at the projected sample point \mathbf{p} can be retrieved in the irradiance texture T . **Right:** combined sampling of importance samples (dots) and uniform samples (squares) for a point (circle) near a shadow border. In this situation, almost all of the importance samples will get a negligible contribution. The uniform samples have a higher chance of receiving a significant contribution.

fraction of the surface Γ , and A' the corresponding area on the tangent plane. For simplicity, A' is considered to be a disc of radius R' .

The transformation implies multiplying with a Jacobian $\left| \frac{dA_p}{dA_{p'}} \right|$, which depends on the mapping between points on the tangent plane and corresponding points on the surface. Since the mapping is projective, we choose to relate the corresponding points via solid angles. One can observe that solid angles occupied by the differential areas surrounding \mathbf{p} and \mathbf{p}' w.r.t. e (the center of projection) are equal:

$$dA_p \frac{\cos \alpha}{d^2} = dA_{p'} \frac{\cos \alpha'}{d'^2},$$

with $d = \|\mathbf{e} - \mathbf{p}\|$ and $d' = \|\mathbf{e} - \mathbf{p}'\|$. Substituting this in equation 3.26 yields:

$$\begin{aligned} I &= \int_{A'} R_d(r) E_\eta(\mathbf{p}') \frac{\cos \alpha}{\cos \alpha'} \frac{d'^2}{d^2} dA_{p'} \\ &= \int_0^{2\pi} \int_0^{R'} R_d(r) E_\eta(\mathbf{p}') \frac{\cos \alpha}{\cos \alpha'} \frac{d'^2}{d^2} r \, dr \, d\phi \end{aligned}$$

The last step results from a transformation to polar coordinates, in which factor $R_d(r)r$ corresponds to the distribution of our importance sampling routine.

The integral is solved numerically using a set of N samples $p_i = (\phi_i, r_i)$, where r_i is generated as described in section 3.4.1 and ϕ_i is uniform:

$$I = \frac{\rho}{2\pi N} \sum_i^N \frac{R_d(r_i)}{R_d(r'_i)} E_\eta(\mathbf{P}'_i) \frac{\cos \alpha_i d_i'^2}{\cos \alpha'_i d_i^2}. \quad (3.27)$$

The importance samples are generated in tangent space, i.e. we generate r'_i , but we actually take samples with distances r_i . Therefore the samples are weighted by $R_d(r_i)/R_d(r'_i)$.

The final outgoing radiance is computed by multiplying I with the Fresnel transmittance for the current viewing direction (see equation 3.5).

Texture-Space Sampling

When a bijective and continuous parametrization is available, we can perform the irradiance sampling in texture space. The algorithm remains almost the same, except that irradiance is computed and stored in texture space. The integration is still performed in image space at pixel accuracy, but irradiance lookups are done on the texture space irradiance image. We compute the irradiance in texture space by rendering our polygon mesh in 2D, using texture coordinates from the parametrization as vertex positions. The samples' positions are again generated in tangent space. The corresponding texture space locations are computed linearly using the 2D basis given by:

$$\left[\frac{\partial s}{\partial E_1}, \frac{\partial t}{\partial E_1} \right] \quad \left[\frac{\partial s}{\partial E_2}, \frac{\partial t}{\partial E_2} \right]$$

where s, t are texture coordinates, and E_1, E_2 define an orthonormal basis in the local tangent plane. This simple approach avoids artifacts due to the inherent anisotropic and scale distortions of the mapping. Finally, in order to compute the distance r_i in equation (3.27), the world space position of each irradiance texel is stored in an additional floating point texture. This texture can also be obtained on-the-fly, the same way as irradiance.

Using texture space sampling, irradiance is represented completely across the object. When a model is lit from the back, we can now render more subtle scattering effects, e.g. near the silhouette. Also, undersampling artifacts are significantly reduced. This is demonstrated in the leftmost image of figure 3.15. We note that the parametrization must be fully bijective, i.e. the triangles in texture space cannot overlap. Also, discontinuous parametrizations consisting of one or more chunks require some additional overhead: care must be taken

when sampling near discontinuities. For simple cases, such as the head model in the leftmost image of figure 3.15, the number of discontinuities is very small and do not occur on the model at obvious locations (e.g. across the face), we can get away with simply ignoring them.

3.4.3 Implementation

This section discusses some noise reduction techniques and the implementation of our algorithm on graphics hardware.

Reducing Noise

Visible noise caused by variance are inherent to Monte Carlo techniques. It has a high-frequency character, since sampling occurs per pixel. Note that this type of noise is less objectionable than the low frequency noise caused by uniform point sampling [JB02].

Some measures can be taken to reduce noise:

- **Quasi-Random Number Generation** We employ quasi-random numbers as input to our importance sampling scheme. They minimize discrepancy (i.e. increase uniformity), yielding an improved convergence rate, and perform well for low dimensional sampling [DBB03]. In particular, the Niederreiter sequence is employed in our implementation [Nie92].

Interleaved sampling [KH01] is applied to avoid banding artifacts, and gracefully transforms noise into a dither-like pattern. Also, it is more amenable for a hardware implementation.

- **Combined Sampling** The importance sampling algorithm only makes allowance for the variation in function $R_d(r)r$. This strategy is efficient as long as irradiance does not vary quickly. This is not a safe assumption. For instance, figure 3.12 depicts how in unlit regions near shadow borders, few importance samples will ever have a non-zero contribution. Uniform samples will perform better in this case. To combat this, we generate a set of uniform samples together with the importance sample set. Using the balance heuristic proposed by Veach et al. [VG95], we can safely combine the two sets in an unbiased fashion.

Using Graphics Hardware

Our algorithm can be implemented on modern consumer graphics hardware. Current graphics technology offers a high degree of programmability at the fragment level, and provides full floating point precision throughout its render pipeline [ati, nvi]. These features make the implementation of our algorithm feasible in hardware.

Before rendering, we first compute a set of importance sampled distances r_i and then precompute an offset vector for each importance sample, which is used to quickly construct a sample from the tangent basis vectors at each pixel. Also the inverse probability density function is precomputed as a weight for each sample, as well as the weight for the balance heuristic [VG95].

On an ATI Radeon 9700Pro, which have used for our implementation, the algorithm requires $N + 2$ passes, where N is the number of samples. The first pass renders an irradiance texture in an off-screen buffer for the camera view. Aside from the radiance, the depth buffer from this view is also stored in a separate texture. An additional depth buffer is employed to perform shadow mapping [Wil78].

The accumulation of the N samples requires two floating point buffers. In one of the passes, one acts as render target and the other as temporary storage of previous passes. In a subsequent pass, the roles are switched in order to avoid a costly texture copy. The alpha channel is used as a counter for the number of samples. It is only incremented if a valid irradiance sample is found, i.e., if the depth of the fetched irradiance pixel is closer than the far plane. The components of the summation in equation 3.27 are computed as follows in the fragment shader:

- The sample weights are passed as a single (global) parameter to the fragment shader.
- The irradiance $E(\mathbf{p}'_i)$ is retrieved using a projective texture fetch.
- The function R_d itself is stored as a 1D floating point texture for a certain range. Using the information in the depth texture (projective fetch), the location p_i on the surface is computed, in order to derive r_i . Now we weight the irradiance sample appropriately (see equation 3.27).

We omit the Jacobian to increase rendering speed. For most cases this is a safe assumption, since the tangent plane approximates the surface for a small area A . Nevertheless, it can be computed easily in the fragment shader as well, albeit an increase in shader instructions.

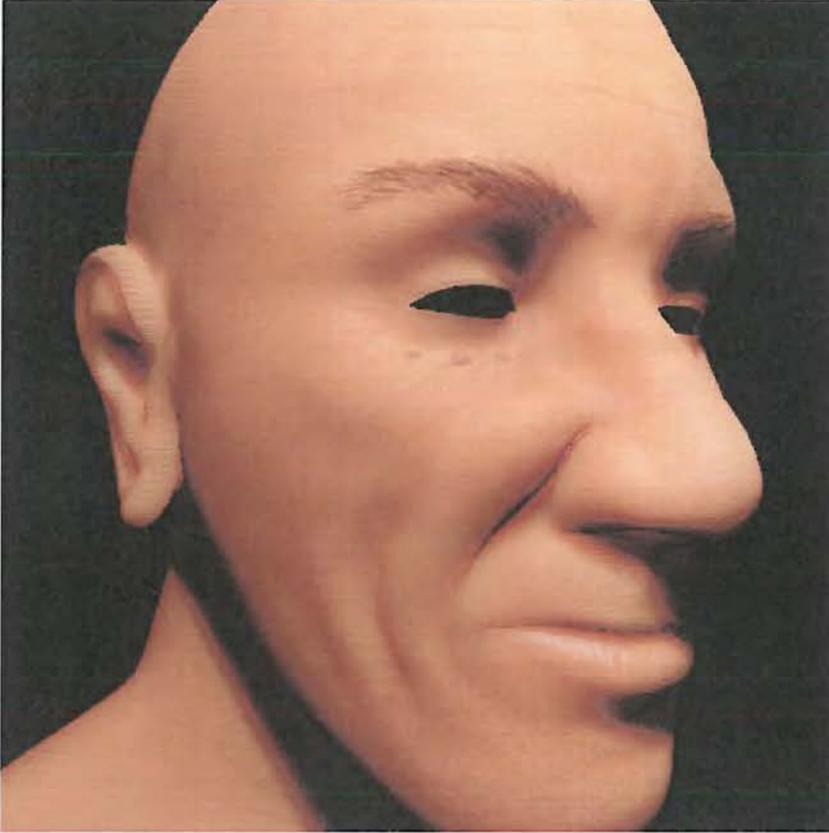


Figure 3.13: An interactive rendering of a bump-mapped model on graphics hardware using measured BSSRDF parameters of human skin [JMLH01b]. The image is rendered at roughly 4 to 5 frames per second.

The final render pass consists of normalizing the resulting pixels with the sample counter in the alpha channel, multiplying with the Fresnel coefficient, and applying a simple tone mapping operator [RSSF02].

3.4.4 Results

The quality of our algorithm can be judged from the image in figure 3.13. A head model whose bounding box was scaled to 10cm^1 was rendered using the scattering parameters for human skin [JMLH01b]. A closeup can be seen in

¹A normal sized head is larger, but we employ a smaller scale to increase the effect of translucency.

figure 3.14. The used irradiance map can be seen in the same figure. A bump map was applied when computing the irradiance values. Subsurface scattering smooths out these bump considerably. The renderings in figure 3.13 and 3.14 have an additionally applied color (texture) map (e.g., for the eye brows). Specular highlights were added as well using Phong's model [Pho75], and are attenuated by a "gloss map" (e.g. for increasing the shininess of the lips).

Rendering speed is about 4 to 5 frames per second for a 500 by 500 image on an ATI Radeon 9700Pro. Our implementation was also tested on an NVIDIA GeForceFX 5800 board, for which we obtained similar timings. Rendering speed degrades roughly linearly with the number of pixels rendered.

Closeups of the head model can be found in figure 3.14. The irradiance texture used in the first rendering can be seen next to it. The bottom-left image shows the shadow boundary from the neck region. The original shadow has actually a sharp boundary, as seen in the irradiance image. Subsurface scattering causes the shadow boundary to be diffused over a larger area. The dithering structure is due to the interleaved sampling [KH01]. The two top-right images show the region on the forehead under different illumination conditions. The typical reddish color shifts are very prominent in this example. More complex lighting can be applied: in the bottom row a rendering is shown using a projective stained glass texture.

Also in figure 3.14, example renderings of two other materials can be seen. The left image was rendered with milk and the right image with marble. Here we did not apply any base or gloss map for the renderings. Even the very translucent milk is rendered without obvious artifacts. All the shadow regions show considerable subsurface scattering (brownish tint).

Figure 3.15 shows a comparison of Jensen et al.'s method [JB02] with our method. The results are virtually the same, apart from some small scaling factor. This is due to the global undershoot from not taking the whole surface into account in the integration. Small differences can be seen at sharp boundaries. E.g., the boundary of the lower lip is not as smooth as with Jensen et al.'s method. This is because no contributions can be gathered from underneath the lip as this part not visible in the irradiance image. Noise at grazing angles of the head is hardly noticeable due to the use of interleaved sampling.

The appearance of spatially varying materials can be simulated by altering the BSSRDF parameters per pixel according to a texture function. In figure 3.15 we demonstrate this by lighting a marble plane with a checkered irradiance pattern. Subtle lighting effects suggest the idea of some internal structure. Even temporal variation in the texture can be rendered at interactive rates. We precompute a set of importance sample offsets, which are

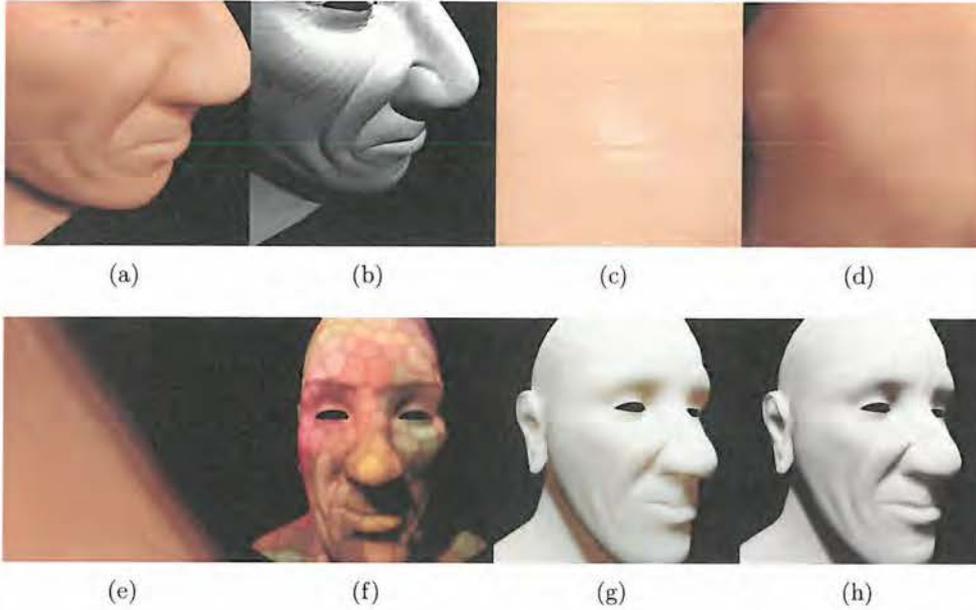


Figure 3.14: Top row: (a) closeup of the head model with (b) the corresponding irradiance map. Notice how the roughness of the surface is washed away due to the subsurface scattering. (c) Forehead lit from above and (d) forehead lit from the side. Note the obvious color shifts due to scattering. Bottom row: (a) shadow region on neck. (b) The same model with skin lit by a stained glass texture. (c) Milk and (d) marble materials applied to the model.

generated for an interpolated set of BSSRDF parameters, and are stored in texture memory. Note that we cannot interpolate the offsets themselves, due to the non-linear effect of altering the BSSRDF parameters. Since the importance sampling algorithm is simple enough, the offsets might also be generated on the GPU, although we did not implement this alternative.

3.5 Comparison of Interactive Rendering Methods

Since the introduction of the FDA [JMLH01a], much research has been spent on developing interactive rendering methods, which includes the work presented in section 3.2. In this section, we will thoroughly discuss the similarities and differences, together with advantages and disadvantages of each one. We briefly explain each technique, preceded by a short identifier for further

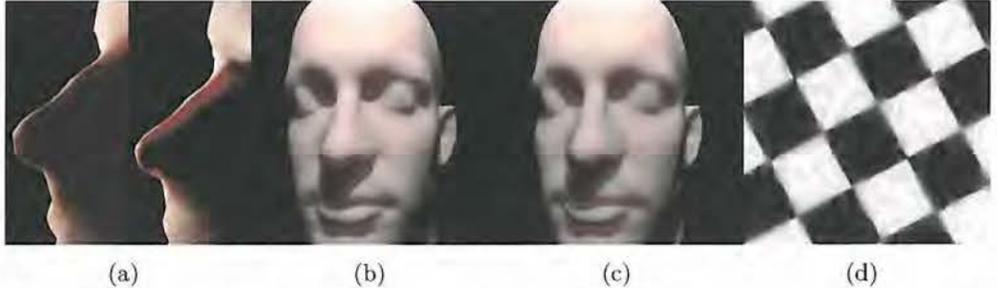


Figure 3.15: (a) Irradiance sampling in image space and texture space, respectively. Comparison of (b) Jensen et al.'s method [JB02] with (c) our method. The results are virtually the same, apart from a small scaling factor which we accounted for here. (d) Rendering with spatially and temporally varying BSSRDF parameters.

comparisons.

LENS : Lensch et al. [LGB⁺02] presented one of the first attempts at interactive rendering of translucency with the FDA. They apply spatially-variant convolution in texture-space to accurately represent the local response to incoming lighting, while the global response is computed vertex to vertex using a large matrix multiplication.

HAO : Hao et al. [HBV03] precompute the FDA surface integration for each incoming light direction, per vertex. During rendering, the incident light direction at each vertex is used to look up the appropriate shading value. Performance was improved [HV04] by compressing the per-vertex tables using spherical harmonics.

CARR : Carr et al. [CHH03] employed a hierarchical mesh for efficient integration. Similar to our method (section 3.3) they construct links and compute form factors between elements in the hierarchy (as precomputation). At run-time, gathering can be performed using graphics hardware, allowing for real-time rendering for varying illumination.

SLOA : The Precomputed Radiance Transfer (PRT) framework [SKS02] is capable of reproducing general global illumination for arbitrary, but distant and band limited illumination. Essentially, PRT precomputes the response in radiance when the object is lit by orthogonal basis functions (spherical harmonics). Real-time rendering for varying illumination is

achieved by linearly combining the basis responses. Its application to rendering translucent objects has been demonstrated by Sloan et al. [SHHS03], using the FDA as an underlying model.

DACH : Dachsbacher et al. [DS03] introduced a shadow mapping-like approach [Wil78] to evaluate the FDA-integral. A hierarchical integration scheme was realized by means of mipmap texture filtering [Wil83], implemented on graphics hardware.

BOSA : Borshukov et al. [BL03] proposed to simulate subsurface diffusion in texture space by means of simple convolution. Later, Sander [San04] implemented this scheme on graphics hardware and obtained real-time performance.

To complete the list, we add our approaches:

MER1 : Hierarchical boundary element method, as discussed in 3.3.

MER2a : Local diffusion (section 3.4) by integration in image space.

MER2b : Same as [MER2a], but with integration in texture space.

3.5.1 Categorization of Common Techniques

Most of these methods were developed concurrently, and therefore share some similarity. We will enumerate them, and discuss the advantages and disadvantages of each.

In general, we can identify several common techniques:

- precomputation
- filtering
- radiosity-like gathering
- hierarchical evaluation

Under the assumption of static geometry, the FDA-integral can be precomputed and tabulated [LENS, HAO, CARR, SLOA]. Texture- or image-space filtering provides an efficient means for subsurface diffusion, either for dealing with local effects [LENS, BOSA, MER2a, MER2b] or global effects [DACH]. In particular, it is amenable for implementation on graphics hardware. Evaluation of the FDA-integral resembles a single gathering step in the radiosity

problem [CW93], hence a similar technique can be applied [LENS, CARR, MER1]. Performance can be improved significantly by employing a hierarchical representation [CARR, DACH, MER1].

Advantages and Disadvantages Precomputation-based approaches assume that geometry is static, and therefore cannot be applied to dynamic, animated models. Also, these methods can be memory-intensive, and therefore require specialized compression schemes [HAO,SLOA].

Texture filtering typically only accounts for local subsurface diffusion (with the exception of [DACH]). To obtain high quality renderings, care must be taken to correctly account for geometry. [LENS] applied a spatially variant filter in texture space. The importance sampling scheme [MER2a,MER2b] can also be interpreted as a spatially invariant convolution, but the kernel does not have to be stored explicitly. [BOSA] neglect the distortion due to texture parametrization, and apply a static filter. A texture parametrization should fulfill certain requirements, as discussed in section 3.4.2, and might be hard to construct. Image-space sampling [MER2a] yields high quality results, but misses certain illumination information, as irradiance is only sampled and stored for the observer's view. Despite this deficiency, the images by [MER2a] still look plausible.

Radiosity-like approaches are very efficient at computing the global response, which proves to be very important for large-scale scattering (e.g. in fluids). On the downside, due to the inherent discretization of the boundary, high frequency details in shading (e.g. shadow borders) might not be represented accurately [CARR, MER1]. [LEN] combine this with filtering, thereby increasing detail. Hierarchical evaluation requires constructing a multi-resolution mesh (with the exception of [DACH]), which is a non-trivial task.

3.5.2 Comparison Table

Table 3.6 gives a concise comparison of all discussed techniques. Let us explain each column:

- **Deformations and Material Properties** refers to the ability to deal with deformations (for animated models) and adjustment of material parameters.
- **General Illumination** indicates whether different types of illumination are supported (e.g. environment mapping), as opposed to only simple point/directional light sources.

- The columns **CPU** and **GPU** indicate the implementation platform. If the technique is implementable on graphics hardware, one can benefit from parallelism, and hence better performance.
- **Parametrization** refers to the requirement of a suitable parametrization.
- **Precomputation, filtering, radiosity-like** and **hierarchical** indicate which techniques (as discussed previously) are employed.
- **Solution Granularity.** The granularity of the solution is indicated by per-(P)ixel, per-(V)ertex or per-(T)exel. The latter indicates that a solution is computed and stored in texture space.

We refrained from including a column about rendering speed. It proves difficult to assess the absolute speed of each method, due to the dependence on factors like hardware, implementation quality, and nature of input data. However, we can safely state that all of the discussed methods have been shown to obtain at least interactive rendering rates (i.e. 1 Hz and faster).

3.5.3 Discussion

From our comparison, it appears that [DACH] is fast and flexible. However, the quality of their results is inferior compared to the other discussed methods. Judging from the renderings in the corresponding paper [DS03], we believe the objects appear too opaque. Also, it is not clear how the aliasing problems inherent to shadow mapping [RSC87] should be avoided.

It seems that our approaches [MER1,MER2a,MER2b] pay more attention to both accuracy and efficiency, compared to the rest. Techniques like [BOSA] and [DACH] yield fast integration, at the cost of reduced accuracy. Precomputation-based methods always have high accuracy, but integration is done offline.

Conclusion In contrast to other techniques, our approach strives for two goals. First, accurate integration of the FDA integral (3.6), at a local level [MER2a,MER2b] and global level [MER1], yielding high quality renderings. Second, it is suitable for dynamic settings, e.g. for animated models.

Technique	Deformations and Mat. Properties	General Illumination	CPU	GPU	Parametrization	Precomput.	Filtering	Radiosity-like	Hierarchical	Solution Granularity
MER1	X	X	X	-	-	-	-	X	X	V
MER2a	X	X	-	X	-	-	X	-	-	P
MER2b	X	X	-	X	X	-	X	-	-	P
LENS	-	X	X	X	X	X	X	X	-	V/P
HAO	-	-	X	-	-	X	-	-	-	V
CARR	-	X	-	X	-	X	-	X	X	V
SLOA	-	X	X	-	-	X	-	-	-	V
DACH	X	-	-	X	-	-	X	-	X	V/P
BOSA	X	X	-	X	X	-	X	-	-	T

Table 3.6: Comparison table of different techniques. See section 3.5.2 for an explanation of each column.

3.6 Analysis of the Fast Dipole Approximation

The dipole model has been shown to be very efficient: it is capable of reproducing plausible rendering, and is several orders of magnitude faster than Monte Carlo based approaches. Essentially, physical accuracy is traded for performance. In this section we will explore the extent to which the assumptions hold, and assess their impact for computer graphics.

3.6.1 Physical Soundness

In section 2.4, we mentioned two conditions a real-world BSSRDF should fulfill: (1) conservation of energy and (2) reciprocity. Let us verify these conditions.

First, consider the case of a semi-infinite medium. Observe equation (2.26) for an infinite plane:

$$\begin{aligned}
\int_{\Gamma} \int_{2\pi} S(\dots) d_{\perp} \omega_i dA_i &= \frac{F_t(\eta, \theta_o)}{\pi} \int_{\Gamma} \int_{2\pi} R_d(r) F_t(\eta, \theta_i) d_{\perp} \omega_i dA_i \\
&= F_t \left[\underbrace{\int_{\Gamma} R_d dA_i}_{\leq 1} \right] \left[\underbrace{2 \int_0^{\frac{\pi}{2}} F_t \cos \theta_i \sin \theta_i d\theta_i}_{\leq 1} \right] \quad (3.28)
\end{aligned}$$

where $r = \|\mathbf{x}_i - \mathbf{x}_o\|$. The integral over R_d is ≤ 1 ; see total reflectance formula (3.24). Clearly, condition (1) is met. Reciprocity is also fulfilled, because S is simply the product of Fresnel coefficients for the incoming and outgoing direction, and a function of distance.

For arbitrary geometry, reciprocity can be ensured through the “rotated” dipole placement scheme (section 3.1.3). However, the dipole BSSRDF violates conservation of energy. Let us illustrate this by means of a counterexample. Consider a marble cylinder, and use measured material coefficients [JMLH01a]:

$$\sigma_a = .0021 \text{ mm}^{-1} \quad \text{and} \quad \sigma'_s = 2.19 \text{ mm}^{-1}$$

and a relative index $\eta = 1.5$. The third factor in equation (3.28) will be 0.4541. Assume the first factor of equation (3.28) is one (i.e. perpendicular view direction). Take the cylinder’s radius $12 \text{ mm} \approx 25l'_{mfp}$ and its height $2 \text{ mm} \approx 4l'_{mfp}$, and place \mathbf{x}_o along the axis, on the top. We now integrate R_d over the cylinder’s upper and lower disc using equation (3.22), and obtain value 1.126. Finally, one can easily see that equation (3.28) reaches value 1.02. The effect increases as we reduce thickness: for $h = 1 \text{ mm}$, the same integration becomes 1.28. Note that we used the rotated dipole scheme here. The normal-aligned configuration already violates condition (1), and thus does not make sense in this context.

From this example, it is easy to see how allowing arbitrary geometry causes the integral over R_d to be unbounded. Especially for objects with a small geometric features w.r.t. the mean free path length (e.g. thin objects), the situation worsens.

When combined with global illumination algorithms [JB02], violation of energy conservation might cause stability issues: interactions with translucent objects will add energy to the system, taking the global transport equation [Kaj86] off balance.

3.6.2 Phenomenological Soundness

From our experience, we found that these components lie at the basis of plausible rendering of translucency:

- Reduction of geometric salience: light can enter the object, and exit at a seemingly arbitrary location. As a result, the appearance of geometric features is diffused.
- Chromatic shifts: due to wavelength dependency of scattering, color variations occur, specially when multiple scattering is abundant. E.g. red light is scattered more in human skin.

In their study, Fleming et al. made similar observations, and analyzed them in detail [FJB04]. They found that these subtle effects cannot be mimicked by ad hoc techniques, like blurring or altering contrast. We believe the dipole model's success is largely due to the ability to faithfully reproducing these effects. But, despite this ability, there are limits to its perceptual soundness.

Apart from the underlying assumptions of diffusion, additional approximations are made in the dipole model. In particular, incident illumination is reduced to being isotropic, and the boundary condition is not satisfied exactly. These are mild assumptions, likely having little perceivable effect. The semi-infinite medium approximation (SIMA) however, is more significant. When the surface is near-planar w.r.t. the diffusion length, it makes sense. For instance, in human skin, the response to incoming light typically reaches only a few millimeters. For more translucent materials it will be less accurate. In particular, materials should not be too thin w.r.t. the scattering scale, as the medium is assumed to be infinitely "deep".

Another implication of the SIMA, is that the dipole model cannot account for the presence of sharp boundaries, like corners. Such features tend to become darker, as there is less of the medium to scatter in, and more light leaks out of the object (see discussion about the boundary condition in section 2.3.2).

For non-convex objects, internal visibility becomes important. E.g., a hollow object will have a different appearance than one without voids. It is tempting to extend the dipole model with a visibility factor between the incoming and outgoing surface location in equation (3.6). However, one should realize that visibility here is not the same as point to point visibility; see figure 3.16.

To summarize, the SIMA has a visible effect on the appearance, especially for highly scattering materials. In particular, objects may become brighter and appear more translucent. Also, geometric dependence is affected: sharp features do not exhibit leaking of light, and internal visibility is not taken into consideration. Later, in chapter 5, we will experimentally explore the impact

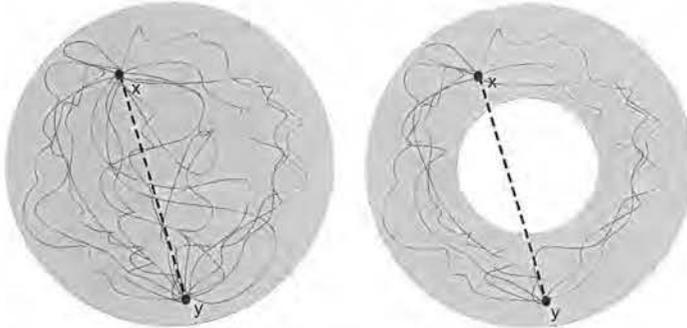


Figure 3.16: Illustration of *internal visibility*. In an optically thick medium, point-to-point visibility (i.e. measuring occlusion along the dashed line), is not relevant since light will likely be blocked along this path. Due to abundant multiple scattering, photons starting from \mathbf{x} may still arrive at \mathbf{y} . The number that reaches this destination is still affected by the presence of the boundaries, similar to point-to-point visibility (e.g. due to voids or concavities). The notion of “indirect” visibility makes more sense: \mathbf{x} and \mathbf{y} share a visibility relation as the aggregate effect of possible paths a photon may travel between them. We call this phenomenon internal visibility.

of these deficiencies.

3.7 Conclusion

We briefly reviewed the FDA, and the accompanying BSSRDF model. Most notably, it reduces the diffusion problem to computing an integral over the object’s surface, and does not require matrix inversion. The FDA model enables us to interactively render translucent objects. We developed two efficient algorithms to realize this.

In the first method, we discretize the solution over all triangles of the model. An efficient semi-analytical procedure was developed for computing per-triangle integrals. Essentially, it computes the total contribution of a triangle scattering light onto a point. Our experiments show that it is an improvement over a previously introduced point sampling scheme in both performance and accuracy. The integration over the whole surface is performed hierarchically by clustering triangles. Interactions are dealt with in different levels in the hierarchy, thereby reducing the complexity for the integration from quadratic to log-linear.

The second method is a simple but efficient algorithm to render the local effect of subsurface scattering. It is applicable for materials with a small scattering response, such as human skin. We derive an exact importance sampling scheme for the dipole BSSRDF. This scheme is employed to perform local integration over the surface in image space or in parameterized texture space. Our algorithm can be implemented on modern consumer graphics hardware.

Compared to other methods, we endeavored to be both efficient and accurate at integrating the dipole BSSRDF surface integral. Our methods feature the ability to vary viewpoint, illumination, material parameters and geometry can be altered, at interactive frame rates.

Finally, we analyzed the underlying assumptions in the FDA. From a phenomenological point-of-view, the BSSRDF model cannot capture certain geometry-dependent effects. Also, it turns out that the BSSRDF violates the law of energy conservation. The rotated dipole placement scheme was introduced to enforce reciprocity.

Chapter 4

Volumetric Simulation

Previously, we discussed methods for interactive rendering of translucent objects, at the cost of abandoning accuracy. This chapter we aim at obtaining a more faithful reproduction of subsurface scattering effects, and the incorporation of heterogeneities in the material. In particular, we develop a computational approach to solve the diffusion equation in 3D, based on a rapid multigrid-based solver [Hac85].

4.1 Overview

We will discretize the diffusion equation on a regular volumetric grid, apply finite differencing, and solve the resulting linear system using the multigrid method. Stam [Sta95] was the first to suggest this. However, he only discusses an illustrative case for a simple 2D homogeneous square slab. Here, we extend his work, and make the multigrid approach practical for arbitrary 3D models. Moreover, we will even apply it to heterogeneous materials.

We employ the following machinery to achieve this:

- Regular discretization does not accurately account for arbitrarily shaped domains, possibly causing stability issues near the surface; see figure 4.1. For rendering purposes, one cannot afford errors here. We propose to use the embedded boundary discretization (EBD) [DCL⁺98, JC98], which accounts for the irregular discretization induced by the presence of the surface.
- To increase performance and to keep memory requirements feasible, we employ adaptive refinement [BO84, DeZ93, BBSW94] of the solution

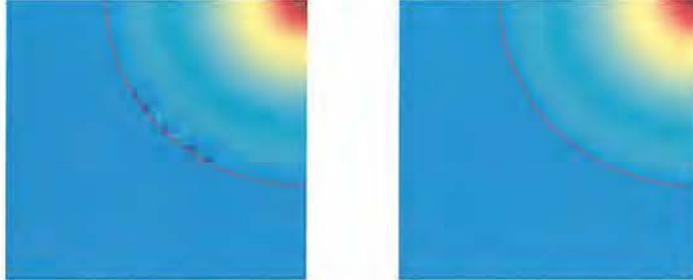


Figure 4.1: Solutions of the diffusion equation in 2D on a 100×100 grid, for the geometry of a circle (outlined in red). The source term is a 2×2 block at the midpoint of the circle. Only the lower left side is shown for illustration purposes. The fluence is shown in false color: blue (low) to red (high). **Left:** Discretization using finite differences yields instabilities near the boundary. Similar artifacts occur in the upper right part. **Right:** Discretization using the embedded boundary results in a consistent solution.

where needed (e.g. near the surface). This is highly recommended, as volumetric representations tend to be prohibitively large.

We start off by explaining the embedded boundary discretization. Next, we detail the application of multigrid, and its adaptive variant. Finally, we present results and conclude this chapter.

4.2 Volumetric Simulation

In this section, we detail our simulation algorithm.

4.2.1 Embedded Boundary Discretization

The EBD builds on an approximation of the divergence found in the diffusion equation (2.10). Therefore, let us define an auxiliary vector function $\vec{F} = D\vec{\nabla}\phi$. The diffusion equation now becomes:

$$\vec{\nabla} \cdot \vec{F} - \sigma_a \phi = -Q_+^0 - 3\vec{\nabla} \cdot [D\vec{Q}_+^1] \quad (4.1)$$

The solution of this PDE will be represented on a 3D uniform cartesian grid, in which its function variables are discretized at the center of each cubical cell. However, our domain boundary, the object's surface, does not necessarily

align with the boundaries of the cells, so care must be taken to consistently represent the information in these cells. We will refer to cells that intersect with the surface as “boundary cells”. The part of a boundary cell that is split by the surface is dubbed “cut cell”. Non-boundary cells are either “full” or “empty”.

Let’s call ϕ the discretization of the unknown function ϕ (fluence). The differential operator in equation (2.10) can be represented in the grid by a weighted summation over its values ϕ , for instance, using finite differences [PTVF92]. One calls the arrangement of the weights a “stencil”, and is akin to the kernel of a convolution filter. The number of terms is very low; typically, only the values in the 4-connected neighborhood are taken into account, and can thus be written as a sparse linear system of equations:

$$\mathbf{A}\phi = \mathbf{S} \quad (4.2)$$

Here, \mathbf{S} is the discretized version of the source term in equation (2.10), and each row in matrix \mathbf{A} contains the stencil for the corresponding element. Note that in equation (4.2), all the ϕ and \mathbf{S} values from the 3D grid have been enumerated as vector.

We will now describe the EBD in detail. Consider $\vec{\nabla} \cdot \vec{F}$ in a single full cell C of size $h \times h \times h$ and approximate this term by averaging it over the cell. Using the divergence theorem, we can relate the resulting volume integral to a surface integral:

$$\vec{\nabla} \cdot \vec{F} \approx \frac{1}{h^3} \int_C \vec{\nabla} \cdot \vec{F} dV = \frac{1}{h^3} \oint_{\partial C} \vec{F} \cdot \vec{n} dA$$

Applying the midpoint integration rule for each cell face yields:

$$\vec{\nabla} \cdot \vec{F} \approx \frac{1}{h} \sum_{face f} \vec{n}_f \cdot \vec{F}(x_f) \quad (4.3)$$

where x_f represents the center of face f . Equation (4.3) can be translated to a stencil using central differences. Note that for full cells the same result can be achieved by direct finite differencing of $\vec{\nabla} \cdot \vec{F}$.

The strength of the EBD is the way boundary cells are treated. A first order approximation of the surface is introduced in order to generalize the integration scheme in equation (4.3):

$$\vec{\nabla} \cdot \vec{F} \approx \frac{\sum_{face f} \alpha_f \vec{n}_f \cdot \vec{F}(x_f) + \alpha_B \vec{n}_B \cdot \vec{F}(x_B)}{\kappa h} \quad (4.4)$$

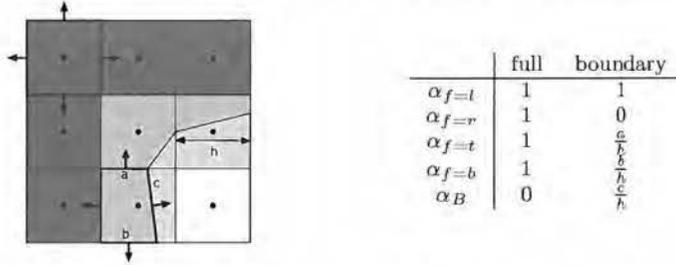


Figure 4.2: **Left:** Geometry of cells in the 2D case for clarity (3D case is analogous). Dark, light and white cells represent full, boundary and empty cells, respectively. The arrows indicate normals at the face center x_f used to evaluate the midpoint integration scheme (see equation (4.3)). **Right:** Face weights used in the discretization of the diffusion PDE into cubic cells of width h (see equation 4.4). Face indices are written as (l)eft, (r)ight, (t)op and (b)ottom.

Here x_B is the center of the boundary surface and n_i^B the normal at this location. Let us clarify the ratio variables (see figure 4.2):

- κ : ratio of cut cell volume to whole cell volume (equals 1 for full cells);
- α_f : ratio of cut cell face area to whole cell face area;
- α_B : ratio of the (first order approximated) surface area to whole face area.

Again using central differences, one can construct a single stencil that can be used on both full and boundary cells. For the case of ordinary finite differences, a different and less accurate stencil needs to be employed in boundary cells. Figure 4.1 illustrates the problems that might occur with the use of finite differencing [PTVF92]. These problems occur because finite differencing requires that any boundaries present in the domain are aligned with the cell faces. Irregular boundaries need to be approximated, leading to inconsistencies. The consequences of such inconsistencies are possible failure of the multigrid algorithm (no convergence) and an inaccurate solution of the radiance at exactly these locations where it is of utmost importance for subsurface scattering. The EBD solves these problems.

The EBD method allows us to naturally incorporate the necessary boundary conditions. Day et al. [DCL⁺98] mention that homogeneous Neumann conditions can be enforced by setting the flux vector at the boundary to zero

(i.e. by cancelling the second term in equation 4.4). In our case we have mixed Neumann and Dirichlet conditions (eq. 2.17):

$$\vec{n} \cdot \vec{F} = -\frac{1}{2A}\phi - \frac{1}{\sigma'_t}\vec{n} \cdot \vec{Q}_+^1 \quad (4.5)$$

In the same spirit, our condition can be easily enforced by substituting $\vec{n}_B \cdot \vec{F}(x_B)$ in the second term of equation (4.4), with equation (4.5). As a result, the embedded boundary stencil can now be applied to both full and boundary cells.

Equation (4.4) is only valid at the center of mass of the cut cells. To solve this problem, we use the data-centering scheme by Johansen and Colella [JC98]; it essentially moves the cell data to the geometric center of the full cell.

4.2.2 Solution by Means of the Multigrid Method

After discretization, we apply the multigrid method [Hac85] to rapidly solve the linear system in equation (4.2).¹ The system is solved on a set of hierarchical grids as follows. First, an approximate solution is computed via a fixed number of relaxation steps on the finest level. Relaxation removes high-frequency errors in the solution, and is therefore referred to as “smoothing”. Then, a correction term is computed by downsampling the residual error $\mathbf{r} = \mathbf{A}\hat{\phi} - \hat{\mathbf{S}}$ onto the next coarser grid. We can now simply solve for the correction term by filling in the restricted residual as right-handed side. This process continues recursively, until the top of the hierarchy is reached. Intuitively, the coarse corrections will deal with low frequency components of the solution. After reaching the top, the direction is reversed and the corrections at each level are upsampled and added to the next finer grid. The whole process is called a “V-cycle” and is iterated until convergence is reached (determined by $\|r\| < \epsilon$).

In our implementation, smoothing is performed via Gauß-Seidel relaxation [PTVF92]. This is a fairly common technique, yielding adequate performance. We downsample corrections using volume-weighted averaging, in order to respect the irregular nature of the boundary cells. For upsampling, we simply replicate parent values in their respective child cells.

Multigrid is key to achieving good performance. Straightforward one-level relaxation becomes very inefficient when the diffusion length is large w.r.t.

¹A more detailed treatment of the multigrid method can be found in Hackbusch [Hac85] or Press et al. [PTVF92].

the object. Many iterations are then required, in order for the fluence to propagate fully throughout the volume. With the multigrid method, “long-distance” diffusion (i.e. low frequencies) can be rapidly established. The number of iterations is less dependent on the diffusion length.

4.2.3 Adaptive Refinement

A dense grid is required to capture fine scale shading details across the surface. However, increasing resolution comes at an unpractical cost of $O(1/h^3)$. Hierarchically subdividing the grid near the surface solves this problem. The refinement process is simple: we subdivide each cell containing the boundary, until a predefined maximum depth is reached.

Inside the object, the same strategy is used to account for sudden changes of incident illumination and material parameters. We take the divergence of the source term and the reduced extinction coefficient σ'_t as refinement criteria. Subdivision halts when these measures drop below a predefined threshold, or when the maximum depth is reached.

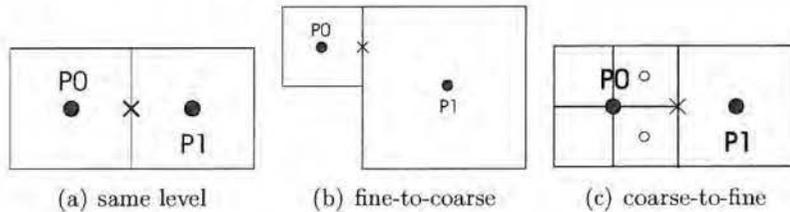


Figure 4.3: Possible cell arrangements when computing the cell face gradient. Dots represent cell centers, crosses indicate central differencing locations. (a) Regular case: cells at the same level. (b) Fine-to-coarse: (c) coarse-to-fine boundary. For case (b) and (c), we apply Popinet’s second order interpolation scheme [Pop03].

Computing the gradient \vec{F} in the hierarchical grid requires some special attention, as neighboring cells may differ in size; see figure 4.3. We apply Popinet’s 3-point interpolation scheme, which basically fits a parabola along each axis [Pop03]. Like non-hierarchical central differencing, this interpolation yields $O(h^2)$ accuracy. The three-point scheme requires that cells do not differ in more than one hierarchy level w.r.t. their neighbors. We enforce this condition as a post-process after initial refinement.

Our refinement scheme will force boundary cells to always lie on the same level (i.e. the finest level). It is conceivable to also allow variation in cell

size here, in order to reflect changes in illumination and curvature. However, to apply the 3-point interpolation scheme, we require access to non-empty neighbors to establish interpolation, which is not possible for boundary cells.

4.2.4 Putting Everything Together

In practice, our simulation algorithm carries out the following tasks:

1. **Preprocessing:** The mesh is loaded from disk and the grid is built using adaptive refinement.
2. **Source Term:** Illumination consists of a point light source. Computing the right hand side of the diffusion equation (2.10), boils down to evaluating equation (2.3) and its gradient. For each cell, we trace a ray from the cell center toward the light, which will intersect with the boundary. At the intersection, we compute incident radiance. Next, we “ray march” back into the volume to compute the optical depth [Jen01]. The gradient is simply obtained via finite differencing.
3. **Multigrid:** The diffusion equation is solved using the multigrid method.

After simulation, at each vertex we interpolate the solution directly from the grid. The final image is rendered using Gouraud shading [Gou71], and can be explored in real-time. We employ equation (2.22) to convert fluence to diffuse radiance:

$$L(\mathbf{x}_o) = \frac{1}{2\pi A} \phi(\mathbf{x}_o)$$

4.3 Results

We implemented our algorithm in C++ on a Pentium 4 1.7Ghz 512 MB RAM configuration.

In table 4.1, we illustrate the performance of our application with different refinement depths.

Table 4.2 illustrates the performance with different materials. We see that as the object becomes more opaque (shorter diffusion length), the rendering time decreases. This is contrary to the behavior of the fast hierarchical dipole solution [JB02], where the number of required samples increases dramatically w.r.t. opaqueness (see our discussion in section 3.2). In our diffusion simulation, light has to travel over a smaller distance, which reduces the number

required of iterations. All timings were gathered on the dragon model under the same lighting conditions. The last four rows represent heterogeneous materials, constructed by interpolation between two different materials with Perlin noise [Per85]. Figure 4.5.1 shows a screenshot of one such rendering.

In figures 4.5.1-2-3, the venus, dragon and buddha models are rendered with a mixture of marble with a ratio of minimum and maximum density of 1:10. The scales of the objects are 10 cm, 2 cm and 4 cm respectively. With “scale”, we refer to the length of the longest side of the object’s bounding box. In figure 4.5.4 a marble dragon at a scale of 2 cm is backlit. Figures 4.5.5-6-7 shows a sculpture with skim milk and whole milk. The last image is backlit: the scattering of light is very obvious for the thin geometric features. Also, notice the chromatic shifts and shadowing. In figures 4.5.8-9-10 the marble dragon is rendered using varying anisotropy g : from left to right, 0.0, 0.5 and 0.9. For small values of g (more isotropic), interactions are localized near the surface, while for higher values light penetrates deeper inside, yielding a more translucent effect.

res	precomp	source	cycles	mgrid	total
64^3	7	2	14	7	16
128^3	11	8	37	109	128
256^3	19	58	40	266	343

Table 4.1: Overview of performance w.r.t. various grid resolution for a marble dragon model (scale 20 mm). Timings are in seconds. The columns indicate grid **resolution**; **precomputation** time, time to evaluate **source** term, number of multigrid **cycles**, **multigrid** execution time and the **total** computation time.

Figure 4.4 shows a comparison of the FDA and our approach. Several visual differences are noticeable. Most importantly, our solution seems to be more dependent on the geometry, causing strong varieties in brightness. The same variations are seen in the reference solution. The dipole solution however, remains smooth regardless of the geometry. One of the causes is the complex internal visibility in this model, which the FDA cannot take into account due to the semi-infinite medium approximation. Note that the FDA rendering is more accurate in terms of chromatic shifting. The brownish tint in our solution can be explained by the way the source term is computed; see section 4.2.4. Incoming radiance attenuation is attenuation by $e^{-\sigma_t d}$, with d the traveled distance inside the object. This function is evaluated at the midpoint of each cell, hence $d > 0$, causing a “chromatic bias”. We work around this problem

material	model	scale	total sim. time
milk mix	dragon	20 mm	63
		10 mm	105
marble	dragon	20mm	215
		10 mm	296
		5 mm	445
marble mix	venus	100 mm	86
		20 mm	206

Table 4.2: Overview of performance with different materials and scales. Timings are in seconds. As the material becomes more translucent, computation time increases.

using a modified source term computation: in each boundary cell, we define the source term simply as the irradiance value at the surface; the inner cells are set to zero. Figure 4.4 shows that this approach yields a more faithful reproduction of color.

Note that both the FDA and our method lack high frequency shading details caused by low order scattering, especially for thin geometry.

4.4 Conclusion

In this chapter, we proposed an computational approach for light diffusion in arbitrary 3D polygonal objects. Basically, we applied the multigrid method to solve the diffusion equation on a regular volumetric grid.

Two important issues are addressed: accurate representation of interactions near the object’s surface, by applying the so-called embedded boundary discretization. In order to improve efficiency and keep memory requirements feasible, the solution adaptively refines where required.

Our diffusion algorithm is accurate as far as the diffusion approximation allows. In comparison to the FDA, internal visibility is resolved correctly, and heterogeneities can be dealt with easily.

Simulation typically takes only a few minutes, after which the solution can be viewed in real-time. Contrary to the FDA-based methods presented in the previous chapter, illumination and geometry changes and cannot be dealt with interactively. Performance depends on the diffusion length. Building the refined grid also incurs an additional cost.

In the future, we would like to look at a more efficient subdivision scheme which accounts for illumination and material variation at the surface. Also,

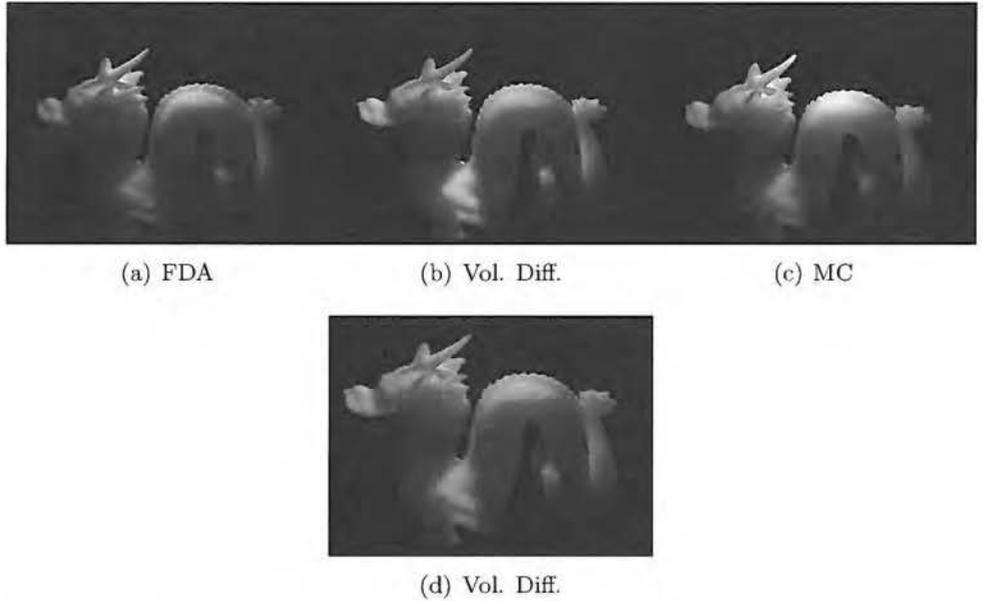


Figure 4.4: The dragon model is lit by a point source placed directly above the model, and rendered at scale 40 mm. (a) Solution using the fast dipole approximation. (b) Result of our volumetric diffusion method. (c) Reference Monte Carlo solution. Our method is capable of resolving internal visibility, contrary to the dipole solution. For instance, the darkening at the lower half of the mouth is caused by visibility issues. However, color is not reproduced correctly in (b). Figure (d) shows another rendering with our method, but with a simplified source term computation. This time, color is reproduced more faithfully.

more accurate source term computation is desirable.



Figure 4.5: Rendering results. 1-2-3: Venus, dragon and Buddha models with a mixture of marble; the ratio of minimum and maximum density is 1:10. 4: Backlit dragon model. 5: Chromatic shift for skim milk. 6-7: Sculpture model with skim and whole milk. 8-9-10: Marble dragon with varying anisotropy ($g = 0$, $g = .5$ and $g = .9$, respectively).

Chapter 5

Boundary Diffusion Model

The fast dipole approximation (FDA) avoids a costly volumetric representation by restricting computations to the object's boundary, which is an attractive property for efficient rendering. However, accuracy is sacrificed in order to improve performance. In this chapter, we show it is possible to formulate an *exact* boundary diffusion model (BDM), which in a way generalizes the FDA. Inspired by integral equations stemming from the traditional boundary element method [HP02], we translate the diffusion problem to a boundary integral equation (BIE) [HASS03].

After discretization, diffusion boils down to solving a linear system, consisting of form factors, exhibiting much similarity to the ones encountered in our FDA-based boundary element method (section 3.3). We show that the corresponding quadrature can be reused here.

As for practical results, we employ the BDM to experimentally validate the semi-infinite medium approximation in the FDA, and confirm the findings of our analysis in section 3.6.

The developments in this chapter are restricted to homogeneous materials, although the model can be extended to deal with composite materials [HASS03].

5.1 Boundary Integral Equation

The diffusion equation for homogeneous media (2.11), can be formulated as the following boundary integral equation [HP02] (see appendix A):

$$\frac{1}{2}\phi(\mathbf{x}) = q(\mathbf{x}) + \int_{\Gamma} \left[G(\mathbf{x}, \mathbf{y}) \partial_y \phi(\mathbf{y}) - \phi(\mathbf{y}) \partial_y G(\mathbf{x}, \mathbf{y}) \right] dA_y \quad (5.1)$$

with $\partial_y = \frac{\partial}{\partial n_y}$ and

$$q(\mathbf{x}) := \int_{\Omega} G(\mathbf{x}, \mathbf{y}) \mathcal{S}(\mathbf{y}) dV_y \quad (5.2)$$

Let us observe ϕ on the boundary Γ , and assume the latter is smooth everywhere. Substituting the boundary condition (2.18) yields the boundary diffusion model:

$$\frac{1}{2}\phi(\mathbf{x}) = q(\mathbf{x}) - \int_{\Gamma} F(\mathbf{x}, \mathbf{y})\phi(\mathbf{y})dA_y \quad (5.3)$$

with

$$F(\mathbf{x}, \mathbf{y}) = \frac{1}{2AD}G(\mathbf{x}, \mathbf{y}) + \partial_y G(\mathbf{x}, \mathbf{y})$$

The integral over the source term in equation (5.2) still requires us to deal with the full volume. Therefore, we simplify it by treating the source term \mathcal{S} similarly as in the FDA. I.e., incoming illumination is represented by internal light sources at a distance z_r below the surface. This simplifies the computation of the volume integral to

$$q(\mathbf{x}) := \frac{\alpha'}{D} \int_{\Gamma} G(\mathbf{x}, \mathbf{y} - z_r \vec{n}_y) E_{\eta}(\mathbf{y}) dA_y$$

Once a solution for ϕ is found, we can simply compute outgoing radiance using equation (2.22):

$$L(\mathbf{x}_o, \vec{\omega}_o) = \frac{1}{2\pi A} F_t(\eta, \vec{\omega}_o) \phi(\mathbf{x}_o)$$

5.1.1 Discussion

Equation (5.3) is a Fredholm equation of the second kind, similar to the radiosity integral equation [CW93]. Function F plays a similar role as the “geometry factor” in radiosity. There are two subtle differences: it depends on both geometry and material, and is not reciprocal.

Term $q(\mathbf{x})$ in equation (5.3) corresponds to the convolution of the source term \mathcal{S} with the Green’s function, effectively representing the solution if no boundaries were present; see equation (2.20). The second term can be interpreted as a correction to account for the boundary.

We simplified the model by approximating the volume integral over the source term (5.2). This slightly weakens our statement about the BDM being exact. However, it is conceivable that more precise evaluation strategies could

be implemented, albeit at a higher cost. We believe this particular approximation will not play a significant role in the end, due to the diffusing nature of subsurface transport.

5.2 Solving the Boundary Integral Equation

We now discuss a practical implementation for solving the BIE.

5.2.1 Discretization

First, the BIE is discretized. We assume our model is given as a mesh consisting of N triangles. Similar to the boundary element method discussed in section 3.3, function ϕ is approximated by constant basis functions over each triangle Γ_j

$$\phi(\mathbf{x}) = \sum_j \phi_j \chi_j(\mathbf{x})$$

Functions $\chi_j : \mathbb{R}^3 \rightarrow \{0, 1\}$ indicate whether a point is inside triangle j . We will solve the problem using point collocation [CW93, HP02]. Each point \mathbf{x}_i is defined as the midpoint of triangle Γ_i . After substituting this discretization in the BIE, and placing the evaluation point on a node \mathbf{x}_i , we obtain:

$$\sum_j \left[\frac{1}{2} \delta_{ij} + \frac{1}{2AD} F_{ij}^a + F_{ij}^b \right] u_j = q(\mathbf{x}_i)$$

With the form factors:

$$F_{ij}^a = \int_{\Gamma_j} G(\mathbf{x}_i, \mathbf{y}) dA_y \quad \text{and} \quad F_{ij}^b = \int_{\Gamma_j} \partial_y G(\mathbf{x}_i, \mathbf{y}) dA_y$$

Eventually, the problem boils down to solving a linear system of equations: $\mathbf{A}\phi = \mathbf{q}$, with

$$\mathbf{A} := \frac{1}{2} \mathbf{I} + \frac{1}{2AD} \mathbf{F}^a + \mathbf{F}^b$$

Vector ϕ is given by $[\phi_1, \dots, \phi_N]^T$, and \mathbf{q} as $[q(\mathbf{x}_1), \dots, q(\mathbf{x}_N)]^T$.

In contrast to the radiosity problem, the resulting system is non-symmetric and non-diagonally dominant. To suit ease of implementation, we apply Gauß-Seidel relaxation [PTVF92]. It turned out to converge in every case in our experiments. An exact analysis of convergence is left as future work. Note that any other “off-the-shelf” solver can be used, such as biconjugate gradient, or GMRES [PTVF92].

5.2.2 Form Factor Computation

In general, we compute the form factors using Monte Carlo sampling. Most of the triangles are distant, and can be approximated safely by a single sample.

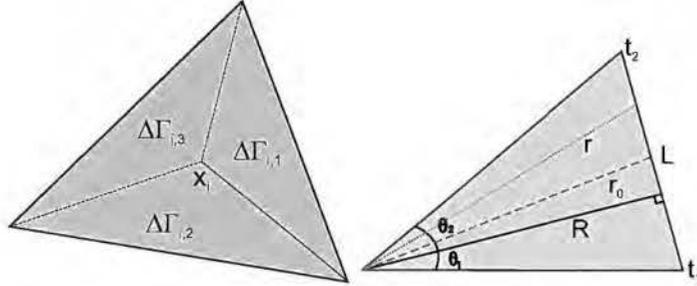


Figure 5.1: **Left:** Splitting up a triangle into subtriangles w.r.t. its midpoint \mathbf{x}_i . **Right:** Geometry for form factor computation of a subtriangle.

Special care must be taken for “self” form factors ($i = j$), which have a singularity in the integrand. Let us consider the first factor F_{ii}^a . We break it into parts $F_{ii}^a = \Delta F_{ii,1}^a + \Delta F_{ii,2}^a + \Delta F_{ii,3}^a$, by integrating separately over each subtriangle $\Delta\Gamma_{i,1}$ (see figure 5.1).

The singularity disappears by integrating in polar coordinates (see figure 5.1):

$$\begin{aligned}
 \Delta F_{ii,1}^a &= \frac{1}{4\pi} \int_{\Delta\Gamma_{i,1}} \frac{e^{-\sigma_e r_i}}{r_i} dA_y \\
 &= \frac{1}{4\pi} \int_{\theta_1}^{\theta_2} \int_0^{r(\theta)} e^{-\sigma_e r_i} dr_i d\theta \\
 &= \frac{1}{4\pi\sigma_e} \left[\theta_2 - \theta_1 - \underbrace{\int_{\theta_1}^{\theta_2} e^{-\sigma_e r(\theta)} d\theta}_I \right] \tag{5.4}
 \end{aligned}$$

where $r_i = \|\mathbf{x}_i - \mathbf{y}\|$. This expression is similar the dipole BSSRDF form factor in equation (3.12), and it turns out we can readily apply the Taylor-based quadrature proposed in section 3.3.1. We only need to make two adjustments. Distance r is substituted for s , and we remove the reciprocal distance in the integrand. Analogously to equation (3.15), the second term can be converted to an edge integral:

$$I = R \int_{t_1}^{t_2} \frac{1}{R^2 + t^2} e^{-\sigma_e \sqrt{R^2 + t^2}} dt$$

Applying the Taylor approximations given in equations (3.17) and (3.18), yields the quadrature rule for an edge segment of length L , in terms of the perpendicular distance R to the edge, and the distance r_0 to the midpoint:

$$I \approx \frac{R e^{-\sigma_e r_0}}{r_0^2} L \quad (5.5)$$

As before, the edge segment should fulfill the conditions stated in section 3.3.1, and can be easily enforced by the corresponding refinement procedure. The remaining subtriangles $\Delta\Gamma_{i,2}$ and $\Delta\Gamma_{i,3}$ are dealt with similarly.

The second term F_{ii}^b reduces to $\frac{1}{2}$, can be found in appendix B.

So far, we have not mentioned how to compute the right hand side \mathbf{q} of our linear system. Due to discretization, computation of each $q(\mathbf{x}_i)$ reduces to a sum of integral terms, each weighted by the corresponding irradiance E_η . These integrals are similar to F_{ii}^a , except for the z_r offset. The only adjustment necessary in our form factor quadrature, is taking $s = r^2 + z_r^2$. Accordingly, we substitute $s_0 = r_0^2 + z_r^2$ for r_0 in equation (5.5). For the remaining triangles simple Monte Carlo integration is utilized, as described earlier.

5.3 Experimental Validation of the Semi-Infinite Medium Approximation

As discussed in section 3.6, the semi-infinite medium approximation (SIMA) in the dipole model, may have a significant effect on appearance. Armed with the BDM, let us explore the limits under which this approximation holds, in an empirical fashion. The results are summarized in table 5.1.

5.3.1 Implementation

The BDM is implemented as described in the previous section. We implement the FDA within the same framework as a single gathering step. FDA form factors are computed with the semi-analytical procedure described in section 3.3 (i.e. the rotated dipole placement scheme is applied as well). Hierarchical evaluation is not employed, as we are currently not concerned with performance.

Our experiments are carried out on a set of simple objects, corresponding to the “problematic” cases discussed in section 3.6. We compute a solution ϕ_{FDA} and a reference solution ϕ_{BDM} . Except where mentioned otherwise, illumination consists of a simple point light, placed above the object. These two solution are displayed using a simple gamma curve: $y = x^{\frac{1}{\gamma}}$. Specular highlights are turned off, and the Fresnel coefficient the for outgoing light direction is ignored for simplicity. Next to the renderings, false colored error plots are shown. To reflect the contrast sensitivity in human vision [Fer98], the error E is computed in log-space:

$$E := |\log(\phi_{FDA}) - \log(\phi_{BDM})|$$

For each rendering, E is displayed in false color (mapping is shown in table 5.1).

In order to push the SIMA to its limit, we conducted our experiments on highly scattering material parameters, i.e. such that the scattering response reaches most of the object. We have opted for marble [JMLH01a], at a scale ranging from 20 mm to 40 mm (i.e. the width of bounding box of the object).

5.3.2 Observations

We begin by observing convex objects.

- **Approximate Semi-Infinite Object**

We start off with the ideal setting for the FDA: a semi infinite medium. This situation is approximated as a finite planar mesh. A point light source is placed directly above the plane. As expected, the results are virtually the same.

- **Smooth Convex Object**

The next case is a sphere. The error is localized in the unlit part, and is generally very smooth. Visually however, we see no significant difference.

- **Sharp Convex Object**

In the case of a cube, the SIMA is brighter at edges and corners.

- **Thin Convex Object**

We show a thin slab rendered from the top and bottom. In the top and bottom view, the FDA produces too much brightness. In the bottom view, we observe a different chromatic shift.

Next, let us verify cases with non-trivial internal visibility.

- **Simple Concave Object**

A torus-like shape is lit from above. We see that the bottom half is slightly too bright.

- **Object with Void**

We constructed a box-like void inside another box. Although possible, reflection inside the void is not accounted for, and thus basically acts as a perfectly absorbing medium, with a refraction index equal to that of vacuum. Its presence is revealed by an “internal shadow” cast on the bottom. Clearly, the FDA cannot reproduce this effect.

- **Split Object**

We conclude with an extreme case for internal visibility: a cube split into two halves, illuminated from the top. The void in between should block all light traveling the bottom half, as visible in the BDM rendering. This effect can also be considered as an internal shadow. However, the FDA continues diffusion across the void.

Finally, we show the cube again, but this time lit by a high dynamic range environment map. Similar errors occur as before, although less pronounced.

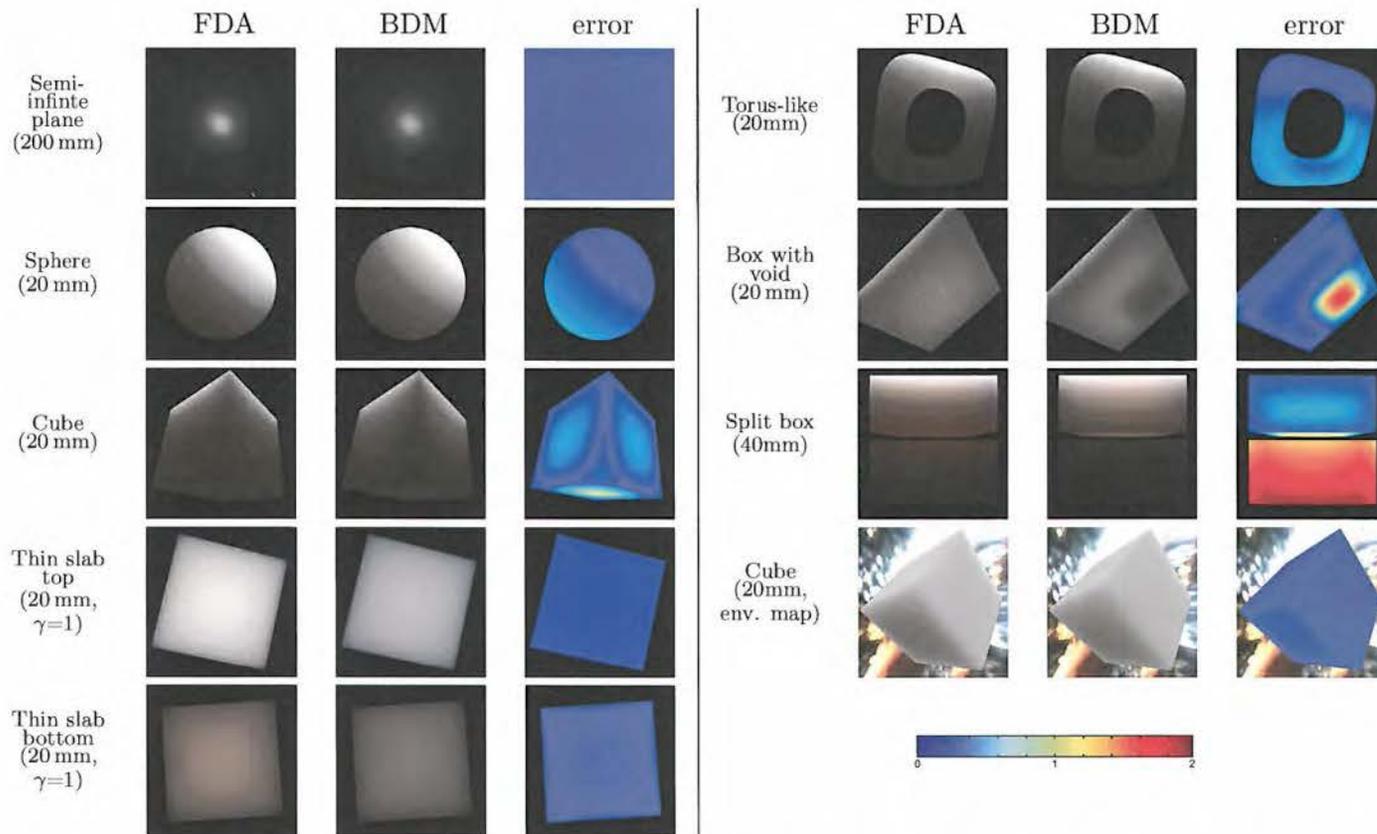


Table 5.1: Experimental validation of the semi-infinite plane approximation. The renderings are tone mapped using gamma correction $\gamma=3$, except where mentioned otherwise.

5.3.3 Discussion

It appears that in directly lit regions, backscattering of light is reproduced reasonably well using the SIMA. For thin geometry, we notice an overall over-estimation of brightness: the object is assumed to be infinitely deep, and will therefore scatter more.

For unlit regions, i.e. where the effect of subsurface diffusion “bleeds” through, artifacts may occur. For smooth, convex objects, such errors do not contain high frequencies, and are thus less noticeable by a human observer. On sharp geometry, the SIMA makes no allowance for darkening at corners and edges due to light “leaking”, a more obvious effect.

Diffusion through thin geometry has a different color and is overly bright (cf. violation of energy conservation; section 3.6).

Internal visibility is a problem for the SIMA. Concavities and voids cause obvious darkening, i.e. internal shadows. The SIMA cannot reproduce this effect. For simple objects like the torus, the lack of internal shadows appears as an overall increase in brightness, and might therefore go unnoticed. For absorbing voids (or similar situations), the impact is more significant.

We must note that real voids filled with air (or vacuum) will behave very differently, and are clearly not absorbing. Still, we decided to include this test, as it nicely illustrates internal visibility problems. It is possible to account for inter-reflection by simply solving a small radiosity system inside the void [CW93].

Illumination by an environment map seems to reduce the effect of the SIMA. We can explain this by observing that, contrary to the case of a point light, irradiance varies less abruptly over the surface. Consequently, the error will be “integrated out”.

As an alternative to the BDM, we could compare to a full Monte Carlo simulation of radiative transfer. However, this turns out to be an unwieldy task, due to the inherently long computation times. More importantly, it complicates the validation, as differences due to both the diffusion approximation and the SIMA cannot be discerned easily. We focused only on the latter; a validation of the diffusion approximation is left as future work. The volumetric method described in chapter 4 could be employed as well. However, we prefer the BDM for two reasons. First, the underlying assumptions about incident illumination are exactly the same (i.e. both use isotropic sources). Second, the granularity of the solutions is different (grid vs per-polygon).

5.3.4 Conclusion

We summarize our findings as follows:

- The visual significance of SIMA-related errors depends on geometry. In particular, smoother objects seem to be less affected.
- Remarkably, even though a serious radiometric deviation occurs in most cases, the corresponding visual impact is not necessarily as significant.
- The error usually manifests itself as a change in brightness. Chromatic deviation is possible, but only observed for thin geometry.
- Soft illumination tends to reduce the artifacts.

5.4 Conclusion

We introduced an exact model for subsurface diffusion, which operates only on the object's boundary. The diffusion equation is reformulated as a boundary integral equation, which shares similarities with the radiosity integral equation [CW93]. With the exception of the hierarchical evaluation, the BDM generalizes the boundary element method introduced in section 3.3. We reused the form factor quadrature which was previously developed for the dipole BSS-RDF.

As for practical results, we employed the BDM for validating the visual quality of the FDA, and confirmed the conclusions of our analysis in section 3.6.

The BDM has not been made practical yet in terms of performance. It is possible to apply Gauß-Seidel relaxation in conjunction with the hierarchical evaluation scheme of 3.3. However, typically many iterations are required before convergence is reached. In the same spirit as our volumetric diffusion algorithm, we could utilize the multigrid method on the surface hierarchy to obtain practical rendering rates.

Chapter 6

Conclusion

6.1 Summary

In this dissertation, we addressed the problem of efficient rendering of translucent objects. In particular, the direction of approximating light transport as a diffusion process was explored.

The main parts of this dissertation can be summarized as follows:

Fast Dipole Approximation The FDA [JMLH01a] significantly simplifies subsurface diffusion, applying a “local” solution (semi-infinite medium), to the case of arbitrary 3D geometry. Thereby, the problem is reduced to computing an integral over the object’s surface.

The FDA model enables us to interactively render translucent objects. We developed two efficient algorithms to realize this. The first is a hierarchical boundary method. The solution is discretized over all triangles of the object, which breaks up the surface integral into triangle integrals. To compute the latter, an efficient semi-analytical integration procedure was developed. Our experiments show that it is an improvement over a previously introduced point sampling scheme [JB02] in both performance and accuracy. The integration over the whole surface is performed hierarchically by clustering triangles. Interactions are dealt with in different levels in the hierarchy, thereby reducing the complexity for the integration from quadratic to log-linear.

The second method renders the local effect of subsurface scattering. It is applicable for materials with a small scattering response, such as human skin. We derive an exact importance sampling scheme for the FDA-based BSSRDF. This scheme is employed to perform local integration over the surface in image

space or in parameterized texture space, and can be implemented on modern consumer graphics hardware.

Compared to other methods, we endeavored to be both efficient and accurate at integrating the dipole BSSRDF surface integral. Our methods feature the ability to vary viewpoint, illumination, material parameters and to alter geometry, at interactive frame rates.

Finally, we analyzed the underlying assumptions in the FDA. From a phenomenological point-of-view, the FDA model cannot capture certain geometry-dependent effects, such as internal visibility. Also, it turns out that the law of energy conservation is violated. An alternative dipole placement scheme was introduced to enforce reciprocity.

Volumetric Simulation As an alternative to the less accurate FDA, we proposed a computational approach for light diffusion in arbitrary 3D objects. It consists of applying the multigrid method for solving diffusion on a regular volumetric grid. Two important issues are addressed. First, to achieve accurate representation of interactions near the object's surface, we apply the so-called embedded boundary discretization. Second, the solution adaptively refines where required in order to improve efficiency and keep memory requirements feasible. Our diffusion algorithm is accurate as far as the diffusion approximation allows. In comparison to the FDA, internal visibility is resolved correctly, and heterogeneities can be dealt with easily. Computation time is typically in the order of minutes.

Boundary Diffusion We introduced an exact model for subsurface diffusion, which operates only on the object's boundary. Basically, the diffusion equation is reformulated as a boundary integral equation, which shares similarities with the radiosity integral equation [CW93]. With the exception of the hierarchical evaluation, the BDM generalizes our boundary element method developed for the FDA (summarized above). Also, the polygon integration procedure can be reused.

As for practical results, we employed the BDM for validating the visual quality of the semi-infinite plane approximation, and confirmed the findings in our analysis of the FDA. The sheer number of methods based on the FDA that have appeared over the past few years, gives an impression of the popularity of this model. We feel that our validation provides some insight into why exactly the FDA is able to reproduce such plausible renderings, despite the crude underlying approximations.

6.2 Future Work

The local diffusion method described in section 3.4 can be extended in several ways. It would be interesting to take into account the importance of both the BSSRDF and the irradiance. In order to reduce noise artifacts, mipmap texture filtering can be applied for sampling more distant locations [Wil83, DS03]. The strengths of the hierarchical boundary method and the local diffusion technique are complementary to each other; it would be interesting to develop a hybrid method, which would reproduce the local response accurately, while being able to account for global diffusion effects.

The volumetric diffusion algorithm can benefit from a subdivision scheme which takes illumination and material variation at the surface into consideration. Also, more accurate source term computation is desirable.

We would like to develop a simple model capable of reproducing the missing phenomenological effects in the FDA. Preferably, it should be free of matrix inversion and restrict computations to the boundary. We believe that the BDM is a good starting point in looking for such a model.

In highly translucent materials, like ice, the optical depth may be small enough for low order scattering to become dominant. Consequently, the appearance will exhibit a pronounced dependence on the direction of illumination, and will be less suitable to be modeled by diffusion. We would like to develop interactive rendering models and methods for such cases, capable of dealing with varying illumination and geometry.

The BDM has not been made practical yet in terms of performance. In the same spirit as our volumetric diffusion algorithm, we could utilize the multi-grid method on a surface hierarchy to obtain practical rendering rates. We would like to extend the BDM to deal with composite materials, similar to Heino et al. [HASS03]. As such, we could simulate internal structure, and heterogeneities (e.g. veins in marble). Compared to our volumetric diffusion approach, this would enable us to easily render volumetric effects, without the costly maintenance of a hierarchical regular grid, and accounting for irregularities of the surface.

A thorough phenomenological investigation of the assumptions underlying the diffusion approximation is desirable. It would help us to understand the requirements for efficient rendering algorithms. This is particularly important, due to the paradox of on the one hand, the high computational complexity of light transport, and on the other hand, the qualitatively simple appearance.

Appendix A

Derivation of Modified Helmholtz Boundary Integral Equation

We briefly discuss the construction of the boundary integral equation (5.1), for completeness. It is analogous to the classic derivation for the Laplace operator (e.g., see Hunter et al. [HP02]).

Consider the 3D modified Helmholtz operator $L = \nabla^2 - \sigma_e^2$ in the diffusion equation (2.11). By integrating $L\{\phi\} = -\mathcal{S}$ over Ω , multiplied by a weight function w , one obtains the so-called “weak” formulation:

$$\int_{\Omega} w L\{\phi\} dV = - \int_{\Omega} w \mathcal{S} dV$$

Operator L 's free-space Green's function is defined as

$$L\{G(\mathbf{x}, \mathbf{y})\} = -\delta(\mathbf{x} - \mathbf{y}),$$

and is analytically known (see equation (2.19)). Filling in G as weight yields

$$- \int_{\Omega} G(r) L\{\phi(\mathbf{y})\} dV_{\mathbf{y}} = q(\mathbf{x}) \quad (\text{A.1})$$

where we denoted $q(\mathbf{x}) := \int_{\Omega} G(\mathbf{x}, \mathbf{y}) \mathcal{S}(\mathbf{y}) dV$

One now applies the second Green's identity, given by:

$$\begin{aligned} \int_{\Omega} g \nabla^2 f dV &= \int_{\Omega} f \nabla^2 g dV \\ &\quad - \int_{\Gamma} \left[g \frac{\partial f}{\partial n_y} - f \frac{\partial g}{\partial n_y} \right] dA \end{aligned}$$

This identity can be applied to the left hand side of (A.1) by substituting $\phi(\mathbf{y})$ for $f(\mathbf{y})$ and $G(\mathbf{x}, \mathbf{y})$ for $g(\mathbf{y})$. After a few simple rearrangements, we obtain:

$$\int_{\Omega} \phi [\nabla^2 - \sigma_e^2] G \, dV - \int_{\Gamma} [G \frac{\partial \phi}{\partial n_y} - \phi \frac{\partial G}{\partial n_y}] \, dA$$

The first term reduces to $\phi(\mathbf{x})$, by definition of the Green's function and the sifting property of δ .

We need to introduce a geometry-dependent weight $C(\mathbf{x})$, arising from the Green's function's singularity [HP02]:

$$C(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \text{ strictly inside } \Omega \\ \frac{1}{2} & \mathbf{x} \text{ on a smooth part of } \Gamma \\ \frac{\omega}{4\pi} & \mathbf{x} \text{ on a sharp part of } \Gamma \text{ with internal solid angle } \omega \end{cases}$$

Putting everything together, we obtain the BIE:

$$C(\mathbf{x})\phi(\mathbf{x}) + \int_{\Gamma} [\phi \frac{\partial G}{\partial n_y} - G \frac{\partial \phi}{\partial n_y}] \, dA = q(\mathbf{x})$$

Appendix B

Computing Form Factor F_{ii}^b

We derive the second “self” factor F_{ii}^b (see section 5.2.2). It is given by:

$$F_{ii}^b = \frac{1}{4\pi} \int_{\Gamma_i} (1 + r_i \sigma_e) \frac{e^{-\sigma_e r_i}}{r_i^3} (\mathbf{x}_i - \mathbf{y}) \cdot \vec{n}_y \, dA_y$$

Note the presence of the hypersingularity $1/r_i^3$.

Without loss of generality (which will be made clear later), we assume the element associated with F_{ii}^b is not a triangle, but a disc of radius R with midpoint \mathbf{x}_i . Consider a point $\mathbf{x}_{i,\epsilon}$, located directly above \mathbf{x}_i at a distance ϵ (along the normal direction). Let us integrate w.r.t. this point, i.e. employing $r_{i,\epsilon}^2 = r_i^2 + \epsilon^2$. Call the resulting form factor $F_{ii,\epsilon}^b$. Under these circumstances, $(\mathbf{x}_{i,\epsilon} - \mathbf{y}) \cdot \vec{n}_y = \epsilon$. Define $R_\epsilon := \sqrt{R^2 + \epsilon^2}$. Integration in polar coordinates now yields:

$$\begin{aligned} F_{ii,\epsilon}^b &= \frac{1}{2} \epsilon \int_0^R (1 + r_{i,\epsilon} \sigma_e) \frac{e^{-\sigma_e r_{i,\epsilon}}}{r_{i,\epsilon}^3} r_i \, dr_i \\ &= \frac{1}{2} \epsilon \int_\epsilon^{R_\epsilon} (1 + r_{i,\epsilon} \sigma_e) \frac{e^{-\sigma_e r_{i,\epsilon}}}{r_{i,\epsilon}^2} \, dr_{i,\epsilon} \\ &= \frac{1}{2} \left[e^{-\sigma_e \epsilon} - \frac{\epsilon e^{-\sigma_e R_\epsilon}}{R_\epsilon} \right] \end{aligned}$$

Let us now compute the second factor as:

$$F_{ii}^b = \lim_{\epsilon \rightarrow 0} F_{ii,\epsilon}^b = \frac{1}{2} - \underbrace{\lim_{\epsilon \rightarrow 0} \frac{\epsilon e^{-\sigma_e R_\epsilon}}{\sqrt{R^2 + \epsilon^2}}}_0$$

The same construction is possible for a triangle, except we would keep the angular integration $\int \dots d\theta$ throughout the derivation, while denoting dependence of radius R on θ . This does not affect the result.

Appendix C

Publications

The work described in the following papers is discussed in this dissertation.

- *Interactive Rendering of Translucent Deformable Objects* [MKB⁺03b]
Tom Mertens, Jan Kautz, Philippe Bekaert, Frank Van Reeth, Hans-Peter Seidel
In proceedings of Eurographics Symposium on Rendering 2003
- *Efficient Rendering of Local Subsurface Scattering* [MKB⁺03a]
Tom Mertens, Jan Kautz, Philippe Bekaert, Hans-Peter Seidel, Frank Van Reeth
In proceedings of Pacific Graphics 2003
- *A Computational Approach to Simulate Subsurface Light Diffusion in Arbitrarily Shaped Objects* [HMBR05]
Tom Haber, Tom Mertens, Philippe Bekaert, Frank Van Reeth
In proceedings of Graphics Interface 2005
- *Efficient Rendering of Local Subsurface Scattering* [MKB⁺05]
Tom Mertens, Jan Kautz, Philippe Bekaert, Hans-Peter Seidel, Frank Van Reeth
In March 2005 issue of Computer Graphics Forum (extended version of the Pacific Graphics paper)

The following work is not part of this dissertation:

- *An Image-Based Radiosity Representation* [MR01]
Tom Mertens and Frank Van Reeth
In proceedings of Image and Vision Computing New Zealand 2001

- *A Self-Shadow Algorithm for Dynamic Hair using Density Clustering*
[MKBR04]
Tom Mertens, Jan Kautz, Philippe Bekaert, Frank Van Reeth
In proceedings of Eurographics Symposium on Rendering 2004

Bibliography

- [App85] A.A. Appel. An efficient program for many-body simulations. *SIAM Journal on Scientific and Statistical Computing*, 16(1):85–103, 1985.
- [Aro97] Raphael Aronson. Radiative transfer implies a modified reciprocity relation. *J. Opt. Soc. Am. A*, 14(2):486–490, February 1997.
- [ati] ATI technologies inc. <http://www.ati.com>.
- [BBSW94] J. Bell, M. J. Berger, J. Saltzman, and M. Welcome. Three dimensional adaptive mesh refinement for hyperbolic conservation laws. In *J. Sci. Comput.*, pages 127–138, 1994.
- [BL03] George Borshukov and J.P. Lewis. Realistic human face rendering for “the matrix reloaded”, 2003. Siggraph 2003 Technical Sketch.
- [BO84] M. Berger and J. Olinger. Adaptive mesh refinement for hyperbolic partial differential equations. In *Journal of Computational Physics*, pages 484–512, 1984.
- [BW59] Max Born and Emil Wolf. *Principles of Optics*. Pergamon Press, New York, 1959.
- [Cha60] Subrahmanyan Chandrasekhar. *Radiative Transfer*. Dover Pubns, 1960.
- [CHH03] Nathan A. Carr, Jesse D. Hall, and John C. Hart. Gpu algorithms for radiosity and subsurface scattering. In *HWWS '03: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 51–59, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

- [CTW⁺04] Yanyun Chen, Xin Tong, Jiaping Wang, Stephen Lin, Baining Guo, and Heung-Yeung Shum. Shell texture functions. *ACM Trans. Graph.*, 23(3):343–353, 2004.
- [CW93] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, Cambridge, MA, 1993.
- [DBB03] Philip Dutré, Philippe Bekaert, and Kavita Bala. *Advanced Global Illumination*. AK Peters Ltd., 2003.
- [DCL⁺98] M. Day, P. Colella, M. Lijewski, C. Rendleman, and D. Marcus. Embedded boundary algorithms for solving the poisson equation on complex domains, 1998. Technical Report Lawrence Berkeley National Laboratory, LBNL-41811.
- [Deb] Paul Debevec. <http://www.debevec.org>.
- [DEJ⁺99] Julie Dorsey, Alan Edelman, Henrik Wann Jensen, Justin Legakis, and Hans Kuhling Pedersen. Modeling and rendering of weathered stone. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 225–234. ACM Press/Addison-Wesley Publishing Co., 1999.
- [DEL⁺99] Julie Dorsey, Alan Edelman, Justin Legakis, Henrik Wann Jensen, and Hans Køhling Pedersen. Modeling and Rendering of Weathered Stone. In *Proceedings of SIGGRAPH 99*, pages 225–234, 1999.
- [DeZ93] D. DeZeeuw. A quadtree-based adaptively refined cartesian-grid algorithm for solution of the euler equations, 1993.
- [DLG90] Nira Dyn, David Levin, and John A. Gregory. A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control. *ACM Transactions on Graphics*, 9(2):160–169, April 1990.
- [DS03] Carsten Dachsbacher and Marc Stamminger. Translucent shadow maps. In *Proceedings of Eurographics Symposium on Rendering*, pages 197–201, jun 2003.
- [EVNT78] G. Eason, A. Veitch, R. Nisbet, and F. Turnbull. The theory of the backscattering of light by blood. *J. Physics*, 11:1463–1479, 1978.

- [far] FarField Technology. <http://www.farfieldtechnology.com>.
- [FDA03] R. W. Fleming, R. O. Dror, and E. H. Adelson. Real-world illumination and the perception of surface reflectance properties. *Journal of Vision*, 3(5):347–368, 2003.
- [Fer98] J.A. Ferwerda. Fundamentals of spatial vision. pages 1–27, 1998. Course 32, SIGGRAPH '98.
- [Fic55] M. Fick. über diffusion. *Ann. Physik und Chemie*, 94, 1855.
- [FJB04] Roland W. Fleming, Henrik Wann Jensen, and Heinrich H B. Perceiving translucent materials. In *Proceedings of the 1st Symposium on Applied perception in graphics and visualization*, pages 127–134. ACM Press, 2004.
- [FPW92] T. J. Farrell, M. S. Patterson, and B. Wilson. A diffusion theory model of spatially resolved, steady-state diffuse reflectance for the noninvasive determination of tissue optical properties in vivo. *Med. Phys.*, 19:879–888, 1992.
- [FYT94] David A. Forsyth, Chien Yang, and Kim Teo. Efficient Radiosity in Dynamic Environments. In *Fifth Eurographics Workshop on Rendering*, pages 313–323, June 1994.
- [GFB83] R. A. Groenhuis, H. A. Ferwerda, and J. J. Ten Bosch. Scattering and absorption of turbid materials determined from reflection measurements. 1. theory. *Applied Optics*, 22:2456–2462, 1983.
- [GLL⁺04] Michael Goesele, Hendrik P. A. Lensch, Jochen Lang, Christian Fuchs, and Hans-Peter Seidel. Disco: acquisition of translucent objects. *ACM Trans. Graph.*, 23(3):835–844, 2004.
- [Gou71] H. Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computers*, 20(6):623–628, 1971.
- [Gre86] Ned Greene. Applications of world projections. In *Graphics Interface 86*, pages 108–114, May 1986.
- [GWH01] Michael Garland, Andrew Willmott, and Paul S. Heckbert. Hierarchical Face Clustering on Polygonal Surfaces. In *Proceedings 2001 ACM Symposium on Interactive 3D Graphics*, pages 49–58, March 2001.

- [Hab03] Richard Haberman. *Applied Partial Differential Equations*. Prentice Hall, fourth edition, 2003.
- [Hac85] Wolfgang Hackbusch. *Multigrid Methods and Applications*. Springer-Verlag, Berlin, 1985.
- [HAP05] Kyle Hegeman, Michael Ashikhmin, and Simon Premože. A lighting model for general participating media. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, April 2005.
- [HASS03] J. Heino, S. Arridge, J. Sikora, and E. Somersalo. Anisotropic effects in highly scattering media. *Physical Review E*, 68(3), September 2003.
- [HBV03] Xuejun Hao, Thomas Baby, and Amitabh Varshney. Interactive Subsurface Scattering for Translucent Meshes. In *Proceedings 2003 ACM Symposium on Interactive 3D Graphics*, pages 75–82, april 2003.
- [HG41] L.G. Henyey and J.L. Greenstein. Diffuse radiation in the galaxy. *Astrophysical Journal*, 93:70–83, 1941.
- [Hic97] A.H. Hielscher. Model-based iterative image reconstruction for photon migration tomography. In R.L. Barbour, M.J. Carvlin, and M.A. Fiddy, editors, *Methods for Solving Ill-Posed Inverse Imaging Problems: Medical and Nonmedical Applications*, SPIE-Proc., volume 3171, pages 106–117, 1997.
- [HK93] Pat Hanrahan and Wolfgang Krueger. Reflection from layered surfaces due to subsurface scattering. In *Proceedings of SIGGRAPH 93*, pages 165–174, 1993.
- [HMBR05] Tom Haber, Tom Mertens, Philippe Bekaert, and Frank Van Reeth. A computational approach to simulate subsurface light diffusion in arbitrarily shaped objects. In *Proceedings of Graphics Interface*, May 2005.
- [HP02] Peter Hunter and Andrew Pullan. FEM/BEM notes, 2002. Available through <http://www.bioeng.auckland.ac.nz>.
- [HSA91] Pat Hanrahan, David Salzman, and Larry Aupperle. A Rapid Hierarchical Radiosity Algorithm. *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25(4):197–206, July 1991.

- [HV04] Xuejun Hao and Amitabh Varshney. Real-time rendering of translucent meshes. *ACM Trans. Graph.*, 23(2):120–142, 2004.
- [Ish78] Akira Ishimaru. "Wave Propagation and Scattering in Random Media", volume 1. Academic Press, 1978.
- [JB02] Henrik Wann Jensen and Juan Buhler. A Rapid Hierarchical Rendering Technique for Translucent Materials. *ACM Transactions on Graphics*, 21(3):576–581, July 2002.
- [JC98] H. Johansen and P. Colella. A cartesian grid embedded boundary method for poisson's equation on irregular domains. In *J. Comput. Phys.* 147, 60, 1998.
- [Jen96] Henrik Wann Jensen. Global Illumination Using Photon Maps. In *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, pages 21–30, New York, NY, 1996. Springer-Verlag/Wien.
- [Jen01] Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping*. A.K. Peters Ltd., 2001.
- [JMLH01a] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A Practical Model for Subsurface Light Transport. In *Proceedings of SIGGRAPH 2001*, pages 511–518, August 2001.
- [JMLH01b] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A Practical Model for Subsurface Light Transport. In *Proceedings of SIGGRAPH 2001*, pages 511–518, August 2001.
- [Kaj86] James T. Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150. ACM Press, 1986.
- [KH01] Alexander Keller and Wolfgang Heidrich. Interleaved sampling. In *Rendering Techniques 2001: 12th Eurographics Workshop on Rendering*, pages 269–276, June 2001.
- [KM31] P. Kubelka and F. Munk. Ein beitrage zur optik der farbanstriche. *Z. Tech. Physik.*, 12:593–601, 1931.

- [KPHE02] Joe Kniss, Simon Premoze, Charles Hansen, and David Ebert. Interactive translucent volume rendering and procedural modeling. In *Proceedings of Visualization 2002*, 2002.
- [Kv01] Jan J. Koenderink and Andrea J. van Doorn. Shading in the Case of Translucent Objects. In *Human Vision and Electronic Imaging VI*, pages 312–320. SPIE, 2001.
- [KVLS99] Leif P. Kobbelt, Jens Vorsatz, Ulf Labsik, and Hans-Peter Seidel. A Shrink Wrapping Approach to Remeshing Polygonal Surfaces. *Computer Graphics Forum*, 18(3):119–130, September 1999.
- [KW86] Malvin H. Kalos and Paula A. Whitlock. *Monte Carlo methods. Vol. 1: basics*. Wiley-Interscience, New York, NY, USA, 1986.
- [LGB⁺02] Hendrik P. A. Lensch, Michael Goesele, Philippe Bekaert, Jan Kautz, Marcus A. Magnor, Jochen Lang, and Hans-Peter Seidel. Interactive Rendering of Translucent Objects. In *Proceedings of Pacific Graphics 2002*, pages 214–224, October 2002.
- [MKB⁺03a] Tom Mertens, Jan Kautz, Philippe Bekaert, Frank Van Reeth, and Hans-Peter Seidel. Efficient rendering of local subsurface scattering. In *Proceedings of Pacific Graphics*, pages 51–58, October 2003.
- [MKB⁺03b] Tom Mertens, Jan Kautz, Philippe Bekaert, Frank Van Reeth, and Hans-Peter Seidel. Interactive rendering of translucent deformable objects. In *Proceedings of Eurographics Symposium on Rendering*, pages 130–140, June 2003.
- [MKB⁺05] Tom Mertens, Jan Kautz, Philippe Bekaert, Frank Van Reeth, and Hans-Peter Seidel. Efficient rendering of local subsurface scattering. *Computer Graphics Forum*, 24(1):41–39, March 2005.
- [MKBR04] Tom Mertens, Jan Kautz, Philippe Bekaert, and Frank Van Reeth. A self-shadow algorithm for dynamic hair using density clustering. In *Proceedings of Eurographics Symposium on Rendering*, pages 173–178, June 2004.
- [MPBM03] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Trans. Graph.*, 22(3):759–769, 2003.

- [MR01] Tom Mertens and Frank Van Reeth. An image-based radiosity representation. In *Proceedings of Image and Vision Computing New Zealand*, pages 7–1, November 2001.
- [MSS88] Keijzer Marleen, Willem M. Star, and Pascal R. M. Storch. Optical diffusion in layered media. *Applied Optics*, 27(9):1820–1824, May 1988.
- [Nie92] H. Niederreiter. Random number generation and quasi-monte carlo methods. *CBMS-NSF Regional Conference Series in Applied Math.*, 63, 1992.
- [NRH⁺77] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. Geometrical considerations and nomenclature for reflectance. 1977. Mono. 160.
- [nvi] nVIDIA corporation. <http://www.nvidia.com>.
- [PAS03] Simon Premože, Michael Ashikhmin, and Peter Shirley. Path integration for light transport in volumes. In P. Christensen and D. Cohen-Or, editors, *Proceedings of the 14th Eurographics Workshop on Rendering*, pages 52–63, Aire-la-Ville, Switzerland, June 25–27 2003. Eurographics Association.
- [PAT⁺04] Simon Premože, Michael Ashikhmin, Jerry Tessendorf, Ravi Ramamoorthi, and Shree Nayar. Practical rendering of multiple scattering effects in participating media. In *Proceedings of the 14th Eurographics Workshop on Rendering*, 2004.
- [PCPS97] Frederic Perez-Cazorla, Xavier Pueyo, and Francois Sillion. Global illumination techniques for the simulation of participating media. In *Proceedings of the Eighth Eurographics Workshop on Rendering*, June 1997.
- [Per85] Ken Perlin. An image synthesizer. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 287–296. ACM Press, 1985.
- [PH00] Matt Pharr and Pat Hanrahan. Monte carlo evaluation of non-linear scattering equations for subsurface reflection. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 75–84. ACM Press/Addison-Wesley Publishing Co., 2000.

- [Pha01] Matt Pharr, 2001. Siggraph 2001 Course notes, organized by Larry Gritz.
- [Pho75] Bui-Tong Phong. Illumination for computer-generated pictures. *Communications of the ACM*, 18(6):311–317, June 1975.
- [PKK00] Mark Pauly, Thomas Kollig, and Alexander Keller. Metropolis light transport for participating media. In B. Peroche and H. Rushmeier, editors, *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)*, pages 11–22, New York, NY, 2000. Springer Wien.
- [Pop03] Stéphane Popinet. Gerris: a tree-based adaptive solver for the incompressible euler equations in complex geometries. *J. Comput. Phys.*, 190(2):572–600, 2003.
- [PTVF92] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C: The Art of Scientific Computing (2nd ed.)*. Cambridge University Press, 1992. ISBN 0-521-43108-5.
- [RH01] Ravi Ramamoorthi and Pat Hanrahan. An Efficient Representation for Irradiance Environment Maps. In *Proceedings of SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 497–500, August 2001.
- [Rip00] Jorge Ripoll. *Light Diffusion in Turbid Media with Biomedical Application*. PhD thesis, Universidad Autonoma of Madrid, 2000.
- [RSC87] William T. Reeves, David H. Salesin, and Robert L. Cook. Rendering antialiased shadows with depth maps. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 283–291, New York, NY, USA, 1987. ACM Press.
- [RSSF02] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. *ACM Transactions on Graphics*, 21(3):267–276, July 2002.
- [SAG94] Brian Smits, James Arvo, and Donald Greenberg. A Clustering Algorithm for Radiosity in Complex Environments. *Computer Graphics*, 28(Annual Conference Series):435–442, July 1994.

- [San04] Pedro V. Sander. Real-time skin rendering on graphics hardware, 2004. Siggraph 2004 Technical Sketch.
- [SHHS03] Peter-Pike Sloan, Jesse Hall, John Hart, and John Snyder. Clustered Principal Components for Precomputed Radiance Transfer. In *Proceedings of SIGGRAPH 2003*, pages 382–391, July 2003.
- [Sil95] Francois X. Sillion. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):240–254, 1995.
- [SKS02] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments. In *Proceedings of SIGGRAPH 2002*, pages 527–536, July 2002.
- [Sta95] Jos Stam. Multiple scattering as a diffusion process. In *Eurographics Rendering Workshop 1995*, pages 41–50, 1995.
- [su3] The stanford 3d scanning repository. <http://graphics.stanford.edu/data/3Dscanrep/>.
- [Tur91] Greg Turk. Generating textures on arbitrary surfaces using reaction-diffusion. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 289–298, New York, NY, USA, 1991. ACM Press.
- [TZL⁺02] X. Tong, J. Zhang, L. Liu, X. Wang, B. Guo, and H.-Y. Shum. Synthesis of bidirectional texture functions on arbitrary surfaces. *ACM Transactions on Graphics*, 21(3):665–672, July 2002.
- [VG95] Eric Veach and Leonidas J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of SIGGRAPH 95*, Computer Graphics Proceedings, Annual Conference Series, pages 419–428, August 1995.
- [WHG99] Andrew Willmott, Paul S. Heckbert, and Michael Garland. Face Cluster Radiosity. In *Eurographics Workshop on Rendering 1999*, pages 293–304, June 1999.
- [Whi80] Turner Whitted. An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349, 1980.

- [Wil78] Lance Williams. Casting curved shadows on curved surfaces. *Computer Graphics (SIGGRAPH '78 Proceedings)*, 12(3):270–274, August 1978.
- [Wil83] Lance Williams. Pyramidal parametrics. In *SIGGRAPH '83: Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, pages 1–11, New York, NY, USA, 1983. ACM Press.
- [WTL05] Rui Wang, John Tran, and David Luebke. All-frequency interactive relighting of translucent objects with single and multiple scattering. In *Proceedings of Siggraph'05*, 2005.
- [ZSS96] Denis Zorin, Peter Schröder, and Wim Sweldens. Interpolating Subdivision for Meshes with Arbitrary Topology. In *Proceedings of SIGGRAPH 96*, pages 189–192, August 1996.

