Efficient Methods for Physically-based Rendering of Participating Media

Dissertation zur Erlangung des Grades des
Doktors der Ingenieurwissenschaften der
Naturwissenschaftlich-Technischen Fakultäten der
Universität des Saarlandes

Vorgelegt durch

Oskar Elek

am Mai 2015 in Saarbrücken, Germany

Betreuender Hochschullehrer – Advisor:
Prof. Dr. Hans-Peter Seidel

Gutachter – Reviewers:
Prof. Dr. Hans-Peter Seidel
Dr. Tobias Ritschel
Prof. Dr.-Ing. Carsten Dachsbacher

Kolloquium – Examination

Datum – Date:
2016 - 02 - 02

Dekan – Dean:
Prof. Dr. Markus Bläser

Vorsitzender – Chair:
Prof. Dr. Joachim Weickert

Prüfer – Examiners:
Prof. Dr. Hans-Peter Seidel
Dr. Tobias Ritschel
Prof. Dr.-Ing. Carsten Dachsbacher

Protokoll – Reporter:
Dr. Dan Casas

Abstract

This thesis proposes several novel methods for realistic synthesis of images containing participating media. This is a challenging problem, due to the multitude and complexity of ways how light interacts with participating media, but also an important one, since such media are ubiquitous in our environment and therefore are one of the main constituents of its appearance. The main paradigm we follow is designing efficient methods that provide their user with an interactive feedback, but are still physically plausible.

The presented contributions have varying degrees of specialisation and, in a loose connection to that, their resulting efficiency. First, the *screen-space scattering* algorithm simulates scattering in homogeneous media, such as fog and water, as a fast image filtering process. Next, the *amortised photon mapping* method focuses on rendering clouds as arguably one of the most difficult media due to their high scattering anisotropy. Here, interactivity is achieved through adapting to certain conditions specific to clouds. A generalisation of this approach is *principal-ordinates propagation*, which tackles a much wider class of heterogeneous media. The resulting method can handle almost arbitrary optical properties in such media, thanks to a custom finite-element propagation scheme. Finally, *spectral ray differentials* aim at an efficient reconstruction of chromatic dispersion phenomena, which occur in transparent media such as water, glass and gemstones. This method is based on analytical ray differentiation and as such can be incorporated to any ray-based rendering framework, increasing the efficiency of reproducing dispersion by about an order of magnitude.

All four proposed methods achieve efficiency primarily by utilising high-level mathematical abstractions, building on the understanding of the underlying physical principles that guide light transport. The methods have also been designed around simple data structures, allowing high execution parallelism and removing the need to rely on any sort of preprocessing. Thanks to these properties, the presented work is not only suitable for interactively computing light transport in participating media, but also allows dynamic changes to the simulated environment, all while maintaining high levels of visual realism.

Kurzzusammenfassung

Diese Arbeit stellt mehrere neue Verfahren zum Erzeugen von realistischen Bildern vor, die trübe Medien enthalten. Dabei handelt es sich aufgrund der vielfältigen und komplexen Art und Weise, wie Licht mit trüben Medien interagiert, um ein anspruchsvolles Problem. Gleichzeitig ist es ein Problem von großer Bedeutung, da diese Medien in unserer Umgebung allgegenwärtig sind und daher maßgeblich zum ihrem Erscheinungsbild beitragen. Unser Hauptparadigma ist der Entwurf von effizienten Methoden, die ihrem Anwender interaktives Feedback ermöglichen, dabei aber physikalisch plausibel bleiben.

Die vorgestellten Beiträge haben unterschiedlich stark ausgeprägte Spezialisierungsgrade und sind, damit einhergehend, unterschiedlich effizient. Zuerst bildet der *Screen-Space Scattering*-Algorithmus Streuung in homogenen Medien wie Nebel und Wasser mit Hilfe eines schnellen Bildfiltervorgangs nach. Die *Amortised Photon Mapping*-Methode beschäftigt sich dann mit dem Rendern von Wolken, einem Medium, das wegen seiner großen Streuungsanisotropie wohl zu den kompliziertesten gehört. Dabei wird Interaktivität durch das Anpassen an bestimmte Beschaffenheiten von Wolken erzielt. Eine Verallgemeinerung dieses Ansatzes ist *Principal Ordinates Propagation*, das eine weitaus größere Kategorie heterogener Medien angeht. Die resultierende Methode kann dank eines speziellen Finite-Elemente-Ausbreitungsschemas mit nahezu beliebigen optischen Eigenschaften solcher Medien umgehen. Abschließend sind *Spectral Ray Differentials* auf eine effiziente Rekonstruktion von Farbdispersionsphänomenen ausgerichtet, die in transparenten Medien wie Wasser, Glas oder Edelsteinen auftreten. Diese Methode basiert auf analytischer Strahlendifferenziation und kann als solche in jedes strahlenbasierte Rendering-Framework eingegliedert werden, was beim Nachbilden von Dispersionen zu einer Effizienzsteigerung von etwa einer Größenordnung führt.

Alle vier vorgestellten Methoden verdanken ihre Effizienz hauptsächlich der Verwendung von höheren mathematischen Abstraktionen, für die Kenntnisse der zugrunde liegenden physikalischen Prinzipien des Lichttransports das Fundament liefern. Die Methoden wurden außerdem so entworfen, dass sie mit einfachen Datenstrukturen auskommen, was eine in hohem Maße parallele Ausführung erlaubt und dazu führt, dass auf Vorberechnungen komplett verzichtet werden kann. Dank dieser Eigenschaften ist die vorliegende Arbeit nicht nur dazu geeignet, Lichttransport in trüben Medien interaktiv zu berechnen, sondern erlaubt auch dynamische Veränderungen der simulierten Szenerie. Das hohe Niveau des visuellen Realismus' wird dabei jederzeit beibehalten.

Summary

This thesis proposes several novel methods for realistic synthesis of images containing participating media. This is a challenging problem, due to the multitude and complexity of ways how light interacts with participating media, but also an important one, since such media are ubiquitous in our environment and therefore are one of the main constituents of its appearance. The main paradigm we follow is designing efficient methods that provide their user with an interactive feedback, but are still physically plausible.

The presented contributions have varying degrees of specialisation and, in a loose connection to that, their resulting efficiency. First, the *screen-space scattering* algorithm simulates scattering in homogeneous media, such as fog and water, as a fast image filtering process. Next, the *amortised photon mapping* method focuses on rendering clouds as arguably one of the most difficult media due to their high scattering anisotropy. Here, interactivity is achieved through adapting to certain conditions specific to clouds. A generalisation of this approach is *principal-ordinates propagation*, which tackles a much wider class of heterogeneous media. The resulting method can handle almost arbitrary optical properties in such media, thanks to a custom finite-element propagation scheme. Finally, *spectral ray differentials* aim at an efficient reconstruction of chromatic dispersion phenomena, which occur in transparent media such as water, glass and gemstones. This method is based on analytical ray differentiation and as such can be incorporated to any ray-based rendering framework, increasing the efficiency of reproducing dispersion by about an order of magnitude.

The text is structured as follows. First, the topic and motivation are introduced in Chapter 1, followed by a review of the relevant theoretical and methodological background in Chapter 2. The core contributions of the thesis are then detailed in Chapters 3–6. Finally, the summary and conclusions of the work are discussed in Chapter 7, including an overview of possible future outlooks.

Screen-Space Scattering Chapter 3 presents an approximate algorithm for computing light scattering within homogeneous participating environments in image space. Instead of simulating the full global illumination in participating media, the scattering process is modelled by a physically-based point spread function. This function is applied efficiently by performing a discrete hierarchical convolution in a texture MIP map. The main problem of this approach, 'illumination leaking', is dealt with by designing a custom anisotropic incremental filter. The algorithm is fully parallel, runs in hundreds of frames-per-second for usual screen resolutions and is directly applicable in most existing 2D or 3D rendering architectures.

Amortised Photon Mapping Chapter 4 presents a novel interactive algorithm for simulating light transport in clouds. Exploiting the high temporal coherence of the typical illumination and morphology of clouds the method builds on volumetric photon mapping, which is modified to allow for interactive rendering speeds – instead of building a fresh irregular photon map for every scene state change, photon contributions are accumulated in a regular grid structure. This is then

continuously being refreshed by re-shooting only a fraction of the total amount of photons in each frame. To maintain temporal coherence and low variance, a low-resolution grid is initially computed, and then upsampled to the density field resolution on a physical basis in each frame. Another key contribution is a technique to store and reconstruct the angular illumination information by exploiting properties of the standard Henyey-Greenstein function, namely its ability to express anisotropic angular distributions with a single dominant direction. The resulting method is physically-plausible, conceptually simple and comparatively easy to implement. Moreover, it operates only above the cloud density field, thus not requiring any precomputation.

Principal-Ordinates Propagation Chapter 5 proposes a novel finite-element method for rendering participating media with multiple anisotropic scattering, achieving real-time speeds on modern GPUs without imposing any significant restrictions on the rendered medium. This is achieved by dynamically decomposing all illumination into directional and point light sources, and propagating the light from these virtual sources in independent discrete propagation domains. These are individually aligned with approximate principal directions of light propagation from the respective light sources. Such decomposition allows to use a very simple and computationally efficient unimodal basis for representing the propagated radiance, instead of using a general basis such as spherical harmonics. The resulting approach is biased but physically plausible, and largely reduces the rendering artefacts inherent to all existing finite-element methods. At the same time it allows for virtually arbitrary scattering anisotropy, albedo, and other properties of the simulated medium, again without requiring any precomputation.

Spectral Ray Differentials Light refracted by a dispersive interface leads to beautifully coloured patterns that can be rendered faithfully with spectral Monte-Carlo methods. Regrettably, results often suffer from chromatic noise or banding, requiring high sampling rates and large amounts of memory compared to renderers operating in some trichromatic colour space. Addressing this issue, Chapter 6 introduces spectral ray differentials, which describe the change of light direction with respect to changes in the spectrum. In analogy with the classic ray and photon differentials, this information can be used for filtering spectral ray properties in the spatial domain. The effectiveness of this approach is demonstrated by filtering in offline spectral light and path tracing as well as for an interactive GPU photon mapper based on splatting. Results show considerably less chromatic noise and spatial aliasing while consistently converging to reference solutions, with negligible tracing overhead in the order of milliseconds.

All four proposed methods achieve efficiency primarily by utilising high-level mathematical abstractions, building on the understanding of the underlying physical principles that guide light transport. The methods have also been designed around simple data structures, allowing high execution parallelism and removing the need to rely on any sort of preprocessing. Thanks to these properties, the presented work is not only suitable for interactively computing light transport in participating media, but also allows dynamic changes to the simulated environment, all while maintaining high levels of visual realism.

Zusammenfassung

Diese Arbeit stellt mehrere neue Verfahren zum Erzeugen von realistischen Bildern vor, die trübe Medien enthalten. Dabei handelt es sich aufgrund der vielfältigen und komplexen Art und Weise, wie Licht mit trüben Medien interagiert, um ein anspruchsvolles Problem. Gleichzeitig ist es ein Problem von großer Bedeutung, da diese Medien in unserer Umgebung allgegenwärtig sind und daher maßgeblich zum ihrem Erscheinungsbild beitragen. Unser Hauptparadigma ist der Entwurf von effizienten Methoden, die ihrem Anwender interaktives Feedback ermöglichen, dabei aber physikalisch plausibel bleiben.

Die vorgestellten Beiträge haben unterschiedlich stark ausgeprägte Spezialisierungsgrade und sind, damit einhergehend, unterschiedlich effizient. Zuerst bildet der *Screen-Space Scattering*-Algorithmus Streuung in homogenen Medien wie Nebel und Wasser mit Hilfe eines schnellen Bildfiltervorgangs nach. Die *Amortised Photon Mapping*-Methode beschäftigt sich dann mit dem Rendern von Wolken, einem Medium, das wegen seiner großen Streuungsanisotropie wohl zu den kompliziertesten gehört. Dabei wird Interaktivität durch das Anpassen an bestimmte Beschaffenheiten von Wolken erzielt. Eine Verallgemeinerung dieses Ansatzes ist *Principal Ordinates Propagation*, das eine weitaus größere Kategorie heterogener Medien angeht. Die resultierende Methode kann dank eines speziellen Finite-Elemente-Ausbreitungsschemas mit nahezu beliebigen optischen Eigenschaften solcher Medien umgehen. Abschließend sind *Spectral Ray Differentials* auf eine effiziente Rekonstruktion von Farbdispersionsphänomenen ausgerichtet, die in transparenten Medien wie Wasser, Glas oder Edelsteinen auftreten. Diese Methode basiert auf analytischer Strahlendifferenziation und kann als solche in jedes strahlenbasierte Rendering-Framework eingegliedert werden, was beim Nachbilden von Dispersionen zu einer Effizienzsteigerung von etwa einer Größenordnung führt.

Der Text ist wie folgt gegliedert. Zunächst werden in Kapitel 1 Thema und Motivation erläutert, gefolgt von einer Besprechung relevanter theoretischer und methodologischer Hintergründe in Kapitel 2. Die Hauptbeiträge dieser Arbeit werden daraufhin in den Kapiteln 3–6 ausführlich beschrieben. Zum Schluss werden in Kapitel 7 eine Zusammenfassung und Schlussfolgerungen gegeben, einschließlich einer Übersicht möglicher zukünftiger Ausblicke.

Screen-Space Scattering Kapitel 3 stellt einen approximativen Algorithmus zur Berechnung von Lichtstreuung in homogenen trüben Umgebungen vor, der im Bildraum operiert. Statt die globale Beleuchtung in trüben Medien vollständig nachzubilden, wird der Streuungsvorgang mit Hilfe einer physikalisch motivierten Punktbildfunktion modelliert. Diese Funktion wird effizient angewendet, indem eine diskrete hierarchische Faltung in einer MIP-Map-Textur durchgeführt wird. Das Hauptproblem dieser Vorgehensweise, ‘Lichtlecks’, wird durch die Konstruktion eines problemspezifischen anisotropen inkrementellen Filters behandelt. Der Algorithmus ist vollkommen parallel, verarbeitet bei gängigen Bildschirmauflösungen Hunderte von Bildern pro Sekunde und ist in die meisten bestehenden 2D oder 3D Rendering-Architekturen unmittelbar einsetzbar.

Amortised Photon Mapping Kapitel 4 stellt einen neuartigen interaktiven Algorithmus zur Nachbildung des Lichttransports in Wolken vor. Die hohe zeitliche Kohärenz der in Wolken zu beobachtenden Beleuchtung und Morphologie ausnutzend, basiert die Methode auf volumetrischem Photon Mapping, das abgewandelt wurde, um interaktive Renderzeiten zu ermöglichen – statt bei jeder Veränderung des Szenenzustands eine neue unregelmäßige Photon Map zu erstellen, werden die Beiträge der Photonen in einer regelmäßigen Rasterstruktur akkumuliert. Diese wird dann fortlaufend aktualisiert, indem pro Einzelbild nur ein Teil der Photonen neu in die Szene geschossen wird. Um zeitliche Kohärenz und eine geringe Varianz beizubehalten, wird anfangs ein Raster mit geringer Auflösung berechnet. Dann findet ein physikalisch basiertes Upsampling auf die Dichtefeldauflösung in jedem Einzelbild statt. Ein weiterer Hauptbeitrag ist ein Verfahren, um die winkelabhängige Beleuchtungsinformation zu speichern und wiederherzustellen. Dabei werden Eigenschaften der gewöhnlichen Henyey-Greenstein Funktion ausgenutzt, genauer gesagt ihr Vermögen, anisotrope Winkelverteilungen mit Hilfe einer einzigen dominanten Richtung auszudrücken. Die resultierende Methode ist physikalisch plausibel, konzeptionell einfach und vergleichsweise leicht zu implementieren. Darüber hinaus operiert sie ausschließlich über dem Wolkendichtefeld und ist deshalb auf keinerlei Vorberechnungen angewiesen.

Principal-Ordinates Propagation Kapitel 5 schlägt eine neuartige Finite-Elemente-Methode zum Rendern trüber Medien mit mehrfacher anisotroper Streuung vor. Dabei werden auf modernen GPUs Echtzeitgeschwindigkeiten erzielt, ohne dem gerenderten Medium wesentliche Einschränkungen aufzuerlegen. Dies wird erreicht, indem die gesamte Beleuchtung dynamisch in gerichtete und Punktlichtquellen zerlegt und Licht von diesen virtuellen Quellen in unabhängige diskrete Ausbreitungsbereiche propagiert wird. Diese Bereiche sind einzeln an näherungsweise Hauptrichtungen des Lichttransports der jeweiligen Lichtquellen angeglichen. Eine solche Zerlegung lässt die Verwendung einer sehr einfachen und recheneffizienten unimodalen Basis zu, um die propagierte Strahlendichte darzustellen, statt eine allgemeine Basis, wie beispielsweise Kugelflächenfunktionen, zu verwenden. Die resultierende Vorgehensweise ist biased aber physikalisch plausibel und verringert großenteils die Rendering-Artefakte, die allen Finite-Elemente-Methoden zu eigen sind. Gleichzeitig erlaubt die Methode nahezu beliebige Streuungsanisotropien, Albedos und andere Eigenschaften des simulierten Mediums, wiederum ohne Vorberechnungen zu benötigen.

Spectral Ray Differentials Licht, das von einem streuenden Gegenstand gebrochen wird, führt zu visuell ansprechenden farbigen Mustern, die mit spektralen Monte-Carlo-Methoden wirklichkeitsgetreu gerendert werden können. Bedauerlicherweise leiden die Ergebnisse oft unter Farbrauschen oder Banding, weshalb im Vergleich zu Renderern, die in einem trichromatischen Farbraum operieren, hohe Abtastraten und große Speichermengen benötigt werden. Kapitel 6 beschäftigt sich mit diesem Problem und führt dabei Spectral Ray Differentials ein, welche die Veränderung der Lichtrichtung bezüglich der Veränderungen im Spektrum beschreiben. Diese Information kann, analog zu klassischen Strahlen- und Photonendifferenzialen, dazu benutzt werden, spektrale Strahleneigenschaften im Ortsraum zu filtern. Die Effektivität dieses Ansatzes wird sowohl mittels Filterung im spektralen Offline-Light- und Path-Tracing, als auch im Rahmen eines interaktiven Splatting-basierten GPU-Photon-Mappers gezeigt. Die Ergebnisse weisen deutlich weniger Farbrauschen und räumliches Aliasing auf, konvergieren dabei durchweg zu Referenzlösungen und verursachen vernachlässigbaren Tracing-Overhead im Bereich von Millisekunden.

Alle vier vorgestellten Methoden verdanken ihre Effizienz hauptsächlich der Verwendung von höheren mathematischen Abstraktionen, für die Kenntnisse der zugrunde liegenden physikalischen Prinzipien des Lichttransports das Fundament liefern. Die Methoden wurden außerdem so entworfen, dass sie mit einfachen Datenstrukturen auskommen, was eine in hohem Maße parallele Ausführung

erlaubt und dazu führt, dass auf Vorberechnungen komplett verzichtet werden kann. Dank dieser Eigenschaften ist die vorliegende Arbeit nicht nur dazu geeignet, Lichttransport in trüben Medien interaktiv zu berechnen, sondern erlaubt auch dynamische Veränderungen der simulierten Szenerie. Das hohe Niveau des visuellen Realismus' wird dabei jederzeit beibehalten.

Contents

Chapter 1

Introduction

This thesis focuses on the development of efficient approaches for simulating the interaction of light with participating media. Several methods with different specialisations and ranges of applicability are proposed. In the first chapter we introduce the topic, describe its relations to computer graphics, explain our motivation for exploring it, and finally outline the thesis structure and its contributions.

1.1 Topic and Motivation

"One picture is worth a thousand words." Nobody knows the precise origin of this saying, likely because there is none. It is a part of common sense that visual information is the dominant channel in our environmental perception – about 90 % of communication is non-verbal [Mehrabian 1981], which is not a surprise since we process visual information four orders of magnitude faster than text [Levin 1989], in the order of 100 Mb of 'raw' data per second [Koch et al. 2006].

All this visual information, of course, is nothing else than visible light emitted or reflected by objects. On the first sight (figuratively speaking) it might seem that the visual richness around us must have many different causes, but in fact all the visible light we observe can be attributed to only fifteen different physical mechanisms [Nassau 2001]. Many of these can be ascribed to interactions with *participating media*, which is the reason that inherently motivated this thesis. Participating media (also "volumetric media" or simply "media" hereinafter) scatter, absorb, and occasionally also emit light and via these interactions give appearance to physical objects and phenomena that we observe.

Guided by physical laws, it is one of the prime interests of computer graphics to understand and simulate these interactions to render realistic images. That is because, naturally, any application that deals with visual content related to real world will likely need algorithmic methods to simulate and visualise such phenomena. That includes applications aimed both at recreation – primarily computer games and digital movies – but also serious ones, such as scientific visualisation, simulator software, or virtual prototyping and manufacturing. We do not focus on any specific one, but generally target the group of *interactive* applications which need to provide immediate feedback to their users.

Characteristics of Participating Media To understand our interest in participating media we would like to illustrate their abundance and contributions to our environment's appearance. From the human perspective, literally the most ubiquitous participating medium is the earth's atmosphere.

While computer graphics—for convenience reasons—often operates on scenes composed of solid objects separated by vacuum, it is not true that the effects of participating media are entirely ignored even in those cases. Several examples can be made:

- **Coloured transparency:** since the early Whitted ray tracing [1980] it is one of the few correctly simulated volumetric effects, commonly found in tinted glasses, gemstones, or pigmented liquids. The accessibility of this effect stems from the fact that it is described by the Beer-Lambert-Bouguer law [Born and Wolf 1999], leading to a simple non-recursive linear integral (in contrast to most other volumetric effects).
- **Diffuse reflection:** a common surface reflectance approximation used in computer graphics (usually calculated by the Lambert or Oren-Nayar [1994] models). It assumes strong scattering below the surface of an object, causing light to diffuse rapidly. This allows to treat the interaction as a local isotropic reflection, leading to the smooth appearance of some opaque objects, such as stone, clay, wood or paper.
- **Exponential fog:** a crude empirical approximation to atmospheric scattering based on a simple distance-based exponential formula, not unlike the Beer-Lambert-Bouguer law (Section 2.4.1). Used mostly in real-time applications—particularly games—to model outdoor aerial perspective and enhance perceived depth.

Most volumetric effects cannot be approximated so easily, however, and require a more explicit treatment. Although an ideal rendering method should be universal, it is often necessary to specialise on a particular type of medium or phenomenon for sake of efficiency. It is therefore useful to attempt at least a loose categorisation of these phenomena:

1. **Sparse phenomena:** usually caused by large-scale, homogeneous media, most notably atmosphere, natural waters and fog (Figure 1.1, a–c). Typical visual effects include colour shifts (due to wavelength-dependent scattering and absorption), blurring of sharp features, and (aerial) perspective effects.
2. **Translucency:** typically caused by optically thick fluid or solid media, for instance clouds, smoke, steam, organic tissues, or wax (Figure 1.1, d–g). Translucency is associated with an overall soft appearance and brightening around object boundaries (silver lining).
3. **Volumetric occlusion:** similar to shadows, usually caused by media with high density gradients (e. g., clouds) or opaque objects being suspended in a sparse medium (Figure 1.1, g–h). The most notable visual feature from this category are crepuscular rays (also sometimes dubbed “god” rays).
4. **General transparency:** similarly to coloured transparency caused by media without an apparent scattering but possible spectral dispersion, such as glass, transparent plastic, and many clear liquids. Typically leads to chromatic aberration and colourful rainbow-type caustics (Figure 1.1, i).
5. **Emission:** usually due to high temperature or excitation of specific media, e. g., plasma, fire or ionised gas. Normally modelled separately since it does not influence the propagation of light through the medium.

The methods presented in this thesis focus primarily on sparse phenomena (Chapter 3), the effects of translucency (Chapters 4 and 5) and general transparency (Chapter 6). Please refer to Section 1.3 for more details.

Figure 1.1: Examples of various types of participating media. Image credit: *(b)* Roban Kramer, flickr.com; *(c)* pexels.com; *(d)* user "rore", flickr.com; *(e)* user "Rondador", commons.wikimedia.org; *(f)* Jorge Royan, commons.wikimedia.org; *(g)* user "kalilo", rgbstock.com; *(h)* Ferdinand Kozeluh, pixabay.com; *(i)* Peter Kutz, peterkutz.com.

1.2 Problem Description

Computer science generally faces two kinds of problems: discrete and continuous. Discrete problems deal with well-defined individual entities, be it concrete or abstract – employees of a company, computers connected by a network, data files, letters of a genome, etc. They are concerned with issues like efficient organisation, sorting, searching or finding relationships between these entities. The latter category, in contrast, attempts to solve questions related to continuous quantities. These can either directly mirror the physical world, or arise as a statistical abstraction from discrete problems where treating each entity individually would be too impractical or even completely intractable. It should be obvious that physically based rendering belongs to the second category, as its object of interest is the physical behaviour of light and its interactions with objects.

In simple terms, the central problem that rendering faces can be described in two steps:

1. determine what the observer sees, and
2. determine the colour it has.

The observer is usually represented by a virtual 'camera' which captures an image of the scene of interest. Since the image is composed of pixels, we need to 'measure' the light arriving from the

objects visible through each of the pixels. Such measurement can be formulated as the *rendering equation* (RE) established in computer graphics by Kajiya [1986]. However, the RE ignores any explicit presence of participating media in the scene, as we previously mentioned.

Problem Statement Light transport in participating media is an instance of the more general *radiative transport* problem [Chandrasekhar 1960]. One of the possible mathematical formulations is the *volume rendering equation* (VRE, see Section 2.4.2), which uses the RE as its boundary condition. On an abstract level, the VRE can be written as

$$L = \int_{\mathcal{D}^3} B \cdot L + C. \tag{1.1}$$

This equation 'measures' the light energy (radiance) L in a certain position and direction. Its right side integrates across a three-dimensional domain $\mathcal{D}^3$, where one of the dimensions are positions along the measured direction and the remaining two are secondary spherical directions. The terms B and C are medium-dependent parameters, and in general are both spatially and directionally dependent. Evaluating the VRE for each pixel yields the solution to the volume rendering problem.

The main difficulty of the light transport problem arises from the recursive nature of Equation 1.1, since the quantity we are trying to calculate—radiance L—is also contained inside the right-side integral. Intuitively, to obtain the radiance arriving from a certain direction ω, one has sum up (integrate) the radiance redirected by the participating medium into ω from all other directions and positions lying along ω. This logic is then propagated further to evaluate the second-order radiance values, and so on. Expanding this definition therefore leads to an infinitely-dimensional integral which—aside from relatively trivial cases—does not have an analytic solution.

In addition to its recursivity, another difficulty in solving the VRE is its global nature, giving rise to the way how the rendering problem is often addressed: "global illumination" (abbreviated GI). This means that solving for the radiance at some location, it is potentially necessary to access the information about any arbitrary point in the scene. A photon emitted or scattered at even distant locations can reach the observer, as long as there exists a light path that can possibly carry it. Consequently, the inability to limit the evaluation of the VRE to a certain local neighbourhood of the points of interest complicates the design of algorithms to solve it and the data structures they use.

Solving the VRE Let us take a brief look at the complexity of this problem. A trivial solution to evaluate Equation 1.1 would be to use regular numerical quadrature. Evaluating m spatial samples and n^2 directional samples would imply a complexity of $O(m^i n^{2i})$, with i being the number of expansions of the right-side integral. Given that in some optically thick media a photon can scatter hundreds of times on average (which is proportional to the number of required expansions i) such naïve solution is not computationally tractable. This is often referred to as the *curse of dimensionality*. More sophisticated numerical integration methods are therefore a necessity.

Two important characteristics of a rendering algorithm are *consistency* and *bias*, which originate from statistics [Lehmann and Casella 2001]. These terms are formulated in respect to the number of samples n that a particular numerical method computes (though what a sample is can significantly differ for each method).

- A method is *biased* if its solution contains a certain systematic error for some or all values of n. An *unbiased* method does not contain any quantifiable systematic error for any n (but still can differ from the true physical solution due to, e. g., variance).
- A method is *consistent* if and only if the solution it produces converges to a certain value and its bias to zero with increasing n, and *inconsistent* otherwise.

The two most frequently utilised frameworks are stochastic *Monte Carlo* (MC) and deterministic *finite-elements* (FE) methods. MC methods perform a point sampling of Equation 1.1 using randomly distributed samples. They are typically consistent and often unbiased. MC methods are usually also more general without necessarily relying on any particular assumptions about the medium. However, the solutions often suffer from uncorrelated noise; the noise vanishes with increasing amounts of samples, but this can take a very long time in some difficult cases. Contrary to that, FE approaches attempt to discretise the integration domain and then use a deterministic iterative process to converge to a solution. They are usually numerically stable, but can suffer from discretisation artifacts and always contain an inherent bias.

Recently, the growing popularity of employing programmable *graphics processing units* (GPUs) for solving global illumination problems has increased the need for local approaches (as opposed to the more traditional global ones). This is caused by the massively parallel computational model of GPUs, where coherent access to underlying data structures is imperative at formulating efficient rendering methods. This gives advantage to FE methods (over MC methods which require random data access), especially their local formulations such as *discrete-ordinates* method (DOM) and *lattice-Boltzmann* method (LBM) [Chandrasekhar 1960; Geist et al. 2004]. Even though local FE methods tend to suffer from various forms of bias even more than the global variants, their simplicity and performance advantages in parallel environments gained them increased popularity in recent years. This also lead us to choose a local FE formulation as the basis for our method presented in Chapter 5. If fact, we employ GPUs in all our work and typically try to centre the algorithmic design around their computational model.

The affinity of GPUs towards coherent processing also leads to considerations about data access patterns. These can be categorised as *gathering* and *scattering* (which should not be confused with the physical interaction of the same name). The distinction between them is that while gathering reads a set of inputs to produce a single output value, scattering iterates over input data to contribute to possibly multiple output values. A classic example is convolution-based (e. g., Gaussian) filtering (see Section 2.8), which can be formulated in terms of both access patterns. It is to be expected that GPUs prefer the former access pattern, which has lead to a whole group of *image-space* (or *screen-space*) methods that attempt to approximate global illumination solutions using only the information about scene illumination and geometry directly visible to the camera (and stored in an intermediate image-based representation). Our method presented in Chapter 3 aims at approximating light scattering using this paradigm as well.

As is often the case in computer science, many methods opt to trade memory for computation speed, and the extent to which this is done determines the resulting performance and usually accuracy as well. Typical examples are techniques that use caching or precomputation to store the radiance function (or other components of Equation 1.1) and repeatedly reuse it during the visualisation stage. Other approaches use prior knowledge about the simulated phenomenon to improve their results, e. g., by an enhanced reconstruction. We tend to avoid precomputation as it limits the possibilities of interacting with the simulated scene, but use caching and priors in the methods described in Chapters 4 and 6, respectively.

Another means to increasing efficiency is to focus on a specific subset of media, possibly allowing to simplify some of the terms in Equation 1.1. The most frequently made assumptions are:

- **Medium homogeneity:** assumes the terms B and C to be spatially invariant. This is a reasonable assumption for media like fog and water, since they tend to have a constant density which normally determines these terms' values. We make this assumption in the method from Chapter 3.

- **Scattering isotropy:** assumes that B is directionally invariant. For most media this is not valid, but in some cases of highly optically thick materials the resulting error may be small. This assumption is the key to allow analytic approximations, most importantly the diffusion method.
- **Single scattering:** substitutes the source term L with only incident energy L_0, preventing the infinite expansion of the right-side integral. This is a crude approximation which we do not employ, although it can lead to qualitatively good results in very sparse, optically thin media.

Such assumptions allow the rendering method to focus computation where it matters more. They are usually motivated by a specific knowledge about the simulated environment. This can be knowledge about the physics of the phenomenon in question, or the perceptual impact of different features that the phenomenon may contain. A typical example is to focus sampling effort to areas where the illumination gradient is high (which can be a consequence of a high medium density gradient or occlusion).

In addition, it is useful to recognise that different application areas generally have different requirements on the methods they employ.

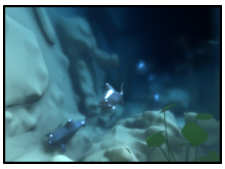
- Virtual prototyping has the highest computational budget. The employed methods must however produce unbiased solutions with physically predictive accuracy.
- Architectural and product visualisation requires still a very high accuracy, which can be traded for lower temporal coherence (since usually only still images are rendered).
- Movie production needs reasonably accurate and temporally coherent methods, but constrains the computation typically to several minutes per frame. Biased methods are acceptable, the main quality metric is perceptual fidelity instead. Another important requirement is flexibility of the employed methods to allow artistic interventions, even beyond the limits of physical correctness.
- Scientific visualisation prioritise interactivity and the ability of the utilised methods to produce expressive results, facilitating understanding and analysis of the visualised data.
- Real-time applications such as training simulators, interactive architectural visualisations and games require the rendering cost to be bounded to a certain limited budget, usually in the order of tens of milliseconds per frame. Physical accuracy is welcome but not required – the accent is on perceptual fidelity and temporal stability of the results.

An important remark is that we do not use the typical classification of algorithms into real-time, interactive and non-interactive (where real-time algorithms are understood to run at 25 Hz or more and interactive ones at least at 6 Hz [Akenine-Möller et al. 2008]). While this is justified in the human-perceptual sense, from the computational perspective such classification is very subjective as it depends on the specific use conditions and rendering hardware. Instead, we consider a method interactive if its response speed (i. e., the time necessary to produce one image) can be reasonably bounded, either theoretically or at least by experiment. For non-interactive algorithms this is not possible – a typical example are stochastic MC methods where the *absolute* convergence is individually dependent on scene configuration and usually cannot be determined a priori.

Finally, it is also good to bear in mind that radiative transport is not an isolated subject. The most directly related fields are particle transport and heat transfer. These lead to major developments of Monte Carlo and finite-elements methods from which rendering greatly benefits. Other related areas include general wave propagation (e. g., electromagnetic or acoustic), fluid flow simulation, and n-body problems. For a more detailed overview please refer to the seminal work of Veach [1997].

1.3 Contributions and Structure

In this section we briefly summarise the contributions presented in this thesis and describe its general structure. After introducing the topic in the current chapter, we describe the background on which this thesis builds in Chapter 2. This includes an exposition of the relevant theory, as well as an overview of the work relevant to this thesis or related to the questions that it tackles. The core part, Chapters 3–6, describes several novel methods that focus on different aspects of light transport in participating media. This part is based on the following publications: [Elek et al. 2012a; Elek et al. 2012b; Elek et al. 2013; Elek et al. 2014c; Elek et al. 2014d; Elek et al. 2014b; Elek et al. 2014a]. The author of this dissertation has been the primary investigator in all of these works.

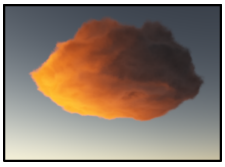


Chapter 3 describes *screen-space scattering*, a fast approximate method for computing the effects of light absorption and scattering in image-space (based on [Elek et al. 2013]). Here we primarily focus on qualitative reproduction of optically thin, homogeneous participating media (such as fog or water). The specific contributions are:

- Design of a screen-space approximation to light scattering based on a physically motivated Gaussian point spread function (PSF).
- Formulation of an image formation model based on a spatially-varying version of this function.
- Design of a GPU-optimised filtering process which applies a custom anisotropic version of the PSF in a pyramidal fashion.

While the approach is limited to homogeneous media, its main advantage is speed – a full HD image can be computed in a fixed time of a few milliseconds, while an unoptimised application of the PSF alone takes several seconds and an unbiased Monte Carlo solution computes in the order of hours. Despite that, results qualitatively comparable to the reference ones are often obtained.

The proposed filtering process builds an incremental Gaussian pyramid of the original scene image; this is a very fast process as each level of the pyramid takes a constant computational effort per pixel. The spatially variant PSF is then evaluated by fetching a level of this hierarchy with the closest correspondence to the required filtering kernel size at each location. Additionally, the proposed solution avoids the artifacts characteristic to this class of pyramidal filtering techniques, most notably energy leaking (commonly encountered in other image-space methods e. g., depth-of-field rendering). This is done by a custom anisotropic modification of the originally isotropic filtering kernel: essentially the energy that would cause the leaking is prevented from propagating to the higher levels of the pyramid.



Chapter 4 targets the rendering of clouds, which are arguably among the most difficult media to simulate due to their inherent optical properties. A real-time approach dubbed *amortised photon mapping*, specifically designed for this task, is presented (based on [Elek et al. 2012a; Elek et al. 2012b]). The specific contributions are:

- Formulating the standard photon beam tracing so that the propagated energy is coherently spread across multiple frames and then aggregated when displayed.
- Design of a custom GPU-friendly representation of the underlying photon map as a series of compact 3D lattices organised in a circular buffer. The directional information in these lattices is stored in a specialised unimodal basis that takes advantage of the illumination properties typical to clouds.

- An efficient reconstruction technique to convert between incident and out-scattered radiance.
- Adaptation of the standard bilateral upsampling by utilizing prior knowledge about the involved distributions, leading to more detailed solutions.

The method reaches real-time rendering speeds and compared to other dedicated cloud rendering techniques is not limited to specific cloud types, nor does it rely on any sort of precomputation. Thanks to this, both the underlying cloud data and the illumination can be dynamic to the extent which is normal in this setting (i. e., to the point when the rate of environmental change is comparable with the update period of the algorithm).

The key to the efficiency of this approach is using a very compact representation of the radiance function within the cloud. First, the resolution of the caching lattices is kept low, allowing for efficient caching in local GPU memory and requiring only a small amount of photon beams to be traced per frame (in addition to the tracing being amortised across multiple frames). Second, we utilise the fact that the dominant source of illumination in clouds is the sun, and use simple unimodal ellipsoid lobes to represent the directionality of the radiant energy passing through the cloud (as opposed to costlier general approaches such as spherical harmonic functions). The low resolution of the caches is then compensated for by an upsampling procedure guided by the cloud density field (which typically has a significantly higher resolution).

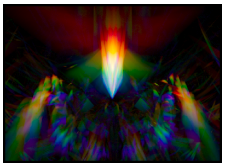
Inspired by the versatility of the unimodal directional basis described above, in Chapter 5 we present a real-time finite-elements method for solving light transport in more general, heterogeneous participating media. This chapter is based on [Elek et al. 2014c; Elek et al. 2014d] and presents the following contributions:

- Reformulating the classic discrete-ordinates method we obtain principal-ordinates propagation (POP). The core idea is to decompose the source illumination into smaller quanta which can be more efficiently propagated in a set of separate discrete domains along their approximate principal directions.
- Representation of the propagated radiance by a superposition of the specialised unimodal distributions, where each lattice contains only a single such distribution (‘lobe’) per cell.
- Deriving a set of physically-plausible rules for propagating radiant energy in such domains.
- Application of this propagation scheme to volumetric transport of light from directional, point, and environment (area) sources.
- Multiple improvements to increase the method’s generality, namely an importance propagation step to improve the illumination discretisation, adaptive illumination prefiltering for eliminating singularity artefacts in highly anisotropic media, and a two-phase extension of the propagation for improved handling of very dense media.

The resulting approach is able, in real-time speeds, to simulate light transport in a very broad range of participating media including fog, smoke, and even clouds. Most notably, very high scattering anisotropy can be handled, a feature eluding virtually all existing rendering approaches (let alone the interactive ones). And yet again, in contrast with many contemporary methods, principal-ordinates propagation does not require any precomputation, so even fully dynamic datasets can be simulated.

This is again achieved by using a suitable representation for execution on GPUs – each of the propagation lattices has a small resolution and a regular topology. Warping this topology then allows to adapt the lattices to specific light source types, so that they are aligned with the intrinsic principal

directions of the radiance energy emitted by the light sources. Also akin to the previous method, the propagation basis is a single unimodal lobe per lattice element. Aligning these lobes with the respective principal directions then allows propagating light without losing its directionality, so that anisotropic scattering can be handled.



Finally, the class of transparent materials is explored in Chapter 6. Here we describe spectral ray differentials (SRD), an efficient technique for the filtering (reconstruction) of chromatic dispersion in refractive transparent media (based on [Elek et al. 2014b; Elek et al. 2014a]). The following contributions are presented:

- Theory of spectral ray differentials, inspired by previous work on spatial ray differentiation in Monte Carlo tracing methods.
- Closed-form solutions for the spectral ray differential update after a refractive transmission between two general dielectric media.
- Family of SRD-based reconstruction techniques to reduce chromatic noise and aliasing in images generated by MC methods (demonstrated for ray and light tracing).
- Combination of SRD with the established theory of progressive kernel reduction (employed widely in, e. g., photon mapping methods), leading to consistent convergence of the resulting solutions.

Ray differentials are partial derivatives of the traced ray's position and direction along its propagation path. While the classic ray differentials track the derivatives in respect to image-space coordinates, spectral differentials analogously track the derivatives w. r. t. the spectral domain. Hence if a polychromatic ray undergoes a dispersive interaction its spectral components are spread along a certain direction which our method allows to calculate. This information is then used depending on the specific technique employed by the rendering algorithm to reconstruct illumination (e. g., splatting in light tracing or kernel density estimation in photon mapping).

The application of SRD typically leads to an order-of-magnitude less noise when reproducing chromatic dispersion effects in the context of ray-based rendering methods. They also incur a minimal overhead thanks to their analytic formulation and the fact that virtually all renderers already track some form of (spatial) ray differentials. The resulting progressive SRD framework also allows a convenient semi-interactive work with dispersive effects, since a smooth solution is produced very rapidly but as more samples are traced the result consistently converges to the true solution (because the initial bias is progressively removed).

The thesis is concluded in Chapter 7 by providing a summary and discussion of the obtained knowledge. Further directions and future outlooks are also discussed for each of the presented topics and their combinations.

Chapter 2

Background

This chapter provides a theoretical background to the contributions of the thesis presented in Chapters 3–6. Sections 2.1 and 2.2 describe basic related physics, followed by Section 2.3 which focuses on participating media. The mathematics of volumetric light transport are detailed in Section 2.4, with a thorough overview of methods targeting this problem in Section 2.5. Then, Section 2.6 focuses on another important aspect of how media interact with light – refraction and dispersion – and reviews approaches specialising on this topic. Finally, Sections 2.7 and 2.8 detail reconstruction and image formation models for both general light transport and in the presence of media.

A few notes about the used notation should be made. Throughout the remainder of the thesis, scalar quantities will be denoted by standard italicised mathematical font (e. g., radiance L or phase function f), while vector quantities will be signified by a boldface font (e. g., position $\mathbf{x}$ or direction $\mathbf{d}$). An exception will be quantities related to the spherical domain, to follow the conventions traditionally used in rendering – most notably the spherical domain Ω and its hemispherical equivalent Ω^+, plus a direction with an associated solid angle ω. Discretised versions of 2D and 3D functions (i. e., tensors) will be denoted by a sans font, for instance an image I or a density field D. For clarity, basic arithmetic operations and functions will be applied to entire vectors and tensors, where (assuming the operands have matching dimensions) the result will again be a vector / tensor where the respective operation is applied for each element separately (for instance per-pixel or per-voxel).

2.1 Optics

Optics is the part of physics which provides the most theoretical knowledge for physically based rendering. It is most commonly divided into four successive levels: quantum optics, electromagnetic optics, wave optics and geometric (ray) optics [Born and Wolf 1999]. Each of these levels is an abstraction upon the previous one, meaning that they provide increasingly higher-level descriptions of optical phenomena but the actual set of described phenomena generally decreases in size.

From the perspective of rendering, by far the most useful abstraction is geometric optics, simply because most phenomena are observed by humans on a sufficiently macroscopic scale. Some of the assumptions about the behaviour of light made by geometric optics are [Dutré et al. 2006]:

- Light travels along straight trajectories and bends only discontinuously, i. e., each light path is a

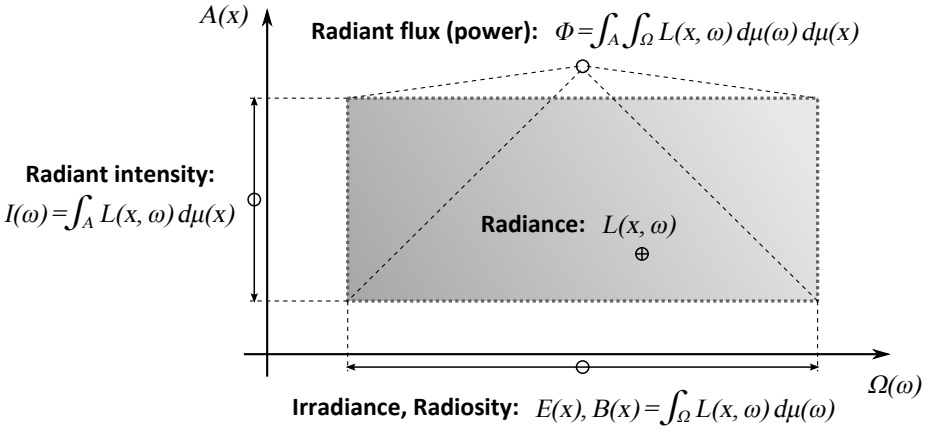


Figure 2.1: Diagram depicting the main radiometric quantities – radiance, irradiance, radiosity, intensity and flux – and their relations.

piecewise-linear construct.

- Light propagates instantly, i. e., the equilibrium energy in a scene can be measured by estimating the number of light paths passing through an area of interest.
- Light is not influenced by external force fields, such as gravitational or electromagnetic.

However, there are specific problems where some of these assumptions must be relaxed, for instance the rendering of phosphorescence, media with continuously varying refractive index, or relativistic effects. Furthermore, some phenomena cannot be modelled by any adaptation of geometric optics and require wave optics to be simulated. The most important examples of these are diffraction and interference.

Light transport in participating media (most notably scattering and dispersive refraction, see Section 2.3) is an intermediate category: it cannot be explained by geometric optics, but it can be simulated by it for most practical purposes. More specifically, *phase function* (Section 2.3.2) in the wave-optical sense is the far-field amplitude of a polarised scattered light wave [van de Hulst 1981]. However, computer graphics regards phase functions as spherical probabilistic density distributions which can be discretely sampled to redirect light paths.

2.2 Radiometry

Radiometry is the study of measuring electromagnetic radiation, including the visible light relevant for us. Radiometry is an objective discipline concerned with purely physical quantities, as opposed to photometry (which accounts for the properties of human visual system). These quantities are primarily related to spatial and directional distributions of energy with given spectral properties.

Arguably the most important radiometric quantity is *radiance* L [$\mathrm{W \cdot m^{-2} \cdot sr^{-1}}$], sometimes (incorrectly) referred to as intensity. Radiance is the quantity measured by sensors, including our eyes. That is likely due to the fact that it is invariant along light paths, meaning that the radiance leaving point $\mathbf{x}$ towards point $\mathbf{y}$ will be equal to the radiance reaching $\mathbf{y}$ from the direction of $\mathbf{x}$ (assuming their perfect mutual visibility).

Other relevant radiometric quantities can be obtained by integrating radiance along its domains

(please refer to the diagram in Figure 2.1):

- *Radiant intensity* I [W·sr^{-1}] is obtained by integrating radiance across an area A (which can be surface area, projected surface area, or any closed 2-dimensional region in free space).
- Integrating in the directional domain Ω yields *irradiance* E [W·m^{-2}] and *radiosity* B [W·m^{-2}], which are particularly useful in surface radiosity methods (hence the name).
- The combined integration in both these domains results in *radiant flux* Φ [W]. Flux is the basic quantity used to specify radiant power of light sources.

In addition, all these quantities are also defined in terms of their wavelength λ. A radiometric entity can have a single wavelength associated with it (and then is qualified as *monochromatic*), or an entire (discrete or continuous) spectrum of wavelengths.

2.3 Participating Media

The notion of participating media in computer graphics has already been introduced in Section 1.1. Let us now look at some important properties of participating media that influence the way how they interact with light. For a detailed exposition of the knowledge summarised here please refer to the excellent book by van de Hulst [1981].

2.3.1 Types of Interaction

We understand participating media as agglomerations of small particles, which can be liquid droplets, solid corpuscles, and even individual atoms and molecules. It is a common assumption in all computer graphics (and most physics as well) that the interaction of light can be described for each particle individually and then extended to the medium as a whole by superposition. Let us temporarily assume that the medium is composed of identical particles, where each is characterised by its absorption and scattering cross-sections C_a [m^2] and C_s [m^2] and a phase function f (see Section 2.3.2), and the density of these particles at a location $\mathbf{x}$ is $\rho(\mathbf{x})$ [m^{-3}]. The values of C_a and C_s are roughly proportional to the particle size, but also relate to its electromagnetic properties.

On the level of individual medium particles and photons, three things can happen: a photon that hits a particle can be absorbed or scattered, or a new photon can be emitted. Statistically, these events can occur at any point within the medium where $\rho(\mathbf{x}) > 0$. Figure 2.2 illustrates these interactions. Max [1995] reviews these interactions from the rendering perspective and derives several image formation models resulting from them.

Absorption If a photon passing through a medium is absorbed, it is lost and its energy is transformed into another form, such as kinetic energy of the medium particle. Macroscopically, absorption decreases the intensity of light passing through the medium; for example, water absorbs light with longer wavelengths slightly more, causing any light passing through to become bluer after some travelled distance.

The absorption intensity of a medium is determined by its *absorption coefficient* σ_a [m^{-1}]:

$$\sigma_a(\mathbf{x})\ [\mathrm{m}^{-1}] = C_a\ [\mathrm{m}^2] \cdot \rho(\mathbf{x})\ [\mathrm{m}^{-3}]. \tag{2.1}$$

Scattering If a photon is scattered its energy is diverted into another direction (which depends on

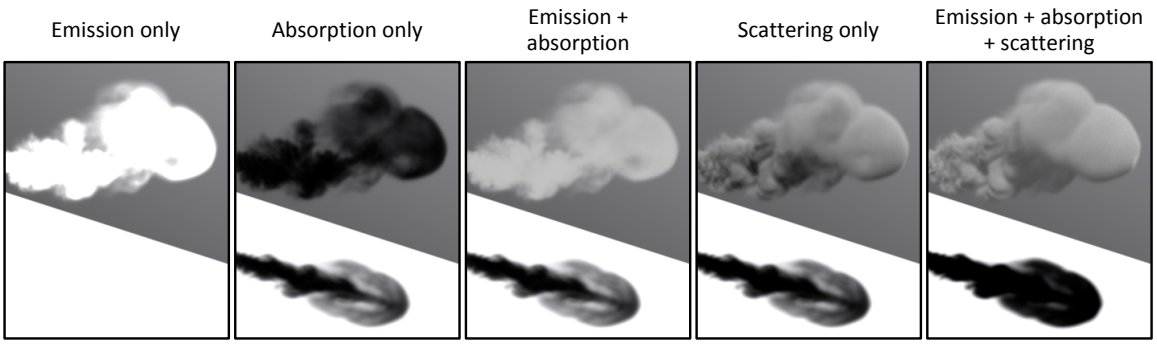


Figure 2.2: Basic interaction types in participating media, demonstrated on a smoke plume (rendered for a single wavelength for clarity).

the medium phase function, Section 2.3.2). For instance, our atmosphere scatters short-wavelength light stronger, diverting more blue photons into the direction of ground observers, making the sky appear blue as well.

Similarly to absorption, scattering is guided by the medium's *scattering coefficient* σ_s [m^{-1}]. Its definition and interpretation are equivalent to the one of σ_a as well:

$$\sigma_s(\mathbf{x})\ [\mathrm{m}^{-1}] = C_s\ [\mathrm{m}^2] \cdot \rho(\mathbf{x})\ [\mathrm{m}^{-3}]. \tag{2.2}$$

Emission Finally, a medium can emit photons from within its volume, most often as a consequence of black-body radiation. The emitted radiant energy L_e is sometimes referred to as *source function*.

Further in this text we will not describe volumetric emission. This is mainly for two reasons: first, we do not focus on modelling emissive media in our work, and second, the inclusion of emission into the simulation is simple, as it is an additive spatially-varying diffuse term (see Section 2.4.2) and by itself usually does not influence how light propagates in the medium.

Extinction This is not a distinct type of interaction, but rather the combined effect of absorption and scattering. Thanks to the linearity of these two interactions one can simply define extinction as an event when a photon is either absorbed or scattered. Extinction therefore decreases the intensity *along* a light beam passing through the medium.

Consequently the *extinction coefficient* σ_t [m^{-1}] can be defined as $\sigma_t = \sigma_a + \sigma_s$. Extinction coefficient inversely defines the mean free path of a photon in a medium; for instance, in a homogeneous environment with $\sigma_t = 0.5\,\mathrm{m}^{-1}$ the average distance a photon travels until it is absorbed or scattered is 2 m.

Furthermore, based on σ_t we define the scattering efficiency or the *scattering albedo* α as

$$\alpha = \frac{\sigma_s}{\sigma_a + \sigma_s} = \frac{\sigma_s}{\sigma_t}. \tag{2.3}$$

The notion of scattering albedo equivalent to the one of material albedo: if a photon hits a medium particle with albedo α, it gets scattered with the probability of α and absorbed with the complementary probability $1 - \alpha$.

In abstract terms, the higher the albedo of a participating medium is, the more difficult is to simulate it. This is because, on average, photons will propagate through a high-α medium for longer, until they either leave the medium entirely or are terminated with the probability $1 - \alpha$. This is even more

prominent in media with high *optical thickness* τ, which is a quantity defined along a linear optical path l as

$$\tau(l) = \int_l \sigma_t(t)\, \mathrm{d}t. \tag{2.4}$$

The function τ accounts for the changes of extinction coefficient σ_t in heterogeneous media; for homogeneous media it is analytically defined as

$$\tau(l) = \sigma_t \cdot \|l\|. \tag{2.5}$$

Most notable examples of difficult media in the above sense are clouds and milk, where typically the albedo is higher than 0.999 and the medium sizes hundreds to thousands times larger than the mean free paths.

Some of the qualitative properties these interactions produce can be seen in Figure 2.2. A (hypothetical) purely emissive medium does not cast shadow, since there is no mechanism that would attenuate light. On the other hand absorption just attenuates light, so the cloud appears black (this is a behaviour similar to, e. g., coal dust). However, none of these two gives a particularly good information about spatial relations and mass distribution in the cloud – only scattering does, as it is the only interaction that influences light directionality. Also notice that the combined effect of absorption and scattering creates a darker shadow than just absorption alone. This is the consequence of the extinction coefficient σ_t containing both σ_a and σ_s; the only difference is that while absorbed light is 'lost', scattered light just changes its direction and ultimately a portion of it also reaches the observer via another light path.

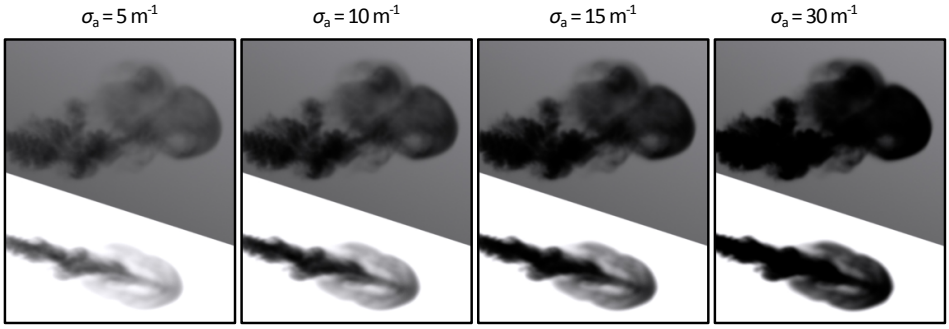


Figure 2.3: The effects of absorption strength increase on medium appearance.

As we will see in Section 2.4.1, the relation between the coefficients and the increase or decrease of light intensity is not linear, but exponential instead. The effect of increasing value of σ_a is illustrated in Figure 2.3; seemingly, the cloud with $\sigma_a = 30\,\mathrm{m}^{-1}$ is less than six times 'darker' than the one with $\sigma_a = 5\,\mathrm{m}^{-1}$.

A consequence of these relations is that the absorption and scattering coefficients scale linearly with local medium density. We can therefore represent participating media by storing their peak σ_a and σ_s and a normalised density field $\rho(\mathbf{x})$. Not only can this strategy decrease memory requirements, but in some cases also allows us to change the medium absorption and scattering properties independently by changing only the values of σ_a and σ_s, instead of altering the entire medium dataset. Likewise, for media composed of multiple substances or particle types, the same representation may be used as long as the relative densities of each particle type remain the same (which often holds for gaseous and liquid media).

2.3.2 Phase Function

So far we have only described how frequently interaction events in a participating medium occur, yet we do not know what happens after such an event takes place.

As already mentioned above, light scattering is a wave phenomenon. If an incident light wave meets a scattering particle the latter will start to oscillate, inducing a secondary 'scattered' wave. The amplitude of this newly created wave is given by the spherical *magnitude function* $f'(\omega_i, \omega_o)$, for a pair of incident and outgoing directions ω_i and ω_o. Since we would like to work in the realm of discrete light rays and particles within geometrical optics, we will instead utilise the *phase function* $f(\omega_i, \omega_o)$, which is derived from f' (please refer to van de Hulst [1981] for details). Both f and f' are also wavelength-dependent.

A commonly made assumption about participating media (which we also employ throughout this work) is that the medium particles have random, uncorrelated orientations. This has two consequences. First, the phase function becomes independent on the incident direction ω_i, and can therefore be expressed in terms of the relative orientation between ω_i and ω_o as $f(\theta, \phi)$, where $\theta \in [0, \pi]$ is the polar angle (also referred to as *scattering angle*) and $\phi \in [0, 2\pi]$ the azimuthal angle. Second, f becomes rotationally invariant around the scattering axis ω_i, which allows us to further omit the azimuth ϕ in the notation. Some limitations of this assumption are discussed in Section 2.3.4.

Each phase function f has to fulfil several key requirements – it must be non-negative (Equation 2.6), energy-conserving (Equation 2.7) and reciprocal (Equation 2.8):

$$\forall \theta, \phi : \; f(\theta, \phi) \geq 0, \tag{2.6}$$

$$\int_\Omega f(\omega') \, \mathrm{d}\mu(\omega') = \int_0^{2\pi} \int_0^{\pi} f(\theta', \phi') \sin(\theta') \, \mathrm{d}\theta' \, \mathrm{d}\phi' = 1, \tag{2.7}$$

$$f(\omega_i, \omega_o) = f(-\omega_o, -\omega_i). \tag{2.8}$$

Equations 2.6 and 2.7 also make every phase function a proper spherical *probability density function* (PDF).

In a rendering system that uses geometrical optics for light simulation, the above properties enable f to be used in two ways. The first, direct use allows us to take ω_i and ω_o at a scattering point and use the corresponding value of f to scale the intensity of the light beam or particle that scatters into ω_o. The second, more interesting use enables us to generate a new outgoing scattering direction ω_o from the incident direction ω_i with the probability density defined by f. Importance sampling or rejection sampling can be used for this purpose. Such approach saves us from scaling down the newly generated beam or light particle intensity – the method itself ensures that the light energy carried by the generated samples will be directionally distributed according to f. And since f is a valid PDF, both of these techniques conserve energy.

However, the main issue with f is that it is generally hard to evaluate. The values of both the magnitude function and the phase function are the result of a complex oscillating behaviour of the scattering particle induced by the incident light wave. In a great majority of configurations (meaning shape of the scatterer and the light wave state) f does not have a closed analytical form and hence involved numerical methods have to be used to evaluate it. To avoid this we can make use of the two existing theoretical approximations, each building on a set of assumptions that hold quite well for certain groups of media.

Rayleigh theory Proposed by John William Strutt, The Lord Rayleigh, in a series of papers starting

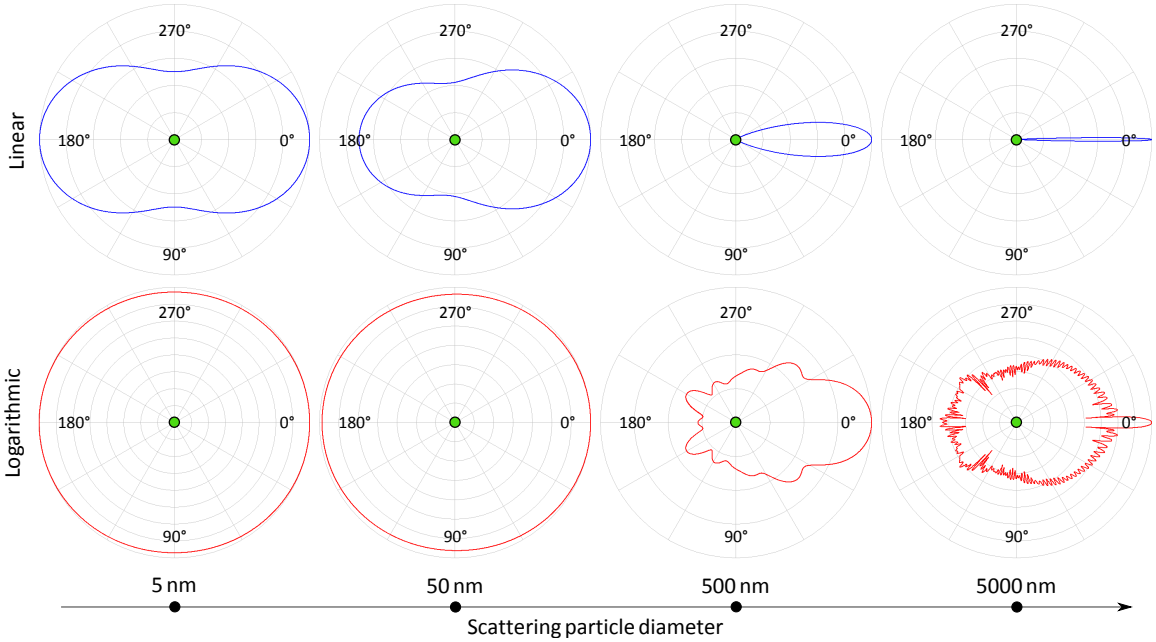


Figure 2.4: Plots of Mie phase functions for several orders of scattering particle sizes (computed for $\lambda = 500\,\text{nm}$). For small particles in the order of single nm, the phase function converges to the Rayleigh approximation *(left)*. Generated with MiePlot (Philip Laven, http://www.philiplaven.com/mieplot.htm).

in 1871 [Strutt 1871], Rayleigh theory describes light scattering by particles which are very small in comparison to the incident light wavelength. Formally, the scattering particle diameter d has to conform to

$$d \ll \frac{\lambda}{2\pi} \tag{2.9}$$

which holds quite well for particles ranging from individual atoms to small organic molecules. If Equation 2.9 holds, the phase function for non-polarised light is

$$f_{\text{Rayleigh}} = \frac{3}{4}(1+\cos^2\theta). \tag{2.10}$$

The plot of f_{Rayleigh} is shown in Figure 2.4, left.

An important property of Rayleigh scattering is that its scattering coefficient σ_s is inversely proportional to λ^4, which means that shorter wavelengths are scattered with much higher intensity than longer ones. For this reason the Rayleigh scattering is chromatic and responsible for many well-known scattering phenomena, the most prominent being the sky colour. Also, as opposed to Mie scattering (below), Rayleigh scattering yields a strongly polarised light, mainly for scattering angles around 90°.

Mie theory The solution of Maxwell's equations for electromagnetic waves interacting with perfectly spherical dielectric particles of an arbitrary size has been described by Gustav Mie [1908]. There is no single Mie phase function; rather, for every particle diameter there exists a unique corresponding phase function. Figure 2.4 shows several examples. Despite being band-limited, Mie functions have complicated a behaviour revealed by the logarithmic plots (bottom). However, for particle sizes comparable to the light wavelength or bigger the dominant portion of scattered light

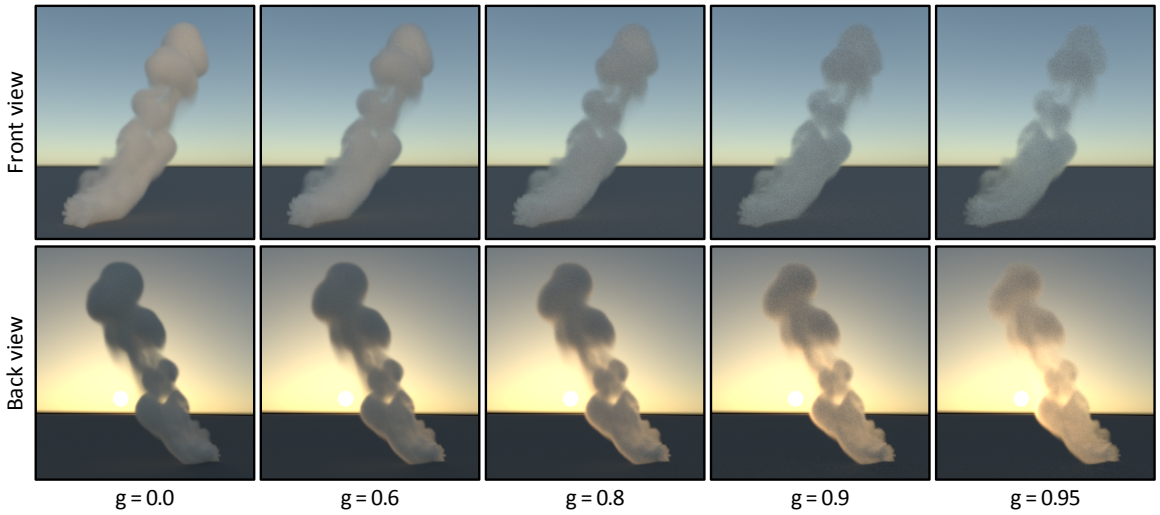


Figure 2.5: The appearance of otherwise identical media with different phase function anisotropy values, for two opposing viewpoints. The particular phase function model used here was the Henyey-Greenstein function (see Section 2.3.3). The top row was exposed 1 EV higher to compensate for the lower brightness. Rendered with Mitsuba [Jakob 2010].

will be redirected forwards. This kind of behaviour is universal for Mie scattering: the larger the scattering particle is, the more light is scattered into forward directions.

Although Mie theory is derived for perfectly spherical dielectric particles, it is widely useful. For one instance, fluid phenomena are constituted by tiny droplets with mostly spherical shapes. However, to work with Mie scattering we need to evaluate its phase function somehow. Since no analytic expression exists (the solution is an infinite series expansion), there are two options to do that: either by using precomputed tabulated values of the rigorous phase functions, or by utilising an empirical approximation. As explained in Section 2.3.3 we opt for the latter variant, due to certain mathematical properties that some of these approximations offer, but also because it is technically simpler and more portable.

Let us also remark that there is no sharp transition between Rayleigh and Mie scattering. Instead, Rayleigh scattering is a limit case of Mie scattering for very small particles. When decreasing the size of an examined spherical particle, its phase function eventually converges to f_{Rayleigh}. One of the implications is that Mie scattering is in fact also wavelength dependent, but the differences of the Mie phase function values for different wavelengths are quickly disappearing as the scatterer size approaches the visible light wavelengths. Because of this, Mie scattering is considered to be an achromatic phenomenon for most purposes.

To understand the way how different phase functions influence the behaviour of light on the macroscopic scale we need to characterise them in some way. As no single expression encompassing the entire space of phase functions exists, we can attempt to characterise them indirectly. The simplest such characteristic is the first angular moment (also *anisotropy coefficient* or just *anisotropy*), commonly referred to as g. This quantity is a weighted spherical cosine average of the given phase function f:

$$g = \int_0^{2\pi} \int_0^{\pi} f(\theta', \phi') \cos(\theta') \sin(\theta') \, \mathrm{d}\theta' \, \mathrm{d}\phi'. \tag{2.11}$$

The interpretation of g, which has values $\in [-1, 1]$, is as follows. For positive g the phase function

scatters predominantly into forward directions, and vice versa for negative values. Special cases occur at the extremes: for $g = -1$ all light scatters back into $-\omega_i$ and for $g = 1$ into ω_i (i. e., the directionality remains unchanged). Due to the symmetric nature of this relation we will focus only on the positive g values from now on. This is in part because virtually all natural media exhibit forward scattering [Narasimhan et al. 2006].

Figure 2.5 demonstrates the impact of different scattering anisotropy values on the appearance of a generic gaseous medium. The medium parameters and illumination are otherwise identical in all cases. It is immediately apparent that more anisotropic scattering causes the medium to appear more translucent and 'thinner'. This is a consequence of the fact that light penetrates into media with stronger forward scattering faster and generally also exits them earlier. Perfectly transparent media correspond to the limit case of $g = 1$, where light travels through the medium undisturbed by scattering. Also notice the non-linearity of the appearance space defined by g. For small values up to roughly 0.6 the visual impact of changing g is typically very small, but for highly forward scattering close to $g = 1$ even small changes of anisotropy can have dramatic effects on the resulting medium appearance.

2.3.3 Henyey-Greenstein Function

One of the most widely established approximations to Mie phase functions is the Henyey-Greenstein function (abbreviated HGF hereinafter), which was originally introduced in astrophysics [Henyey and Greenstein 1941]. This is the phase function model which we use throughout all our work.

HGF is a unimodal spherical distribution parametrised by the anisotropy coefficient $g \in [-1, 1]$ (see Equation 2.11):

$$f_{\mathrm{HG}}(\theta, \phi, g) = \frac{1}{4\pi} \cdot \frac{1 - g^2}{(1 + g^2 - 2g\cos\theta)^{3/2}}. \tag{2.12}$$

Notice that the function is independent on the scattering azimuth ϕ due to its symmetry around the scattering axis. For further convenience, f_{HG} can be written as

$$f_{\mathrm{HG}}(\vartheta, g) = \frac{1}{4\pi} \cdot \frac{1 - g^2}{(1 + g^2 - 2g\vartheta)^{3/2}} \tag{2.13}$$

by defining $\vartheta = \cos\theta$. It is not difficult to verify that f_{HG} indeed fulfils the requirements given by Equations 2.6, 2.7 and 2.8.

Basic properties HGF has proven useful for approximating scattering phase functions, since it represents the preferred directionality of scattered distributions by only the anisotropy coefficient g and has several useful mathematical properties. HGF behaves intuitively in respect to g, in agreement with the general intuition given in Section 2.3.2:

- positive values of g correspond to dominantly forward scattering,
- symmetrically, negative g's represent backward scattering,
- $g = 0$ defines a constant (isotropic) phase function, and
- $g = 1$ and $g = -1$ correspond to perfectly forward / backward scattering, i. e., Dirac function in the corresponding direction.

Please refer to Figure 2.6 for 2D and Figure 2.7 for 3D plots of FHG. Several sample images rendered with HGF are shown in Figure 2.5 for different values of g. An interesting property of HGF is also

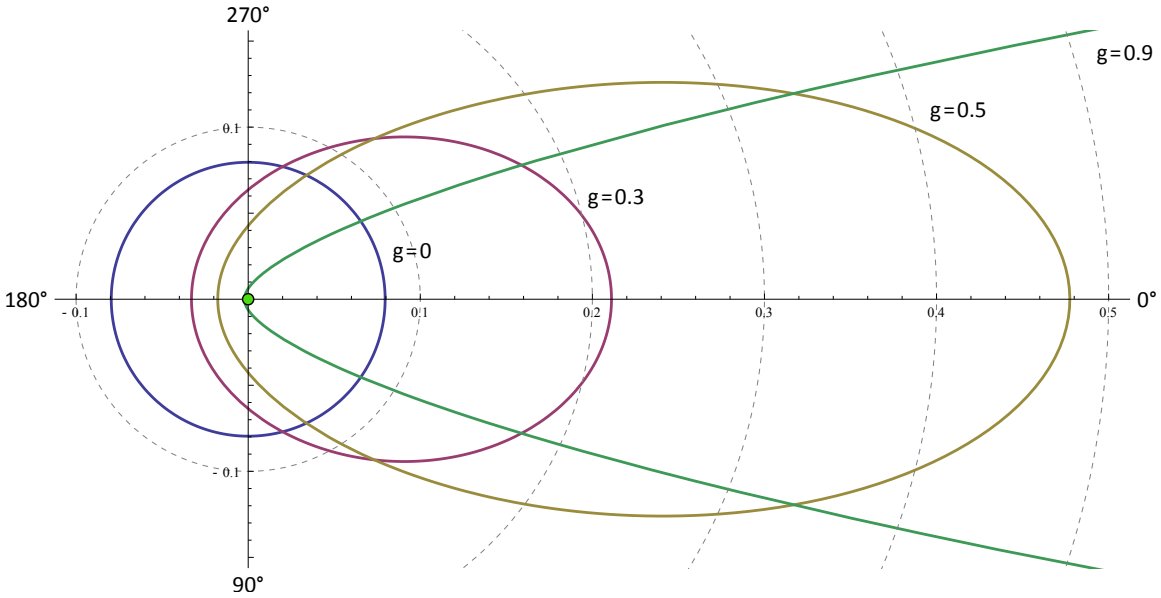


Figure 2.6: Linear polar plots of f_{HG} for $g \in \{0, 0.3, 0.6, 0.9\}$ w. r. t. scattering angle θ. At $g = 0$ the function is isotropic, and for negative g values the function is just mirrored around the y-axis. Note that although HGF is properly normalised on a sphere (according to Equation 2.7) the plots show only the polar component and as such appear to have larger areas for increasing g's.

that it is an identity function in respect to g:

$$g = 2\pi \cdot \int_0^{\pi} f_{\mathrm{HG}}(\cos(\theta'), g) \cos(\theta') \sin(\theta') \, \mathrm{d}\theta' \tag{2.14}$$

$$= 2\pi \cdot \int_{-1}^{1} \vartheta' \cdot f_{\mathrm{HG}}(\vartheta', g) \, \mathrm{d}\vartheta', \tag{2.15}$$

where the second part (Equation 2.15) integrates along the cosine of the scattering angle θ instead.

Sampling In Monte Carlo methods it is useful to generate samples distributed according to the given phase function. This is easy to do for HGF thanks to its analytical invertibility – a spherical direction (θ, ϕ) with the distribution $f_{\mathrm{HG}}(g)$ can be generated [Pharr and Humphreys 2010] as

$$\cos\theta = \frac{1}{|2g|} \cdot \left(1 + g^2 - \left(\frac{1 - g^2}{1 - g + 2g\xi_1}\right)^2\right) \tag{2.16}$$

$$\phi = 2\pi\xi_2 \tag{2.17}$$

where $\xi_1, \xi_2 \in [0, 1]$ are uniformly distributed random variables. This technique is used in Chapter 4 to trace photons through the medium.

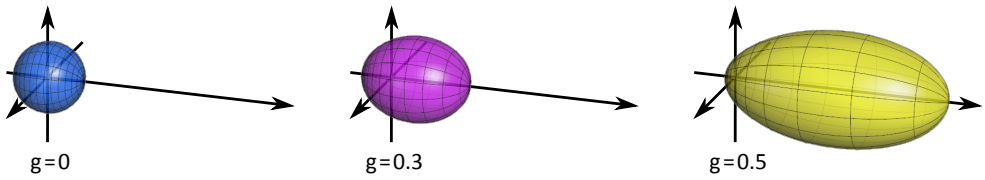


Figure 2.7: 3D plots of f_{HG} for $g \in \{0, 0.3, 0.6\}$.

Integration In our work we utilise HGF no only to model the scattering phase functions, but for representing directional radiance distributions (specifically in Chapter 5). This of course limits us to unimodal distributions of radiance, but allows efficiently calculating the radiance represented by HGF for a certain range of directions thanks to its analytical integrability. Calculating the spherical integral of Equation 2.13 means evaluating its cumulative distribution function

$$F_{\mathrm{HG}}(\vartheta,\phi,g) = \int_0^{2\pi}\int_{-1}^{\vartheta} f_{\mathrm{HG}}(\vartheta',\phi',g)\,\mathrm{d}\vartheta'\,\mathrm{d}\phi', \tag{2.18}$$

for which a closed-form solution exists:

$$F_{\mathrm{HG}}(\vartheta,\phi,g) = \phi\cdot\frac{1-g^2}{4\pi g}\cdot\left(\frac{1}{(1+g^2-2g\vartheta)^{1/2}}-\frac{1}{1+g}\right). \tag{2.19}$$

Convolution Another useful feature of HGF, which we utilise in Chapter 5, is its *self-convolution* property [Max et al. 2004]. This means that the distribution resulting from i successive (spherical) convolutions of multiple f_{HG} is again a HGF parametrised by the product of the convolved distributions' parameters:

$$\underbrace{f_{\mathrm{HG}}(g_1)*\cdots*f_{\mathrm{HG}}(g_i)}_{i\text{ terms}} = f_{\mathrm{HG}}(\prod_{j=1}^{i} g_j). \tag{2.20}$$

This requires $\forall j\in[1..i]:\ g_j\in[0,1]$. If this is fulfilled we can see that the resulting function will have a lower anisotropy g that any of the convolved distributions. This agrees with the intuition that applying a convolution to a distribution should result in distributions with decreasing anisotropy.

2.3.4 Other Phase Function Models

HGF has proven to be an efficient model for anisotropic phase functions in computer graphics and this is the reason we have chosen to work with it. In spite of that it is certainly not always the best choice, especially when numerical precision is the primary simulation objective. In such cases it might be preferred to precompute a tabulated version of the respective rigorous phase function [Toublanc 1996; Bai et al. 2011], including the rendering of certain specific appearance features of clouds [Bouthors et al. 2006; Bouthors et al. 2008]. It has also been noted [Binzoni et al. 2006; Zolek et al. 2008] that certain issues might arise when HGF is sampled. Additionally, HGF also does not represent well the mentioned transition between Rayleigh and Mie scattering – in such case a hybrid function is more suitable [Cornette and Shanks 1992; Liu and Weng 2006].

In principle it is possible to use any spherical function that conforms to the requirements discussed in Section 2.3.2. Many such functions are available, for instance the efficient expression by Blasi et al. [1993] or the more general von Mises-Fisher distribution [Fisher 1953] which is a spherical version of isotropic 2D Gaussian distribution. The latter has been used in recent work [Gkioulekas et al. 2013; Zhao et al. 2014] to create a hybrid bimodal distribution to improve the appearance reproduction of (mainly) solid materials.

Finally, some media are composed of particles or structures with highly elongated or correlated shapes. If that is the case it is necessary to use the more general model of phase function $f(\omega_{\mathrm{i}},\omega_{\mathrm{o}})$ which allows a bi-directional scattering anisotropy. Examples of such media include hair [Zinke et al. 2008; Ren et al. 2010], fur [Kajiya and Kay 1989], textiles [Zhao et al. 2011] or any general material composed of small fibres or filaments [Zinke and Weber 2007].

2.4 Light Transport

Section 2.3 describes how participating media interact with light locally. Here we introduce the principles that describe this behaviour on a global level.

2.4.1 Volumetric Visibility

Section 2.3.1 explains how to quantify the mean free path of a photon in a participating medium. However, this does not tell us how much will a beam of light be attenuated if it travels a certain distance in a medium. This attenuation is described by the **Beer-Lambert-Bouguer law** (Equation 2.21) for a thin pencil of light carrying radiance L that enters a participating medium with extinction coefficient of σ_t. The (reduced) radiance L' after travelling along an optical path l will be $L' = L \cdot T(l)$, where T is *transmittance* function defined as

$$T(l) = e^{-\tau(l)}, \tag{2.21}$$

and τ is the optical thickness (Equation 2.4). The optical path l is an abstraction: in most cases it will simply correspond to a linear segment between two points, but in general can be curved for light travelling in media with continuously varying refractive index (these we however do not consider).

The reason why the Beer-Lambert-Bouguer law is exponential can intuitively be understood as follows. The process of light attenuation by absorption and scattering is a constant continuous decay process in the limit case. As such it is naturally described by the exponential function. For instance, if the attenuation rate of light in a homogeneous medium is 100 % (i. e., $\sigma_t = 1\,\mathrm{m}^{-1}$), then the light travelling a path of length 1 m in this medium will be attenuated by the factor of $e^{-\sigma_t} = 1/e$. This is however exactly what we expect from a continuous decay process along a unit path, since

$$\sigma_t = 1 \quad \Leftrightarrow \quad \lim_{n\to\infty}(1 - \frac{\sigma_t}{n})^n = \frac{1}{e}. \tag{2.22}$$

Free path sampling Algorithms based on tracing rays (or particles) generally need to perform two basic operations in each tracing step: finding the point where the light interacts with the medium, and (if the interaction was scattering) finding a new scattering direction. The second task is usually solved by importance-sampling the phase function (see Section 2.3.3). As for the first one, one needs to follow the principle of exponential extinction described by the Beer-Lambert-Bouguer law.

The simplest case are homogeneous media. Here the extinction is governed by the simple analytic function given by the Equation Equation 2.5. Finding the distance to the next extinction event d_{event} (w. r. t. to a given starting point $\mathbf{x}_0$) means evaluating the following equation:

$$d_{\text{event}} = -\frac{\log \xi}{\sigma_t}, \tag{2.23}$$

where ξ is a uniform unit random variable. This procedure is stochastic as scattering and absorption are defined in probabilistic terms.

However, we typically deal with heterogeneous media. Here it is necessary to search along the examined light path to find the next interaction event, since the medium composition along the path is not known in advance. Therefore to find the distance d_{event} we must evaluate the following implicit equation derived from Equation 2.21:

$$\int_0^{d_{\text{event}}} \sigma_t(\mathbf{x}_0 + t \cdot \omega)\, \mathrm{d}t = -\log(1 - \xi). \tag{2.24}$$

This equation has to be evaluated numerically since the distribution of σ_t cannot usually be expressed analytically. The most straightforward method is to use *ray marching*: the algorithm steps through the medium (either using constant steps, or random-sized steps to avoid aliasing) and accumulates the value of the τ function (which is basically the left side of Equation 2.24). This is done until the accumulated value of τ exceeds the value on the right side, or until the stepping does not leave the medium boundary.

Input:
$\mathbf{x}_0$, ω: origin and direction of the starting ray
σ_t: majorant extinction coefficient
$[d_{\min}, d_{\max}]$: interval of the ray to account for
$\text{rand}() \in [0,1]$: uniform random number generator
Result:
d_{event}: distance along the ray where the interaction occurs
begin
  $d_{\text{event}} = d_{\min} - \log(1-\text{rand}())/\sigma_t$
  while $d_{\text{event}} \leq d_{\max} \wedge \sigma_t(\mathbf{x}_0 + d_{\text{event}} \cdot \omega)/\sigma_t < \text{rand}()$ **do**
    $d_{\text{event}} = d_{\text{event}} - \log(1-\text{rand}())/\sigma_t$
  end
  return d_{event}
end

Algorithm 1: Woodcock tracking.

Ray marching is a conceptually simple, but unfortunately biased algorithm – the bias arises from the choice of the step size. A better technique to use is *Woodcock tracking* (Algorithm 1, also known as delta tracking) which was originally proposed by Woodcock et al. [1965] for neutron tracing in nuclear reactions, and introduced into computer graphics by Raab et al. [2006].

Woodcock tracking works in the following way. First, the so-called *majorant* extinction coefficient σ_T is computed: $\sigma_T = \max(\sigma_t(\mathbf{x}))$ for all $\mathbf{x}$ lying inside the medium. The algorithm then performs a randomised ray marching through the medium, but treats it as homogeneous with the extinction coefficient σ_T. In each tracking step the algorithm then looks up the actual σ_t at the current position $\mathbf{x} = \mathbf{x}_0 + d_{\text{event}} \cdot \omega$. The event is accepted as a real extinction with the probability $\sigma_t(\mathbf{x})/\sigma_T$; otherwise the event is declared as virtual, discarded and the tracking continues. After generating a real event the decision if it is absorption or scattering is decided based on the medium albedo α (Equation 2.3).

This method was proven to generate correct unbiased mean free path values [Coleman 1968]. However, for media with high density variation, Woodcock tracking can become very inefficient due to the acceptance probability σ_t/σ_T being low in a majority of the volume. Accelerations methods based for example on regular grids [Szirmay-Kalos et al. 2011] and kD-trees [Yue et al. 2010] have been proposed to alleviate this issue.

Transmittance evaluation Another frequent operation in light transport computation is evaluating the visibility between two points. In the absence of participating media this is a binary function. In media, however, visibility is fractional and corresponds to the transmittance function (Equation 2.21). Similarly to the free path sampling problem, this can be integrated by numerical quadrature (i. e., ray marching) in a biased manner, or using Woodcock tracking to produce an unbiased but potentially inefficient solution. Recent work by Novák et al. [2014] has dramatically improved the situation by adding a new unbiased but efficient transmittance estimator dubbed *residual ratio tracking*.

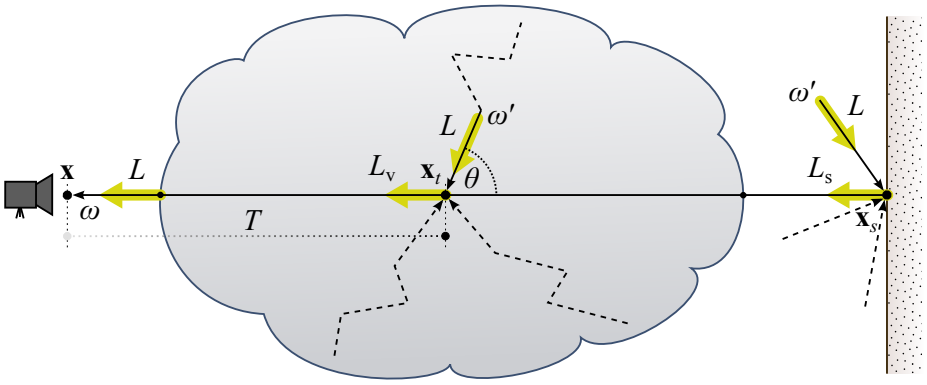


Figure 2.8: Scheme depicting the volume rendering equation (VRE, Equation 2.27). Both the surface (Equation 2.28) and volume (Equation 2.26) contributions are shown for specific respective sample points $\mathbf{x}_s$ and $\mathbf{x}_t$ and directions ω'.

2.4.2 Volume Rendering Equation

At this point we have all the necessary components that define the behaviour of light in participating media: how light interacts with the medium particles (Section 2.3.1), how frequently these interactions occur (Section 2.4.1) and how the light propagation changes after each interaction (Section 2.3.2). With that we can now formalise the light transport problem which has already been briefly exposed in Chapter 1.

The first step is to unify all the interaction types in a single relationship. This is called the *radiative transport equation* (also radiative transfer equation or just RTE), which has seen one of its first uses in heat transport [Chandrasekhar 1960]. The RTE defines the differential radiance change due to the combined effects of the medium at a given location $\mathbf{x}$ along a direction ω:

$$\begin{aligned}(\omega \cdot \nabla)L(\mathbf{x},\omega) &= \underbrace{L_\mathrm{e}(\mathbf{x},\omega)}_{\text{emission}} + \underbrace{\sigma_\mathrm{s}(\mathbf{x})L_\mathrm{i}(\mathbf{x},\omega)}_{\text{in-scattering}} - \underbrace{\sigma_\mathrm{a}(\mathbf{x})L(\mathbf{x},\omega)}_{\text{absorption}} - \underbrace{\sigma_\mathrm{s}(\mathbf{x})L(\mathbf{x},\omega)}_{\text{out-scattering}} \\ &= \underbrace{L_\mathrm{v}(\mathbf{x},\omega)}_{\text{contribution}} - \underbrace{\sigma_\mathrm{t}(\mathbf{x})L(\mathbf{x},\omega)}_{\text{extinction}}.\end{aligned} \tag{2.25}$$

We can see that the loss of energy due to absorption and out-scattering is expressed together as extinction (Equation 2.25). In contrast, energy increases due to the combined effect of emission and in-scattering. This is expressed by the *volume contribution* radiance L_v defined as

$$\begin{aligned}L_\mathrm{v}(\mathbf{x},\omega) &= L_\mathrm{e}(\mathbf{x},\omega) + \sigma_\mathrm{s}(\mathbf{x})L_\mathrm{i}(\mathbf{x},\omega) \\ &= L_\mathrm{e}(\mathbf{x},\omega) + \sigma_\mathrm{s}(\mathbf{x}) \int_\Omega f(\mathbf{x},\omega',\omega)L(\mathbf{x},\omega')\,\mathrm{d}\mu(\omega').\end{aligned} \tag{2.26}$$

While the emitted radiance L_e is included trivially, we can see that the in-scattered radiance L_i expands into a convolution of the phase function f and the incoming radiance L within the spherical domain Ω. It is precisely this operation that changes the directionality of light passing through a medium, and is consequently responsible for all directionally dependent visual phenomena that participating media exhibit.

Now the key step to obtaining the volume rendering equation (VRE) is to integrate both sides of Equation 2.25. By doing so we obtain a 'measurement' of the total radiance L along the optical path

in the direction ω, starting at $\mathbf{x}$:

$$L(\mathbf{x},\omega) = \underbrace{\int_0^s T(\mathbf{x},\mathbf{x}_t)\cdot L_\mathrm{v}(\mathbf{x}_t,\omega)\,\mathrm{d}t}_{\text{total volume contribution}} + \underbrace{T(\mathbf{x},\mathbf{x}_s)\cdot L_\mathrm{s}(\mathbf{x}_s,\omega)}_{\text{surface contribution}}. \tag{2.27}$$

This is depicted in Figure 2.8. The integral on the right side accumulates all energy that the volume emits or in-scatters into ω, attenuated by the transmittance function T at the respective location $\mathbf{x}_t$. The integration is stopped at the closest surface point $\mathbf{x}_s$, which additionally contributes the radiance reflected by this surface. This can be expressed as

$$L_\mathrm{s}(\mathbf{x},\omega) = L_\mathrm{e}(\mathbf{x},\omega) + \int_{\Omega^+} r(\mathbf{x},\omega',\omega)L(\mathbf{x},\omega')\langle\mathbf{n},\omega'\rangle\,\mathrm{d}\mu(\omega'), \tag{2.28}$$

where $\mathbf{n}$ is the surface normal and r a hemispherical function called the *bi-directional reflectance distribution function* (BRDF) – it is essentially the surface equivalent of phase function.

It can be observed that there are only minor differences between the surface contribution (Equation 2.28) and the volume contribution at a given point (Equation 2.26). The surface contribution is gathered from only the upper hemisphere Ω^+ (in respect to $\mathbf{n}$) and this further necessitates the use of a different distribution (a hemispherical distribution r instead of a spherical distribution f). In fact, in the absence of any participating medium (i. e., $\sigma_\mathrm{a} = \sigma_\mathrm{s} = 0$ and $L_\mathrm{e} = 0$) the volume contribution in Equation 2.27 will be zero and hence it reduces to the standard surface rendering equation [Kajiya 1986]. Since we focus on solving the volumetric portion of light transport, we will not discuss this part further and refer the reader to the standard texts [Dutré et al. 2006; Pharr and Humphreys 2010] instead.

2.5 Methods for Solving the Light Transport Problem

The first effort in numerically solving transport problems in participating media comes from the areas of nuclear physics and heat transport [Mishra and Prasad 1998]. Blinn [1982] has been one of the first to bring considerations about participating media into computer graphics. Since then a lot of effort has been invested into solving the related problems – see, e. g., the (somewhat dated) survey of Cerezo et al. [2005].

This section provides an overview of methods that attempt to solve the light transport problem, formulated either by the RTE (Equation 2.25), the VRE (Equation 2.27), or a simplified version of one of them. Rather than providing a chronological overview we divide these methods into categories, and that in a hierarchical manner with the focus on the ones closely related to this work. We will see that, somewhat paradoxically, many of these methods were adapted from solutions of the RE on surfaces, in spite of the fact that transport problems in other fields were primarily defined for participating media.

2.5.1 Stochastic Methods

The concept behind stochastic methods is to compute the solution by probabilistic sampling of the function that defines a given problem. These methods are usually unbiased or consistently biased, and also very general thanks to small numbers of assumptions they typically make (see the discussion in Section 1.2). On the other hand they are always subject to variance stemming from the stochastic sampling they perform.

Monte Carlo methods The most general and direct implementation of the above idea are MC methods. Suppose we want to evaluate a (definite) integral in the form of

$$I = \int_{\mathcal{D}} f(x)\, \mathrm{d}\mu_{\mathcal{D}}(x), \tag{2.29}$$

where $\mathcal{D}$ is a domain with any (even infinite) number of dimensions and $\mu_{\mathcal{D}}$ the measure on this domain. MC methods approximate this integral using the following probabilistic *estimator* $\hat{I}_n$:

$$\hat{I}_n = \frac{1}{n} \cdot \sum_{i=1}^{n} \frac{f(X_i)}{p(X_i)}. \tag{2.30}$$

Here $X_i \in \mathcal{D}$ are random variables sampled according to the probability density function (PDF) p. From the linearity of the expected value $\mathrm{E}[X]$ if follows that since $\mathrm{E}[f(X)/p(X)] = I$ then also $\mathrm{E}[\hat{I}_n] = I$ for all n. Note that, as $\hat{I}_n$ is defined in terms of the random variables X_i, it is a random variable itself and will differ from I by an error $\overline{\epsilon}_n \propto \mathrm{Var}[\hat{I}_n] = \mathrm{Var}[\hat{I}_1] \cdot n^{-1/2}$. However this error normally tends to zero unless $\mathrm{Var}[\hat{I}_1] = \infty$ (which under some conditions can happen).

The most direct MC method is path tracing, first introduced in graphics by Kajiya [1986] and extended to participating media by Pattanaik and Mudur [1993] and Hanrahan and Krueger [1993], for normal media and subsurface scattering respectively. Path tracing (PT) directly applies the estimator from Equation 2.30 to evaluate the VRE (Equation 2.27) by generating light paths: starting at the camera, each path is built independently by incrementally adding vertices until a light source is found. Each vertex corresponds to an interaction with scene objects or medium. For efficiency sake the length of each path can be limited by heuristics such as Russian roulette (unbiased) or relative thresholding (biased).

By combining path tracing with light tracing [Arvo 1986], Veach and Guibas [1994] have proposed bi-directional path tracing (BPT) which was extended to light transport in media by Lafortune and Willems [1996]. BPT traces paths both from the camera and light sources and optimally combines them by unbiased heuristics [Veach and Guibas 1995]. In addition, BPT is frequently used as the starting point for Metropolis light transport [Veach and Guibas 1997; Pauly et al. 2000; Raab et al. 2006], a class of methods belonging to the sub-category of Markov chain Monte Carlo approaches.

Caching-based methods The main insight that motivates the use of caching in light transport solutions is that the sampled functions are actually smooth in most of the scene domain, with discontinuities caused by visibility variation or other strong sources of illumination anisotropy. Caching methods therefore store parts of the VRE (the radiance itself, visibility, etc.) and then re-use the cached information multiple times, which usually involves some kind of reconstruction. This is typically done in multiple stages called *passes*.

Arguably the most widespread caching method is photon mapping (PM) originally proposed by Jensen [1996], and together with Christensen [1998] extended it to volumetric media. The basic version of the algorithm works in two passes as follows:

1. **Tracing** All emitted flux in the scene is uniformly divided among n particles simply called *photons*. These are traced through the scene using random walks, in the same way as in other MC methods. The only difference is that every time a photon interacts with a surface or medium its incident flux is stored in a photon map (a spatial data structure, e. g., a kD-tree or regular grid). At the end of the first pass the stored photons jointly represent the energy distribution in the scene.

2. **Reconstruction** The illumination is reconstructed by tracing short paths from the camera into the scene. When a path hits a surface at a location $\mathbf{x}$ the resulting outgoing radiance is reconstructed by a kernel density estimate from the m nearest photons as

$$L_s(\mathbf{x},\omega) \approx \sum_{i=1}^{m} r(\mathbf{x},\omega_i,\omega)\frac{k_d(\|\mathbf{x}-\mathbf{x}_i\|)\cdot\phi_i}{\pi d^2 K}, \tag{2.31}$$

where d is the distance to the farthest of the m photons, ϕ_i and ω_i are the i-th photon's stored flux and incident direction, r is the surface BRDF and finally $K = \sum_i k_d(\|\mathbf{x}-\mathbf{x}_i\|)$ is the normalisation constant corresponding to the reconstruction kernel k_d (typically Gaussian). The volumetric estimate is similar, except that it needs to be executed *along* the entire portion of the path which intersects the medium. Also the normalising term πd^2 becomes $\frac{4}{3}\pi d^3$ as the kernel is spherical instead of circular.

Significant improvements to volumetric PM have been achieved by Jarosz et al. [2008b; 2011a] by extending the dimensionality of both the traced primitives and the estimate. The state of the art in photon mapping methods can be found in the recent report by Hachisuka et al. [2013]. Additionally, a unified way to combine different kinds of estimation methods (even beyond density estimation) has been described by Křivánek et al. [2014].

Other methods that rely on storing an re-using illumination information include radiance caching [Jarosz et al. 2008a; Moon et al. 2008], instant radiosity with virtual point lights [Engelhardt et al. 2010; Engelhardt et al. 2012] and its ray-based extension [Novák et al. 2012b]. Caching has also been successfully applied to render sub-surface scattering, most notably using analytic approximations such as diffusion [Stam 1995] and dipole [Jensen et al. 2001] models – these approximations are deterministic, but still utilise stochastic methods to compute the initial energy distribution in the scene.

Beside noise, the main issue of stochastic methods are optically dense, high-albedo media, since these methods' complexity is linear in the number of scattering events that light undergoes on average. This might not seem significant, but in effect it might take hundred times longer to render a glass of milk than a room full of smoke. A partial remedy can be the application of similarity theory [Wyman et al. 1989; Zhao et al. 2014], but only by a constant factor.

Anisotropic scattering can be handled by stochastic methods, although much higher sampling rates are necessary in very anisotropic media (for example, four times more samples were necessary to render the results in Figure 2.5 for $g = 0.95$ than for the other, lower g values). In pure MC methods this is caused by the increased difficulty of sampling high-frequency distributions that are induced by anisotropic scattering. Caching methods require higher cache densities since anisotropic scattering decreases the robustness of the reconstruction techniques that these methods respectively utilise (such as the kernel density estimate in PM, Equation 2.31).

Progressive approaches On a high level the basic premise of progressive methods is to rapidly produce a coarse solution and then progressively refine it until a certain quality is met. These kinds of approaches exist outside computer graphics as well, e. g., in image processing or compression. In rendering, progressiveness is a natural part of most unbiased algorithms, with examples including regular and bi-directional path tracing, or Metropolis transport. Here it is a consequence of the estimator's expected value matching the true one, so that progressively converging to the true solution amounts simply to adding more samples.

In biased algorithms, however, additional considerations are necessary to ensure that the bias decreases over time and the solution converges correctly. In surface rendering, the basic photon mapping

[1996] has been formulated in progressive manners by Hachisuka et al. [2008; 2009] and Knaus and Zwicker [2011]. Jarosz et al. [2011b] have proposed a progressive variant of volumetric PM by utilising the same framework as Knaus and Zwicker. Furthermore, the combination of beam tracing and virtual lights by Novák et al. [2012b] has been made progressive as well [Novák et al. 2012a].

Denoting the error of pass i as ϵ_i, the average error after n passes is $\overline{\epsilon}_n = \frac{1}{n}\sum_{i=1}^{n}\epsilon_i$. Since each pass is assumed to be independent, the errors ϵ_i can be interpreted as realisations of a random variable $\overline{\epsilon}_n$. The expected value (bias) and variance (noise) of $\overline{\epsilon}_n$ can then be expressed as

$$\mathrm{E}[\overline{\epsilon}_n] = \frac{1}{n}\sum_{i=1}^{n}\mathrm{E}[\epsilon_i], \tag{2.32}$$

$$\mathrm{Var}[\overline{\epsilon}_n] = \frac{1}{n^2}\sum_{i=1}^{n}\mathrm{Var}[\epsilon_i]. \tag{2.33}$$

An algorithm converges to a stable solution if $\mathrm{Var}[\overline{\epsilon}_n] \to 0$ as $n \to \infty$ [1]. The main difference between unbiased and biased methods is that the average error $\mathrm{E}[\overline{\epsilon}_n] = 0$ in unbiased methods at any point during the solution computation; for biased methods this does not hold. However the bias in approaches such as progressive photon mapping is *consistent*, which means that also $\mathrm{E}[\overline{\epsilon}_n] \to 0$ as n approaches infinity and therefore the result converges to the true unbiased solution. This is achieved by decreasing the size of the density estimation region in each pass by a small factor; this of course increases the variance of each pass $\mathrm{Var}[\epsilon_i]$, however the *overall* variance of the solution $\mathrm{Var}[\overline{\epsilon}_n]$ still decreases.

Despite unbiased approaches being mathematically advantageous to the biased ones, often in practice the latter are preferred. The reason is their usually better robustness, meaning that the *actual* error of the solution $\overline{\epsilon}_n$ tends to be smaller in biased algorithms, albeit its expected value $\mathrm{E}[\overline{\epsilon}_n]$ is non-zero. An additional benefit that progressiveness can provide is interactive manipulation and feedback [Laine et al. 2007; Dammertz et al. 2010], allowing immediate changes to scene parametrisation. This is because the solution can be visualised after each pass, so as long as the passes can be made sufficiently short to provide a rapid response an artist can interactively edit the scene and then let the solution converge automatically afterwards. Because of this, progressive approaches are being adopted by an increasing number of production [Arnold 2015; Octane 2015] and research [Kroes 2015] renderers with volumetric transport capabilities. This is one of the factors that has motivated the progressive formulation of spectral ray differentials (Chapter 6).

2.5.2 Deterministic Methods

The core of all deterministic, or finite-elements (FE), methods is to discretise the domain where light transport takes place (the scene, in other words). The simplest such discretisation is a regular 3D grid (lattice) that divides the scene into a set of mutually exclusive elements, often referred to as *cells* or *sites* in volumetric methods. For reproducing anisotropic effects it is also necessary to discretise the directional domain, which is often done using basis functions. FE methods typically operate in an iterative manner, where each iteration computes the amount of energy that the discrete cells exchange. The exchange rate is defines by the so-called *form factor*, which loosely corresponds to the mutual (volumetric) visibility between the cells.

The above properties constitute the basic design space of FE methods – the main differences usually

[1] It does not matter whether n expresses the number of samples or passes, as there is usually a fixed proportion between the two.

lie in how the discretisation is performed, which groups of cells are considered during the energy exchange, and how are the form factors evaluated.

One of the first volumetric FE methods used in rendering was the zonal method by Rushmeier and Torrance [1987], albeit the concept and terminology originate from earlier work in general radiance transfer [Hottel and Sarofim 1967]. The zonal method simulates (isotropic) light transport as a *global* energy exchange between the cells, i. e., all combinations are considered. This is akin to the classic surface radiosity methods [Goral et al. 1984; Cohen et al. 1993]. Also in a similar fashion to these also the zonal method has been formulated in a hierarchical manner [Bhate 1993; Sillion 1994; Sillion 1995]. The use of this approach is limited due to the limits on the illumination directionality, but in some cases such as low-frequency atmospheric rendering this can be acceptable [Haber et al. 2005].

For handling anisotropic effects it is necessary to represent the illumination directionality. This can be done using constant basis functions or spherical harmonics expansion [Kajiya and Herzen 1984; Bhate and Tokuta 1992]. Here the common predecessor from classic radiance transport work is the *discrete ordinates method* (DOM) [Chandrasekhar 1960]. In contrast to the zonal method, these approaches are *local*, which implies that only a small neighbourhood around each cell is considered in the energy exchange. Considerations about the light directionality however only emphasise the main issues of FE methods [Lathrop 1968], mainly *false scattering* (i. e., numerical dissipation) and *ray effects* (selective over- or under-estimation of scattered light due to the discrete directional representation). Max [1994] attempted to solve this issue by a finer sampling in the directional domain, although a more accurate method based on coherent 'sweeps' through this domain has been proposed recently by Fattal [2009]. The diffusion approximation [Stam 1995] can also be used to accelerate the computation of local form factors, as employed by, e. g., Zhou et al. [2008b] to interactively render isotropic smoke. Another local method based on the lattice-Boltzmann (LB) propagation has been introduced by Geist et al. [2004]. LB transport uses a different propagation scheme than DOM, but its behaviour is otherwise similar, including convergence properties and the types of media that can be simulated.

Although most FE methods handle highly scattering media well, they only consider isotropic or moderately anisotropic scattering to avoid ray effects caused by their piecewise constant (LB) and spherical-harmonic (DOM) representations. This is necessary to avoid the numerical issues mentioned above. Despite isotropic scattering being acceptable in some cases, it is generally not a valid approximation since most real-world media exhibit relatively high anisotropy (with values $g \approx 0.9$ or more [Narasimhan et al. 2006]). We address this inherent problem in Chapter 5 by aligning the propagation domain(s) with approximate principal directions of the source illumination, and by using a custom propagation basis well suited to represent anisotropic scattering and radiance distributions.

2.5.3 Analytic Methods

It is widely conjectured that the radiative transport problem does not have an analytic solution for sufficiently general definitions of participating media. However, arriving to exact or at least reasonably accurate solutions is possible if certain simplifications are made about the medium and transport conditions.

There are two main obstacles in solving transport problems analytically. First, for general phase functions the RTE does not have a closed-form solution, even for the simplest anisotropic models such as the HG function (Section 2.3.3). Second, analytically defining boundary conditions for

basically any realistic medium is not feasible. Consequently, the methods from this category simplify the problem in either or both of these two regards.

Sub-surface scattering Approaches from this group focus on optically thick media with well defined boundaries, usually solids (stone, wax, plastic, organic tissues) and liquids (milk, juice). The main advantage of optically thick media is that it is reasonable to expect any high-frequency and anisotropic illumination features to dissipate very quickly so that isotropic transport can be used. In addition, solid and liquid media can be assumed to have hard (discrete) boundaries and only a small degree of internal heterogeneity. Combining the two factors, these methods try to solve the transport for semi-infinite media where the only boundary condition is the surface defined as an infinite plane.

Under such conditions it can be expected that light does not propagate very far under the medium surface which can therefore be treated as locally planar, leading to the concept of *bi-directional scattering-surface reflectance distribution function*. BSSRDF can be seen as an extended definition of BRDF (see Equation 2.28) but with additional spatial dependence to account for the sub-surface light propagation. In graphics, the work in this direction has been pioneered by Jensen et al. [2001], who defined a BSSRDF based on the so-called dipole model. The solution predicts that under the simple assumption of a semi-infinite isotropic medium the incident light will scatter according to a Gaussian profile, essentially amounting to diffusion.

Subsequent work has focused on treating some of the issues of the basic diffusion model, and generalising it to a broader range of media. Jensen and Buhler [2002] designed a more practical hierarchical two-pass approach based on the dipole model. Its further extension to multi-layer materials [Donner and Jensen 2005] has enabled the simulation or much richer materials, most interestingly human skin and other biological tissues [D'Eon et al. 2007; Donner et al. 2008]. D'Eon and Irving [2011] generalised the model for multiple Gaussians, so that thicker layers can be simulated in conjunction with more localised light sources.

The limitations to isotropic propagation has been addressed by Donner et al. [2009] by designing an empirical closed-form fit to MC-simulated data. This expression allows the simulation of low-albedo materials with anisotropic scattering, although no directional effects can be reproduced. The directional dipole by Frisvad et al. [2014] additionally includes the incident light direction in their expression. The hybrid approach based on photon beam diffusion by Habel et al. [2013] has similar properties and results, but in the context of stochastic methods. In spite of the great progress in this area, a full directional model capable of treating the directionality of both the incident and outgoing ('emergent') light has yet to be designed.

Sparse media An alternative to sub-surface scattering is to assume infinite but optically thin media, possibly allowing for anisotropic scattering. Such an approach is mainly suitable for various environmental conditions, such as scattering in atmospheres, fog, or deep water.

In such environments it is necessary to analytically describe light transport on a global scale, since the RTE (Equation 2.25) defines only the local behaviour of light in the presence of a participating medium, and the VRE (Equation 2.27) is not suitable for analytical integration unless significant simplifications are made about the source radiance function. One way to bridge these two paradigms on a larger scale is to use the *path integral formulation* [Tessendorf 1987; Premože et al. 2003; Premože et al. 2004; Tessendorf 2011]. However, instead of considering discrete paths as, e. g., Veach [1997], a continuous notion of path space is used. This is similar to the concept of *sum over particle histories* [Feynman and Hibbs 1965] widely used in quantum mechanics to statistically describe the propagation of individual particles or their agglomerations.

As Premože, Tessendorf and others observed, in light transport we can similarly obtain the light intensity in a certain point and direction by integrating all possible photon paths that conform to this single boundary condition. This can informally be written as

$$L \approx \int_{\text{all paths}} e^{-\tau(\text{path})}. \tag{2.34}$$

Here τ defines an *effective attenuation* of a path, essentially stating its contribution to the result in accordance with the Beer-Lambert-Bouguer law (Equation 2.21) – assuming that all paths are equivalent, i. e., that they initially carry the same radiant energy. Obviously the difficult part of this relatively intuitive idea is to rigorously express the infinitely-dimensional integral over the path space, and this has been the focus of the above-cited works.

A step towards formalising this idea is to consider a pair of boundary conditions, namely the starting coordinates of the path $\mathbf{x}', \omega'$ and its end $\mathbf{x}, \omega$. In this case the RTE can be reformulated using the *Green's function* G (also called Green's propagator or evolution operator [Green 1828]) as

$$L(\mathbf{x}, \omega) = \int \int G(\mathbf{x}, \mathbf{x}', \omega, \omega') L_e(\mathbf{x}', \omega') \, \mathrm{d}\mu(\mathbf{x}') \, \mathrm{d}\mu(\omega'), \tag{2.35}$$

with L_e being the initial radiance distribution in space. It can be seen that G expresses how the initial energy distribution changes when the individual path's boundary conditions are taken into account.

It is however not immediately obvious how this mathematical formulation relates to the underlying physical process. What the Green's function describes is an averaged high-level behaviour of many individual photons, which are randomly scattered in the medium. This method states that the probability of a particle being in a certain position in space is obtained by superposing all possible paths the particle might have taken within the given boundary conditions. Deriving increasingly accurate approximations to G has been the primary focus of Premože, Tessendorf et al. [1987; 2003; 2004; 2011], primarily using the conceptual framework of *most probable paths*.

The most directly useful for rendering is the result of Premože et al. [2004], which we employ in Chapter 3. This work considers only the *spatial* spreading of a collimated pencil of light between two points $\mathbf{x}$ and $\mathbf{x}'$. The multiple-scattering distribution after travelling a distance s is assumed to be Gaussian, akin to most of the aforementioned methods. The standard deviation ('width') W of the resulting Gaussian profile based on the travel distance s is then expressed as

$$W(\|\mathbf{x} - \mathbf{x}'\|) = W(s) = \sqrt{\frac{1}{2}\left(\frac{2\sigma_a}{3s} + \frac{4}{s^3 \sigma_s (1-g)}\right)^{-1}}. \tag{2.36}$$

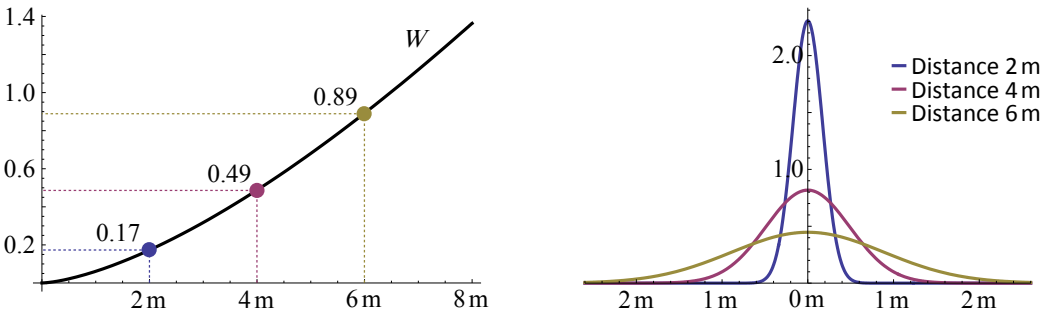


Figure 2.9: Behaviour of the function $W(s)$ (Equation 2.36) for $\sigma_a = 0.1$, $\sigma_s = 0.3$ and $g = 0.9$: plot of the function in dependence on the travel distance s *(left)* and the resulting Gaussian profiles for three different travel distances *(right)*. The corresponding standard deviations of the curves in *(right)* are marked in *(left)*.

The behaviour of Equation 2.36 is shown in Figure 2.9.

Another approach has been chosen by Narasimhan et al. by expanding the HG phase function into an infinite series using Legendre polynomials. The resulting expression for *angular* spreading can be used for analysing weather conditions [Narasimhan and Nayar 2003], direct rendering [Narasimhan et al. 2004], but also acquisition of volumetric parameters of diluted media [Narasimhan et al. 2006].

Air-light integral In conditions similar to the above, it is possible to arrive to an exact solution if only a single scattering bounce is considered. Sun et al. [2005] arrived to an analytic solution for isotropic in homogeneous media and point light sources. A significant extension has been achieved by Pegoraro et al. [2011] who can handle both anisotropic scattering an light sources. The limitation to single scattering is obviously quite restricting, but under some conditions can still provide relevant visual cues about the medium.

Additionally, for specific media with known conditions, good low-order approximations can be derived. A prime example are planetary atmospheres, where the density distribution is vertically exponential and radially symmetric. Utilising this knowledge, closed-form models with good visual fidelity can be derived [Preetham et al. 1999; Hošek and Wilkie 2012], again, based on analytic fitting to simulated data.

2.5.4 Interactive and Specialised Methods

Aside from the more general, core works (Sections 2.5.1 and 2.5.2) and their analytic approximations (Section 2.5.3) there is a whole spectrum of approaches focused primarily on interactive performance. Some of these methods are fast adaptations of the respective non-interactive versions, while others are built on certain simplifying assumptions or focus only on a particular group of media. Rather than a complete enumeration, a representative overview is provided.

Photon mapping Several attempts to speed-up photon mapping by continuously updating the photon map exist. Airieau et al. [2011] developed an interactive technique for continuous streaming of photons, where parts of the scene where illumination changes more rapidly are updated with higher priority. A similar (interactive) method has been described by Dmitriev et al. [2002], who utilised the periodicity properties of Halton sequences to identify the photon groups to update. Jiménez et al. [2005] then generalised this approach to participating media, which however causes the algorithm to perform about an order of magnitude slower than necessary for interactive frame rates. Notable GPU implementations of photon mapping were proposed by Purcell et al. [2003] and Zhou et al. [2008a], but only Krüger et al. [2006] consider participating media. Photons can be *splatted* from surfaces efficiently using the GPU instead of performing costly density estimations [Herzog et al. 2007; McGuire and Luebke 2009]. We employ this idea in Chapter 4 within the amortised volumetric photon mapping approach.

Local FE methods As already hinted in Section 1.2 the locality of transport methods is preferred on contemporary computing architectures, and it will likely remain that way. Both CPUs and GPUs benefit from local, coherent memory access patterns as their predictability improves caching efficiency. Due to their massively-parallel architecture, GPUs especially suffer from random access patterns that global methods use, as incoherent memory access often causes execution stalls. Efficient interactive methods based on the local paradigm have therefore been developed. Kaplanyan and Dachsbacher [2010] developed *light propagation volumes* (LPV) for surface light transport as an

adaptation of DOM [Chandrasekhar 1960], which has subsequently been extended to multiple scattering for various types of media [Billeter et al. 2012; Borlum et al. 2011; Koa and Johan 2014]. Bernabei et al. [2011] obtained interactive performance by combining LB propagation with photon tracing for initialising the lattice. The core limitation of these methods is similar to what has been described in Section 2.5.2: general spherical bases are used to represent the propagated energy (either constant or spherical harmonic), which significantly limits the anisotropy of the resulting illumination. Our proposed solution to this is to use a different propagation basis, presented in Chapter 5.

Precomputation Sometimes precomputation is used to speed up the rendering of translucent objects. Sloan et al. [2002] include simple participating media in their precomputed radiance transfer approach, which pre-computes the entire transport operator in the scene for a limited number of locations and their combinations. Static discretisations of the medium are used for instance by Wang et al. [2010] and Zhou et al. [2008b]. The former work uses a static tetrahedral discretisation and applies a BSSRDF approximation to compute multiple scattering. Zhou et al. decompose the medium into a low frequency representation and a high-frequency residual field, and solve the transport by applying the diffusion equation to the low-frequency portion. Since these approaches rely significantly on precomputation they are limited to static media. Similarly to the previous approaches, these methods are typically limited to isotropic scattering. We generally try to avoid these limitations as they decrease the range of media that can be simulated by a significant amount.

Low-order effects Approaches that consider only a subset of light paths in the medium are also common, partly because they usually map well to modern GPU architectures. They restrict themselves to modelling low-order or usually even single scattering, often focusing on individual optical phenomena to achieve interactive or real-time performance. These include light shafts [Ren et al. 2008; Engelhardt and Dachsbacher 2010], volume caustics [Krüger et al. 2006; Wyman and Ramsey 2008; Liktor and Dachsbacher 2011] and shadows [Lokovic and Veach 2000; Baran et al. 2010; Salvi et al. 2010]. Ihrke et al. [2007] use wavefront rendering to evaluate the Eikonal equation in volumes, considering continuous refraction and single scattering in heterogeneous media.

Cloud rendering Although specialised methods for rendering of clouds are more typical in the interactive domain, early non-interactive methods focused on simulating this phenomenon exist. Examples of these include works by Gardner [1985] and Nishita et al. [1996]. The interactive category they can roughly be divided into empirical and physically-based methods. From the former group we mention the classic billboard-based technique by Wang [2003] used for instance in Microsoft Flight Simulator 2004 or CryEngine2 [Wenzel 2006].

On the other hand, several different paradigms have been followed in the existing physically-based approaches (for a full review please refer to Bouthors [2008]). Riley et al. [2004] used the half-angle slicing technique of Kniss et al. [2002]. The method, however, is still semi-empirical and considers only the forward portion of multiple scattering and a single light source. The concept of Monte-Carlo illumination networks by Szirmay-Kalos et al. [2005] is more theoretically sound, its main drawback however is the necessity to recalculate the entire network from scratch every time the cloud density field changes. Another approach has been taken by Bouthors et al. [2006; 2008] who analysed the behaviour of light in plane-parallel homogeneous slabs and computed an ad-hoc analytic fit to the obtained data for a rapid evaluation of the cloud illumination. This approach produces interesting results, it is however limited to simple light sources and builds primarily on theoretical assumptions rather than on actual observations of the simulated environment, which as Chapter 4 shows we believe to be important for specialised methods.

Visualisation Various approaches can also be found in volume visualisation literature, e. g., half-angle slicing [Kniss et al. 2002] which empirically computes only the forward portion of multiple scattering. For an comprehensive overview please refer to Jönsson et al. [2014]. In general, volume visualisation works with similar phenomena, but focuses on different priorities such as a good level of data abstraction instead of physical realism.

Discussion We can observe that the interactive methods can be categorised in yet another way. The first group are approaches that can very rapidly simulate low-order scattering, but handle multiple scattering either in a supplemental way or even not at all. On the other hand, approaches that compute multiple scattering (typically) using lattice-based propagation or diffusion techniques fail at reproducing low-order scattering and anisotropic effects. In addition to the already discussed issues, another difficulty of anisotropic media lies in the fact that the transition from low- to high-order scattering is very gradual, and thus these effects need to be handled in a consistent manner. To remedy this situation is the target of Chapters 4 and 5.

2.6 Refraction and Dispersion

Refraction is a high-level phenomenon exhibited by the specific category of transparent media. In a fashion similar to scattering (Section 2.3.1), refraction is explained by the wave-optical framework, but can comfortably be simulated by geometric optics – even more so, given its deterministic nature.

The primary effect of refraction is the bending of light which enters a transparent medium such as glass or water (see Figure 2.10, a). The internal structure of transparent media causes light to 'average out' on the level of individual waves, so the effects characteristic to scattering (translucency, spreading, etc.) are not visible. In spite of that, scattering still occurs, causing a decrease of the phase velocity in which light propagates within the medium. Refraction is an apparent consequence of the sudden *change* of this velocity at the medium boundary. In addition, since these effects are typically wavelength-dependent, different parts of the visible spectrum are bent in different directions (Figure 2.10, b), causing the so-called *dispersion* (not to be confused with *diffraction*, a related but different phenomenon).

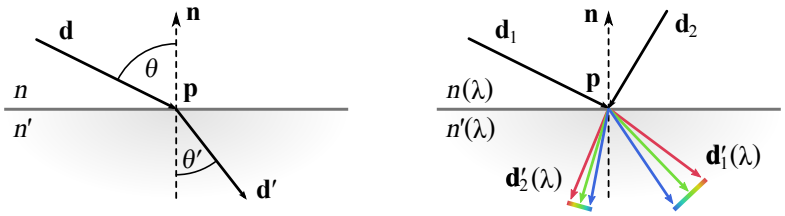


Figure 2.10: Refraction for a single monochromatic ray $\mathbf{d}$ *(left)* and dispersion of rays $\mathbf{d}_1, \mathbf{d}_2$ depending on wavelength λ *(right)*. Notice that the dispersion magnitude depends on the angle of incidence.

When a ray hits a smooth surface it is reflected and/or refracted depending on the nature of the object and the incoming direction. This is determined by the Fresnel's law [Born and Wolf 1999]. If refraction occurs its strength depends on the ratio of indices of refraction (IOR) between the original and the entered medium, commonly denoted by $\eta = \frac{n}{n'}$ (with n and n' being the respective media IORs). Dispersion occurs because η is usually not constant but depends on the light wavelength in vacuum λ (Figure 2.10). As with all other quantities we will assume wavelength-dependency from now on and contract $\eta(\lambda)$ to just η for better readability. Now, according to the Snell's law, light

propagating along the direction $\mathbf{d}$ will be refracted into $\mathbf{d}'$ as

$$\frac{\sin\theta}{\sin\theta'} = \frac{n}{n'}. \tag{2.37}$$

Here θ, θ' are the angles between $\mathbf{d}, \mathbf{d}'$ and the local vertical axis, i. e., $\theta = \arccos\langle \mathbf{d}, \mathbf{n}\rangle$ and $\theta' = \arccos\langle \mathbf{d}', -\mathbf{n}\rangle$, with $\mathbf{n}$ being the refractive interface's normal. In the vector form, Snell's law can be written as

$$\begin{aligned} \mathbf{d}' &= \eta\mathbf{d} - \nu\mathbf{n}, \quad \text{where} \\ \nu &= \eta\langle \mathbf{d}, \mathbf{n}\rangle + \sqrt{1 - \eta^2 + \eta^2\langle \mathbf{d}, \mathbf{n}\rangle^2}. \end{aligned} \tag{2.38}$$

Consequently, the refracted direction also becomes dependent on the wavelength λ.

Realistic image synthesis While many renderers assume RGB or other trichromatic models, dispersion is a typical example where spectral rendering excels. Stochastic Monte-Carlo methods, both the forward-tracing [Arvo 1986; Jensen 2001] backward-tracing ones [Kajiya 1986; Thomas 1986], can be trivially extended to support dispersion. Regardless of the particular algorithm, spectral rendering introduces an additional dimension (the spectral domain) into the rendering equation. The prevalent strategies to sample this domain are stochastic sampling, which leads to chromatic noise (cf. [Wilkie et al. 2014]), and regular sampling in uniform intervals (so-called *bands*), which causes aliasing artifacts that manifest as colour fringes. The overhead of spectral rendering is due to the higher memory and computational costs of storing and processing colour spectra and also due to the fact that more rays need to be traced as each path is monochromatic or carries only a small fraction of the spectrum. Please refer to Chapter 6 for additional discussion.

Light tracing and photon mapping are the primary alternatives that connect light paths to the sensor for simulating caustic phenomena that are difficult to capture when only eye paths are considered. Caustics are a particularly intriguing manifestation of reflection and refraction, where light interacts at a smooth interface and then concentrates at a diffuse (rough) surface. In the case of dispersive interaction, caustics can have a coloured rainbow-like appearance. Photon mapping naturally extends to spectral rendering, but special considerations need to be taken when a photon interacts with a dispersive interface and the aforementioned challenges in storing and processing spectral information persist (cf. [Lai and Christensen 2007]).

Lens simulation Dispersive and other chromatic aberration phenomena are prominent in both the human eye and optical lens systems. When simulating these, blurring has been used to reduce variance between colour bands such as for apertures simulated with Fraunhofer diffraction [Kakimoto et al. 2004], the human eye [Ritschel et al. 2009] or lens systems [Hullin et al. 2011]. In these, either the special properties of diffraction are exploited (Fraunhofer diffraction at one wavelength is a scaled copy of another wavelength) or the blur size and direction is defined by the lens system. In comparison, spectral ray differentials allow to trace this information directly.

Real-time rendering Using different directions to perform look-ups into an environment map is a classic approximation technique for rendering dispersive refractions [Kilgard 2001]. Artists can use image editing tools to turn a regular refraction in an RGB image into a dispersive refraction with coloured edges by blurring with an appropriate empirical filter. One exceptional use of ray differential-like considerations is the filtering of caustics by Wyman and Dachsbacher [2008]. Without considerations about the spectral behaviour of light transport, however, the results suffer from under-sampling where a spatially blurred version of discrete colour bands is visible instead of a continuous spectrum with sharp geometric features.

Chromatic aberration Filtering between different colours is a classic task when dealing with sensors [Laroche and Prescott 1994]. Chromatic aberration is present in many optical systems, and commercial software such as Adobe Photoshop offer tools to reduce it. A careful analysis of chromatic aberrations and ways to reduce them can be found in Kang [2007]. Boult and Wolberg [1992] proposed a warping-based approach to remove aberration, that is, with the opposite intent than simulating it. Remarkably, dispersive effects are present in all real images but often are missing in synthetic images [Johnson and Farid 2006], a fact that may be used to detect forged images.

2.7 Reconstruction

Discrete sampling and reconstruction of continuous signals are topics of high importance for rendering. Pharr and Humphreys [2010] provide an excellent overview, especially with regard to image-space sampling, reconstruction and resulting aliasing issues. General high-dimensional sampling is extensively reviewed by Veach [1997]. For the purposes of this thesis, the more directly relevant is the analysis and reconstruction of illumination, since the method proposed in Chapter 6 focuses on doing so in the specific context of dispersive refraction.

Statistics Simple statistical methods can be used to analyse and reconstruct illumination. In addition to the kernel density estimation widely used in photon mapping and other approaches that use caching in unordered point representations (see Section 2.5.1), a good example is the irradiance filtering approach by Kontkanen et al. [2004] based on local variance estimation. More advanced methods, such as greedy error minimisation [Rousselle et al. 2011] or unbiased risk estimation [Li et al. 2012], can be used to adaptively sample and reconstruct global illumination in general stochastic settings.

Fourier analysis Light transport in the frequency domain has been theoretically analysed in the seminal work of Durand et al. [2005]. Sheared filters within this domain can be used to reconstruct complex multi-occluder soft shadows [Egan et al. 2011] and even combined distribution effects [Yan et al. 2014]. Due to the cost of sheared filters, however, using simpler filtering in the primal domain or even image space is preferred (see also Section 2.8). With some assumptions, approximations by axis-aligned Gaussians have shown to be feasible to reconstruct effects like soft shadows [Mehta et al. 2012], diffuse inter-reflection [Mehta et al. 2013], or motion blur and depth of field [Belcour et al. 2013; Mehta et al. 2014]. Similar techniques based on covariance matrix tracking can even be used to filter and adaptively sample low-order scattering [Belcour et al. 2014].

Ray differentials Fourier analysis is mathematically quite involved and requires making certain compromises in order to derive the reconstruction filter parameters. This is even more so in the case of more advanced, nonlinear effects, such as dispersion in the context of this thesis. A more tractable alternative is ray differentiation, with the added benefit of an easier incorporation to ray-based rendering methods.

The concept of ray differentials has been introduced by Igehy [1999] and is commonly used to improve for instance texture filtering. Ray differentials describe the spatial change of a ray (that potentially undergoes several interactions) when changing its sensor coordinate by a small, differential step. Later, path differentials [Suykens and Willems 2001] extended the idea to full paths, which can start at the camera or the light. Photon differentials [Schjøth et al. 2007] are a special case of path differentials for light paths, where the traced footprint is used to improve density estimation when connecting to the eye paths. Fabianowksi and Dingliana [2009] have used differentials for reconstruction on diffuse surfaces in the context of photon mapping. So far, no previous work has

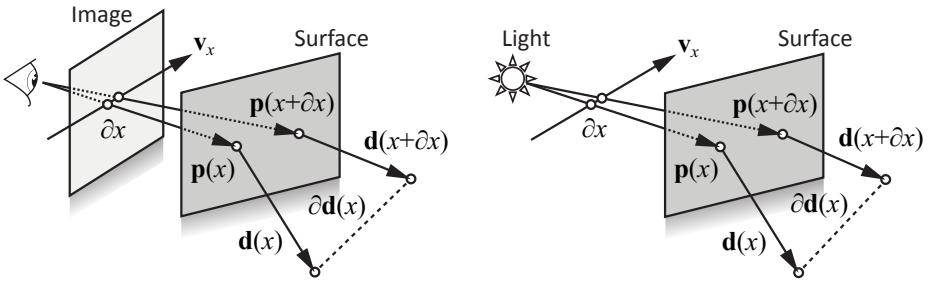


Figure 2.11: Concept and notation of the classic ray differentials, for eye paths *(left)* and light paths *(right)*.

considered using ray differentials within the spectral domain, which is the aim of Chapter 6.

The classic ray differentials describe how the position and direction of a ray change when its initial sensor position is differentially offset (please refer to Figure 2.11). Here the sensor is assumed to be a simple pinhole camera. Let a position $\mathbf{p}$ and direction $\mathbf{d}$ define a ray $\mathbf{R} = (\mathbf{p}, \mathbf{d})$, and x and y be the sensor coordinates. Let also $\mathbf{R}(x,y)$ be the ray through that pixel. The initial ray position $\mathbf{p}(x,y)$ is set to the camera position $\mathbf{e}$ for all rays starting at the sensor. The ray direction $\mathbf{d}(x,y) = \mathbf{v} + x\mathbf{v}_x + y\mathbf{v}_y$ depends on the view direction $\mathbf{v}$ and its reference frame spanned by $\mathbf{v}_x$ and $\mathbf{v}_y$. The ray differential is then a pair of partial derivatives with respect to the sensor coordinates x and y:

$$\frac{\partial \mathbf{R}}{\partial x} = \left(\frac{\partial \mathbf{p}}{\partial x}, \frac{\partial \mathbf{d}}{\partial x}\right) \quad \text{and} \quad \frac{\partial \mathbf{R}}{\partial y} = \left(\frac{\partial \mathbf{p}}{\partial y}, \frac{\partial \mathbf{d}}{\partial y}\right). \tag{2.39}$$

Igehy [1999] presented closed-form solutions for transport (rays travelling in free space), reflection and *monochromatic* refraction. For more complex camera models (e.g. thin-lens), finite differences can be used to compute the initial directional differential.

The application of ray differentials has two phases. First, each ray maintains its differential as it is traced through the scene, updating it at each interaction. The reconstruction itself then depends on the particular rendering algorithm – in path tracing, for example, an approximate 2D filtering footprint is determined from the positional differential $(\frac{\partial \mathbf{p}}{\partial x}, \frac{\partial \mathbf{p}}{\partial y})$.

2.8 Image Formation

In a direct relation to image and illumination reconstruction (Section 2.7) is filtering and methods that operate in image space based on it. Filtering is a canonical operation in image processing and traditionally performs removal or modification of a certain frequency range in an image. Recently, however, filtering-like methods have also become popular for performing various tasks in computer graphics, including synthesis. In our context, the screen-space method presented in Chapter 3 is directly based on this paradigm, and Chapters 4 and 5 also employ techniques from this category.

Filtering Arguably the simplest way to filter an image is convolution with a linear (usually *low-pass*) filtering kernel k, such as the box function or more frequently (truncated) Gaussian kernel [Gonzalez and Woods 2001]. Applying large filters can however be quite costly – discrete convolution in (2D) image space has the complexity of $O(nm^2)$ where n is the number of image pixels and m is the filter size; for filters with size comparable to the image $m \approx \sqrt{n}$ the complexity becomes $O(n^2)$, which is often prohibitive. Most isotropic kernels are separable and can be applied in $O(nm)$ time. In addition, convolution can be performed in the frequency domain with the complexity of $O(n \log n)$, regardless

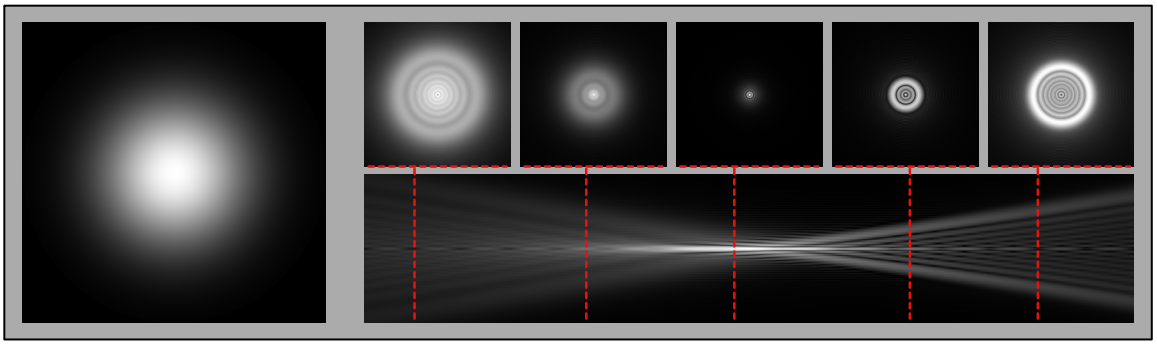


Figure 2.12: Two examples of point spread functions: circular Gaussian with the standard deviation of 3 *(left)* and a circular aberration of an optical system in different de-focus levels *(top right)* with the corresponding longitudinal section *(bottom right)*.

of the filter size. However, neither of these optimisations can be applied to *spatially varying* filters. An interesting alternative is pyramidal filtering [Burt 1981], which has the complexity of $O(n \log \sqrt{n})$ and can approximatively deal with spatially variant filters.

However, most interesting filters are not only spatially variant but also anisotropic and non-linear. This is because local adaptivity is required to preserve features of the original signal. The simplest feature-preserving technique is bilateral filtering [Tomasi and Manduchi 1998] which includes the image range in calculating the pixel distance that the kernel operates on. Adding other (so-called *guiding*) signals yields a joint (also cross) multi-lateral filter, which can be used, e. g., for increasing the apparent resolution of a signal that is only available in low-resolution or is costly to obtain [Kopf et al. 2007]. Multi-lateral filters are however even more expensive, especially if the filtered signal has more than two dimensions (such as in Chapters 4 and 5, where joint bilateral upsampling is used on 3D data). For this reason, various advanced equivalents to multi-lateral filters have been developed, such as guided filtering [He et al. 2010], Gaussian kD-tree filtering [Adams et al. 2009], domain transform [Gastal and Oliveira 2011], or adaptive manifolds [Gastal and Oliveira 2012].

One of the typical uses of filtering for rendering purposes is removing noise from MC solutions. More advanced methods employed in this context include guided filtering [Bauszat et al. 2011] and non-local means filtering [Rousselle et al. 2012]. Better results can be achieved if additional information be embedded in each pixel, for instance surface-incident radiance [Schwenk 2013] or sample colour histograms [Delbracio et al. 2014]. Also, since the noise in MC images tends to vary spatially and comes from different sources than in natural images (as considered by image processing), additional effort is needed to determine its actual amplitude and resulting filter strength [Sen and Darabi 2012; Kalantari and Sen 2013]. Suppression of 'fireflies' (i. e., high-amplitude noise) can be done efficiently using outlier detection and removal [DeCoro et al. 2010], even beyond the image domain.

Point spread function The aim in Chapter 3 is the development of an approximate image-space method for modelling light transport in homogeneous media. The most natural approach of doing so is to use a PSF model. In general, PSF is a function $f(\mathbf{x}) \in \mathbb{R}^2 \to \mathbb{R}^+$ which defines the spreading (i. e., distortion) of a point signal, which is usually represented by the Dirac pulse function. Figure 2.12 shows two characteristic examples. Phenomena modelled by PSFs usually cause blurring or other distortion of the original signal and appear for various causes, e. g., flaws or limitations of measurement systems, or, as in our case, as a direct result of a physical process.

Akin to filtering, a PSF is normally applied by convolving the original signal with a particular

expression of the PSF, which can be discrete and tabulated or might have a closed analytical form. Unfortunately as is often the case in filtering, PSFs are potentially spatially varying, and hence do not conform to the definition above. We therefore define the functional that maps every spatial location to its corresponding PSF $F(\mathbf{x},\mathbf{y}) \in (\mathbb{R}^2 \times \mathbb{R}^2) \rightarrow \mathbb{R}^+$. Applying this spatially varying, distance-dependent PSF to an image is not a simple convolution with the PSF itself, but a convolution-like operation with a kernel that is only similar to the PSF. Additionally this kernel will likely be anisotropic, although the original PSF might have been isotropic.

Screen Space Approximating costly computation of 3D illumination effects by simple 2D image operations has become increasingly popular, especially in practical interactive applications such as computer games. The general idea behind such *screen-space* methods is that modern GPUs produce more than just RGB colour when making use of deferred shading pipelines [Deering et al. 1988]; often per-pixel depth, normal or reflectance information complements the colour framebuffer. Screen-space techniques utilise this additional information and, in parallel over all pixels, compute advanced shading effects from it.

Ambient occlusion [Mittring 2007] was one of the first effects to be simulated in screen-space, followed by later extensions to directional occlusion and indirect light [Ritschel et al. 2009]. As for participating media, Jimenéz et al. [2009] simulated sub-surface scattering in human skin using image-space blurring with an approximate diffusion profile. Another approach to convey the presence of sparse media has been proposed by Lopez-Moreno et al. [2008], who however aimed at their artistic depiction without the focus on efficiency or physical plausibility. Besides light transport, approximations to depth-of-field [Rokita 1993; Lee et al. 2009], glare [Kawase 2005] or motion blur [Ritchie et al. 2010] are of practical importance.

In line with the previous discussion in this section, all screen-space methods face the difficulty of balancing between fast and simple but approximate local linear filtering and complex non-linear filtering that has to account for a high number of neighbouring pixels. Again, PSFs define how much a pixel contributes to other pixels (resulting in the 'scatter-type' access patters). This is not identical to functions that encode for every pixel how much to 'gather' from its neighbours. Nonetheless in these methods, scatter-type computation is routinely replaced by gathering ones as they fit the execution on modern GPUs much better.

The problems with large filter kernels however remain. The aforementioned filtering optimisations certainly help, notably the class of pyramidal filtering methods [Burt 1981]. In practical terms this amounts to building an appropriately down-sampled (usually Gaussian) MIP map, which in addition can employ anisotropic kernels to avoid the problems stemming from a strong spatial variation of the underlying PSF. Such MIP maps contain only values that actually should contribute to the result, depending on additional criteria such as scene depth. Our screen-space scattering (Chapter 3) or the depth-of-field method by Lee et al. [2009] follow this approach. Alternatively, the computation can be accelerated using stochastic sub-sampling [Jimenez et al. 2009] or separable filtering [Huang et al. 2011], if applicable. Although, diagonal structures or high-dynamic-range content are likely to cause artefacts with both simplifications.

Naturally, the biggest limitation shared by all screen-space methods is that objects or lights which are not visible in the framebuffer do not contribute to the image. Still, in performance-oriented applications, screen-space methods – due to their efficiency, ease of implementation and control – are likely to remain the solution of choice until the full simulation of distribution effects [Cook et al. 1984] becomes feasible.

Chapter 3

Screen-Space Scattering

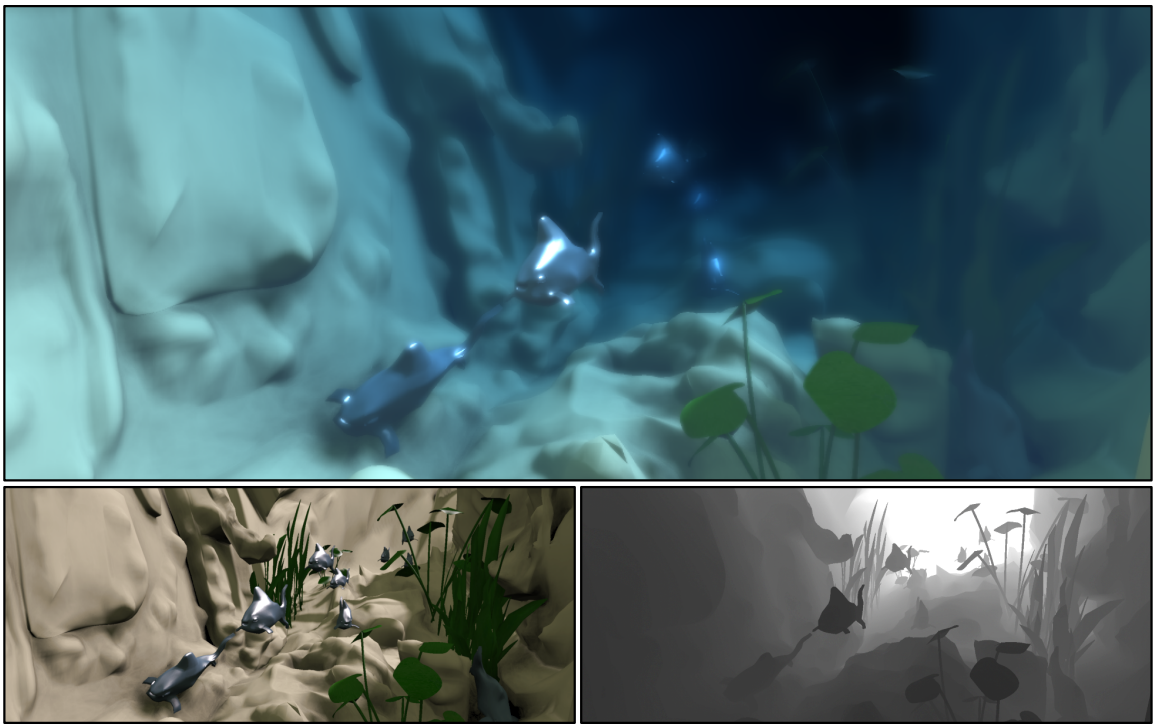


Figure 3.1: The presented method reproduces blurring and colour shifts in participating media, such as sea water, from just a single HDR image and its depth buffer *(bottom)* in real-time (4.3 ms for the scattering, 40 ms framebuffer generation (HDR, PCF soft shadows, SSAO), 2048×1024 resolution).

3.1 Introduction

This chapter focuses on rendering sparse (or more precisely, optically thin) homogeneous media in high real-time speeds, while still retaining physical plausibility. Prime examples of such conditions are foggy outdoor scenes and underwater environments (see an example in Figure 3.1).

As Chapter 2 demonstrates, light scattering is a complex physical phenomenon and under general

conditions it is indeed difficult to simulate. On the other hand, understanding the high-level behaviour of this process is easier. For instance, if a beam of collimated light enters a medium its photons will gradually scatter, resulting in a blurring of the beam. The blur magnitude will increase with advancing penetration, as well as higher scattering coefficient of the medium.

Based on this intuition, it is conceivable to design a method that approximates scattering as a blurring operation in image space. Akin to other image-space methods (Section 2.8), instead of considering individual light sources in 3D space explicitly, *every pixel* of the rendered image is implicitly a light source. The blurring of each pixel will be proportional to its distance from the camera and the optical thickness of the medium itself. Accounting for the scattering of the visible portion of the scene then roughly entails

1. obtaining the corresponding distance-dependent blurring kernel (i. e., PSF) at every pixel, and
2. convolving the rendered scene image with this kernel.

Section 2.5.3 discusses some possible approaches to obtain a physically plausible PSF for modelling the desired effects. The most relevant are the works of Narasimhan et al. [2003; 2004] and Premože et al. [2003; 2004]. The presented approach utilises the results of the latter work, since the resulting PSF is conveniently expressed by a Gaussian (instead of an infinite series expansion).

To avoid convolution by large spatially varying filter kernels, we adapt and improve a variant of hierarchical anisotropic filtering [Burt 1981; Lee et al. 2009]. The main improvements (described in Section 3.2) lie in modifying the anisotropic Gaussian filter that prevents the main issue of these approaches—*illumination leaking*—with greater efficiency. Additionally, a way to relate the per-pixel distance-dependent amount of light scattering with the corresponding level of the filtering hierarchy (represented by a standard MIP map) is described.

3.2 Screen-Space Scattering

This section exposes the SSS method. An overview is provided first, followed by detailed descriptions of all the core algorithmic steps.

Overview The input is an HDR image $\mathsf{I} : \mathbb{N}^2 \to \mathbb{R}^3$ containing outgoing scene radiance (in the sensor's direction) and a distance map $\mathsf{D} : \mathbb{N}^2 \to \mathbb{R}$ of the same resolution. D contains the world-space distance of every pixel to the camera; the common depth buffer is approximately such a distance map. The output is an image J which includes the scattering and absorption caused by a specified homogeneous medium.

The challenge of rendering participating media efficiently in screen space is to avoid performing convolutions by large and spatially varying PSF kernels. Instead, SSS approximates scattering computation by a *gathering* computation and performs only constant per-pixel-time local image modifications, combined with a specially constructed anisotropic Gaussian MIP map. The algorithm proceeds in three steps (Figure 3.2), each of which operates fully in parallel over image pixels:

1. First, the attenuation of the medium is accounted for. This step produces three images: $\mathsf{I}' : \mathbb{N}^2 \to \mathbb{R}^3$ which contains the radiance that did not interact with the medium at all, $\mathsf{I}^{[0]} : \mathbb{N}^2 \to \mathbb{R}^3$ containing the radiance that was scattered in the medium, but not absorbed, and $\mathsf{W}^{[0]} : \mathbb{N}^2 \to \mathbb{R}$ which stores additional distance-dependent information necessary to build the MIP map in the next step. Section 3.2.1 describes this step in detail.

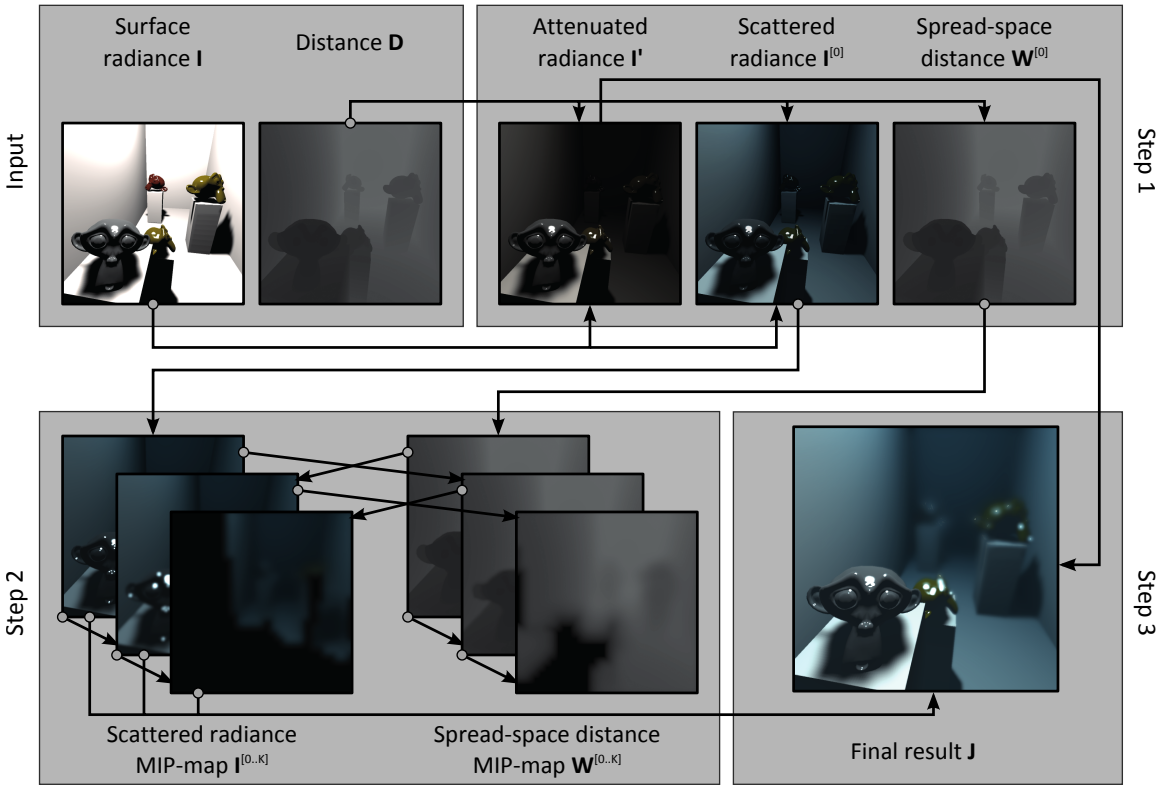


Figure 3.2: The three stages of the proposed algorithm shown for a box filled with water and glossy objects. The input is first split into attenuated and scattered radiance as well as spread-space distance. Second, the scattering and spread-space MIP maps are constructed (the first three levels are shown). Finally, the result is produced by combining the attenuated and the scattered radiance blurred according to the spread-space distance.

2. The second step produces an K-level MIP map $\mathsf{I}^{[1..K]}$ out of $\mathsf{I}^{[0]}$. This MIP map is used to approximate the spatially-varying filter required in Step 3. Every level of $\mathsf{I}^{[1..K]}$ contains an increasingly blurred version of $\mathsf{I}^{[0]}$. In contrast to classic Gaussian MIP maps, SSS accounts for the non-linearity caused by applying the scattering PSF by a Gaussian gather function. To this end, a second MIP map $\mathsf{W}^{[0..K]}$ storing the so-called *spread-space* distance is constructed and used. This is the core step of the algorithm and is detailed in Section 3.2.2.
3. Finally, the distance s of every pixel in I stored in D is used to select a MIP level to fetch an approximation of the convolution of I with a Gaussian similar to $W(s)$ (Equation 2.36). This fetched value is then simply composited with I', producing the final image J. Section 3.2.3 describes this step.

3.2.1 Preprocessing

This step produces the images I', $\mathsf{I}^{[0]}$ and $\mathsf{W}^{[0]}$ (see Figure 3.2, Step 1):

$$\mathsf{I}' = e^{-\sigma_t \mathsf{D}} \cdot \mathsf{I} \tag{3.1}$$

$$\mathsf{I}^{[0]} = e^{-\sigma_a \mathsf{D}} \cdot (1 - e^{-\sigma_s \mathsf{D}}) \cdot \mathsf{I} \tag{3.2}$$

$$\mathsf{W}^{[0]} = W(\mathsf{D}) \tag{3.3}$$

Equation 3.1 calculates the attenuated radiance according to the Beer-Lambert-Bouguer law, that is, the fraction of light that reaches the camera directly without interacting with the medium. This is the original colour of the scene, but attenuated in proportion to the pixels' distance and the current σ_t. I' will only be utilised in the final Step 3 (Section 3.2.3).

Equation 3.2 obtains the part of the radiant energy that *was not* absorbed (first factor) but *was* scattered (second factor) on its way to the camera. This is the energy that will be blurred in Step 2, therefore will show colour shifts caused by the medium absorption, e. g., in water will turn increasingly blue for more distant pixels. Again this effect can be seen in Figure 3.2, Step 2. The consequences of decoupling absorption and scattering in this way are discussed later, in Section 7.1.1.

Finally, Equation 3.3 projects the pixels' distances into what we call the *spread space*. The motivation for this is to provide a guidance for our anisotropic filter so that it more closely matches the scattering behaviour. This will be detailed in the following section.

3.2.2 Filtering

This section introduces the screen-space reference solution, as well as two approximations to it based on pyramidal filtering: a naïve (isotropic) and the proposed (anisotropic) approximations. This is the method's core phase (see also Figure 3.2, Step 2).

Screen-space reference To compute scattering in screen space for every pixel location $\mathbf{x}$ in $I^{[0]}$, one could (quadratically) iterate over all other pixels $\mathbf{y}$ and evaluate how much the (spatially varying) PSF of $\mathbf{y}$ contributes to $\mathbf{x}$ according to $F[\mathbf{y}](\mathbf{y}-\mathbf{x})$. Even if the set of pixels to visit for every pixel can be limited to a smaller neighbourhood – which is generally not possible for HDR conditions or optically thicker media – the effort is substantial and can take up to several tens of seconds for a single image. We will denote this method the *screen-space reference* and compare it to SSS in Figure 3.4.

Naïve approximation If the PSF was spatially invariant, we could compute the convolution of the image with the PSF by building a simple Gaussian MIP map. Gaussian MIP maps can be constructed efficiently (cf. Section 2.8), and the convolution of the image and the PSF could be evaluated accurately by a single fetch from the corresponding MIP level, at each pixel.

Using this approach to apply a spatially variant PSF naturally bears some limitations. First, fetching a value for a pixel corresponding to its distance assumes that each pixel from its neighbourhood contributes to it with an identical PSF. In addition, the convolution kernel is not even a Gaussian anymore, but for sake of our approximation we assume that it can be reasonably approximated by one. We call this the *naïve* method and compare to it in Figure 3.3.

This approximation works reasonably well if the PSF does not change abruptly across the image. Unfortunately this is not the case for distance discontinuities on object boundaries. The most visible consequence of such variation is *illumination leaking* (Figure 3.3, a), which happens because bright image locations close to the camera are blurred into the higher MIP levels that correspond to more distant parts of the scene. When the radiance of a distant pixel is then read from the corresponding MIP level, it is influenced by this leaked radiance, although it should not be as the source highlight is closer to the camera. Regrettably, this issue is especially apparent for scenes where the effects in question are most desired: scenes with high dynamic range, such as the one shown.

Our approximation We remedy the above problem in a similar manner as done for depth-of-field

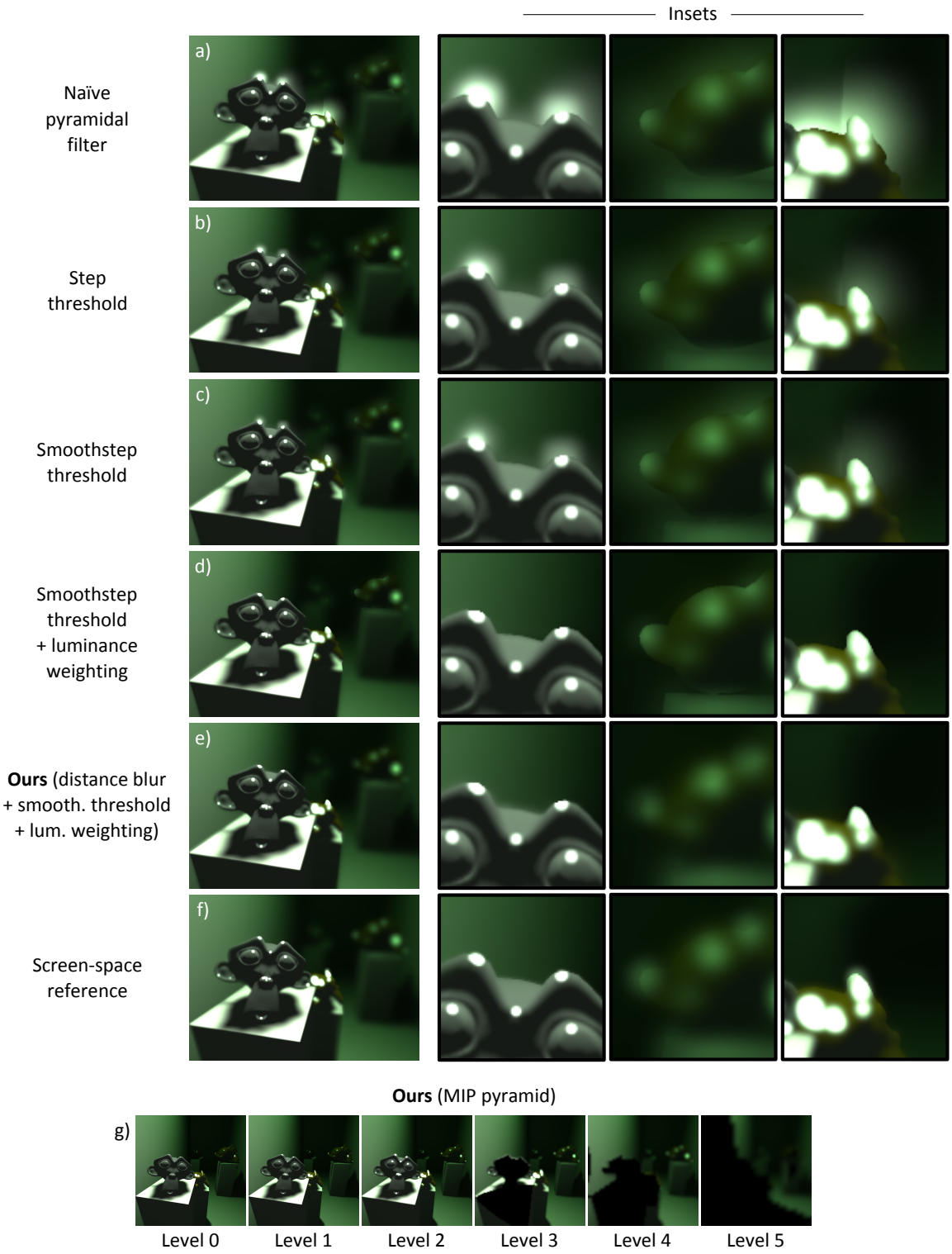


Figure 3.3: Analysis of SSS in various stages and compared with other, basic filtering approaches. Applying the standard Gaussian MIP map *(a)* leads to light leaking, which is slightly reduced by applying step *(b)* and the smoothstep *(c)* thresholding, but then significantly by weighting the spread-space distance by the pixels' luminance during the averaging *(d)*. The discontinuities that still remain are successfully removed by blurring the distance map *(e)*. The screen-space reference *(f)* resulting from a spatially variant Gaussian PSF computed by gathering from a 100×100-neighbourhood is three orders of magnitude slower (2.1 s versus 2.7 ms for the proposed solution). For illustration, *(g)* shows the scattering MIP map used to produce *(e)*.

rendering by Lee et al. [2009]. To prevent leaking of illumination from 'closer' MIP levels to higher, more 'distant' levels during the MIP map construction we need to know the distance of the target pixels and incorporate it into the filtering process. The resulting *anisotropic* filter then masks out such leaking pixels as the higher MIP levels are constructed. For this the MIP pyramid $\mathsf{W}^{[0..K]}$ is built, containing the distance projected to spread space, which can be used in the same manner as the distance itself thanks to the direct dependence between them (see Figure 2.9). The spread-space distance performs better for the intended purpose, because it allows to directly relate the MIP levels with the distance-dependent masking thresholds, as detailed shortly.

The construction of the MIP pyramid $\mathsf{I}^{[1..K]}$ is described first. To obtain $\mathsf{I}^{[k]}$ (where $k = 1..K$) we need to mask out those pixels of $\mathsf{I}^{[k-1]}$ during the filtering that correspond to smaller distances in the scene, in order to avoid the leaking of illumination:

$$\mathsf{I}^{[k]} = (\mathsf{M}^{[k]} \cdot \mathsf{I}^{[k-1]}) * \mathsf{G}', \tag{3.4}$$

where G' is a discrete filter approximating Gaussian; the representative implementation of SSS uses a 4×4 filter with weights $\{0.13, 0.37, 0.37, 0.13\}$ for each dimension. As shown by Burt [1981], for the given size this is the filter that leads to the smallest deviation from the exact Gaussian distribution. The auxiliary mask M at level k can be defined as

$$\mathsf{M}^{[k]} = \text{smoothstep}(t, (1+\epsilon)\cdot t, \mathsf{W}^{[k-1]}). \tag{3.5}$$

The definition uses the standard smoothstep function to perform the masking smoothly. The parameters t and ϵ control the masking threshold distance and width. Burt demonstrates that, when a discrete hierarchical convolution of this kind is performed, the width of the corresponding result filter doubles with every level of the hierarchy, and therefore the threshold that separates these levels must double in size as well. Because of this we define $t = c \cdot 2^{k-1}$, where c is a scaling constant described in Section 3.2.3.

In lockstep with $\mathsf{I}^{[1..K]}$ the MIP chain $\mathsf{W}^{[1..K]}$ is built. The MIP map $\mathsf{W}^{[0..K]}$ holds the average *spread-space distances* for whole groups of $2^k\times2^k$ (original) pixels at the corresponding levels k, as they are needed for computation of $\mathsf{I}^{[1..K]}$ (Equation 3.5):

$$\mathsf{W}^{[k]} = \frac{(\mathsf{Y}^{[k]} \cdot \mathsf{W}^{[k-1]}) * \mathsf{U}}{\mathsf{Y}^{[k]} * \mathsf{U}}, \tag{3.6}$$

where U is the uniform distribution of the same size as G'. We further define the auxiliary mask Y at level k as

$$\mathsf{Y}^{[k]} = y(\mathsf{I}^{[k-1]}). \tag{3.7}$$

The average is obtained by weighting the distance of each pixel by its absolute RGB luminance y, as brighter pixels ought to contribute more to the average distance that will be used as a masking criterion for obtaining the next MIP level (since *their* energy leakage we need to avoid primarily). In other words, the brighter an area is the more precise will our approximation at that position be.

We mathematically describe the averaging process in Equation 3.6 as weighting W by Y and then performing the averaging as a convolution with the box filter U. The uniform distribution is used instead of a Gaussian in this case, because for the average spread-space distance there is no reason to favour the central samples – we only need to take into account the pixels which can potentially cause the leaking.

The effectiveness of our formulation in reducing the light leaking is shown in Figure 3.3, b–d and the overall comparison to the screen-space reference is provided in Figure 3.4.

3.2.3 Final Compositing

Having the filtered MIP map that represents the scattering computed, obtaining J essentially amounts to reading each pixel's corresponding distance-dependent MIP level from $\mathsf{I}^{[0..K]}$ and adding it to I'.

The mapping from a pixel's distance to its MIP level needs to take into account that both the corresponding filter size and the masking threshold t grow exponentially with the increasing MIP level (Section 3.2.2). Therefore, the corresponding MIP level will grow logarithmically with spread-space distance as

$$\ell(\mathsf{D}) = \operatorname{clamp}\left(\log_2 \frac{W(\mathsf{D})}{c}, 0, K\right) \tag{3.8}$$

The scaling constant c determines the actual width of the corresponding filter in the MIP chain. Burt [1981] uses $c = 0.56$ for a 5×5 incremental filter, we however found that the value of 0.8 matches the screen-space reference solution much better in the case of the 4×4 filter used here. And, since the MIP levels must closely correspond to the masking threshold applied to them, c is also included in the definition of t in Section 3.2.2.

The issue with the direct application of Equation 3.8 to obtain the appropriate MIP level is the emergence of discontinuities in the resulting image, as shown in Figure 3.3, a–d. This problem is inherent to pyramidal convolution methods. The common workaround to this problem first introduced in Bertalmio et al. [2004] is to create a uniformly blurred version of the distance map and use this to determine the appropriate MIP level instead. This creates smooth transitions around depth discontinuities and therefore also gradual increase of the blurring level in these places.

The simplest way to perform this is applying an isotropic Gaussian pyramidal blur to D. This is very fast as the isotropic Gaussian filter is separable and D is a single-channel image. However the main flaw of simply using an uniformly blurred distance map is that the areas closer to the camera are blurred just as much as distant areas. A logical solution is based on the intuitive observation that the size of the blurring transition at distance discontinuities should be proportional to the blurring due to the scattering itself. This translates to using the distance of a pixel to first determine the strength of blurring the distance map, and then use *this* blurred distance value to calculate the query to the scattering MIP map and obtain the final image, J:

$$\mathsf{L} = \ell(\mathsf{D}^{[\ell(\mathsf{D}^{[0]})]}) \tag{3.9}$$

$$\mathsf{J} = \mathsf{I}' + \mathsf{I}^{[\mathsf{L}]}. \tag{3.10}$$

This solution is similar to Lee et al. [2009], but simpler. It is made possible by the fact that the blurring caused by light scattering increases monotonically (albeit non-linearly) with the distance from camera, while in depth-of-field simulation it increases in both directions with increasing distance from the focal plane. Therefore, instead of building the specialised anisotropic MIP map (described by Lee et al.) SSS builds a simple isotropic MIP pyramid, which can further be utilised by other screen-space methods, may the need arise.

3.2.4 Implementation

This section discusses additional performance considerations about the algorithm described above. First, it should be noted that it is usually not always necessary to build the full MIP chain of I (i. e., down to the 1-pixel level). If the maximal distance in the scene is known, then the value of the

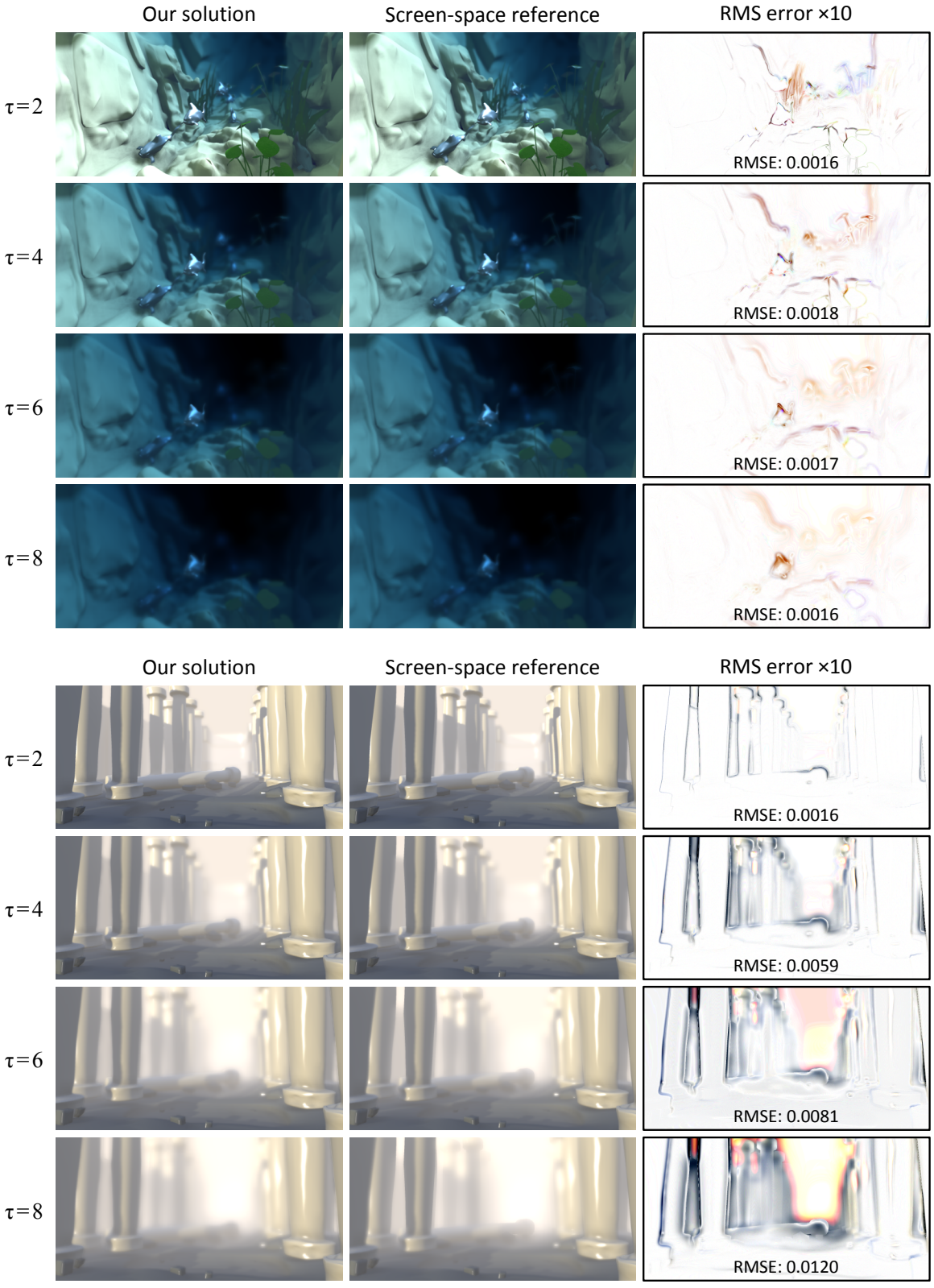


Figure 3.4: Evaluation of SSS for different values of medium optical thickness τ in a highly absorbing underwater environment *(top)* and a high-albedo fog *(bottom)*. Numerical evaluation against the screen-space reference is included to gauge the performance of our filtering solution under different optical conditions.

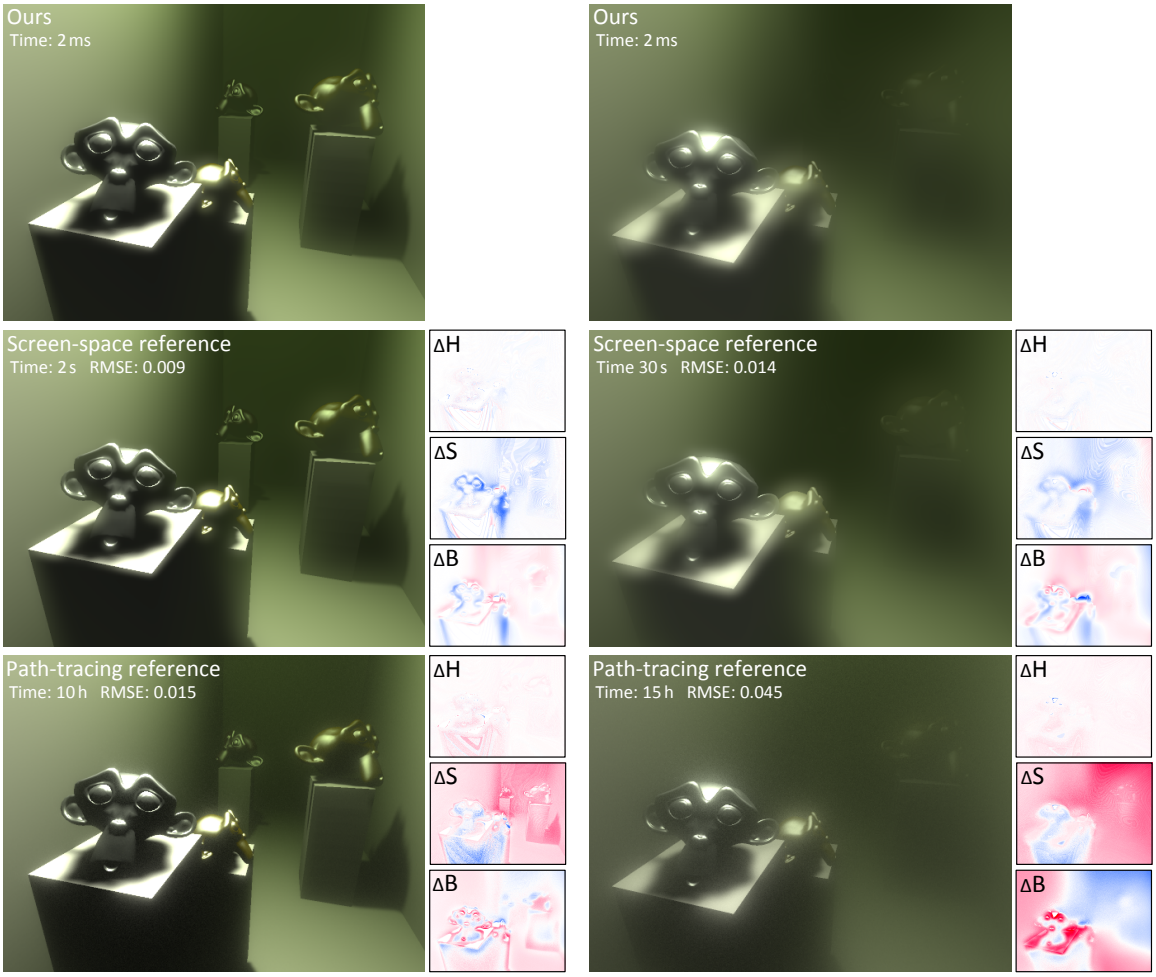


Figure 3.5: Comparison to screen-space and path-tracing references for two different scattering intensities. We measured the average RMS error for RGB intensities and also per-pixel differences in HSB space. The peak error that the HSB difference images represent is 20 %.

function W can be bounded with this distance and the used medium parameters. With the bound of W we can then determine the highest necessary MIP level K in the same way as ℓ is obtained in Equation 3.8. In our implementation $K = 6$ has proven to be sufficient in all cases.

Additionally, as an optimisation the spread space distances W can be packed into the alpha channel of the radiance texture when constructing the MIP map of I. Only one MIP map is therefore constructed, just its RGB channels are computed differently than its alpha channel.

Finally, to read $\mathrm{I}^{[0..K]}$ in Equation 3.10 a linear-bi-cubic fetching operation is used to obtain as smooth result as possible. This means that to obtain a value at level ℓ (which is generally fractional as defined by Equation 3.8), the two adjacent levels $\lfloor \ell \rfloor$ and $\lfloor \ell \rfloor + 1$ are read, each with a bi-cubic fetch, and then linearly combined to obtain the result. This would however require reading 32 samples in total, leading to an excessive cost of this step. We therefore modify the bi-cubic fetch in the way that groups of 4 pixels are fetched at the same time by a single bi-linear fetch from a position obtained by relating the respective pixels' weights given by the bi-cubic filtering function [Sigg and Hadwiger 2005]. This decreases the number of performed fetches to 8 bi-linear ones, which on

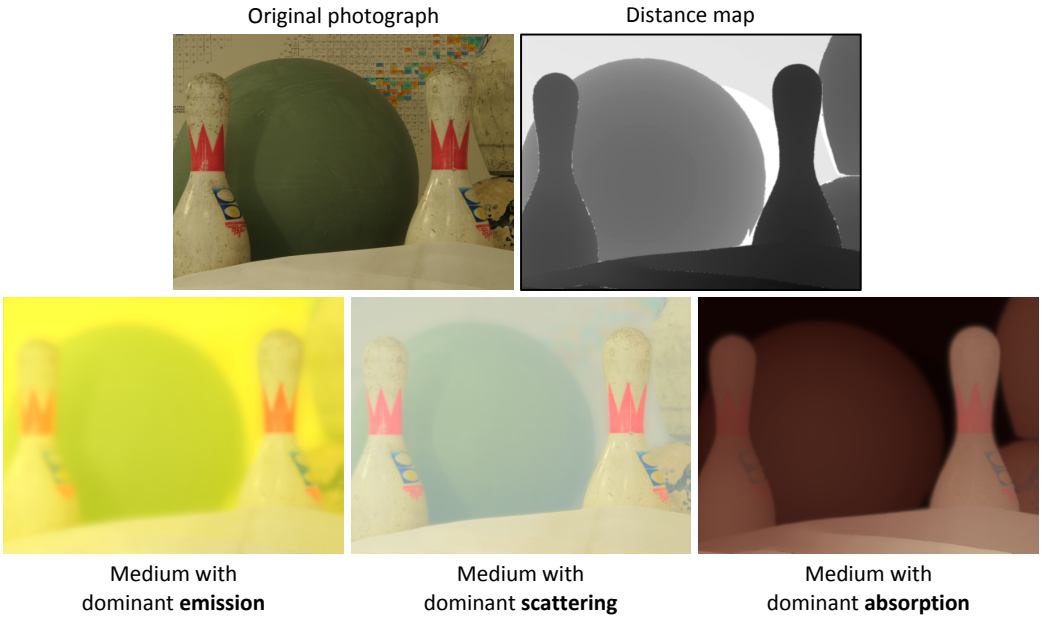


Figure 3.6: Application of SSS to a photograph with an automatically segmented distance map using three different media.

GPUs are virtually as fast as point fetches. We have also experimented with the circular filtering described in Lee et al. [2009], but found it to perform considerably worse in our context, in spite of the same number of fetches it requires. This might be a consequence of the different PSFs that the two simulated phenomena yield.

3.3 Results

The described algorithm has been implemented within an interactive rendering framework written in C++ and GLSL, using OpenGL. The GPU used in all tests and examples was NVidia GeForce GTX 485 Mobile with 2 GB video memory.

Figure 3.1, Figure 3.4 and Figure 3.5 show typical results of thee proposed approach (please refer to the captions for the corresponding settings and details). Figure 3.6 shows the application of the method to an RGBZ photograph. Such images are likely to become increasingly popular in the future, with depth sensors such as Microsoft Kinect commonly available to the mass market.

A quantitative evaluation is shown in Figure 3.5. In addition to the screen-space reference we also compare to a path-tracing solution. Please note that path tracing was applied just to the scattered portion of the illumination, instead of the full global illumination. The reason for this is to eliminate differences caused by the traditional approximations to global illumination on surfaces that we utilise, such as soft shadow maps or ambient occlusion. It can be seen that despite numerous approximations SSS builds on (cf. Sections 3.4 and 7.1.1), it is still able to obtain qualitatively comparable results even for very dense media.

The performance of the technique is detailed in Table 3.1; as can be seen it is always in hundreds of frames per second, making it a good candidate, e. g., for a post-processing step integrated in a real-time graphics engine. Probably the most important observation is that – in agreement with the prediction – the computation time scales linearly with the number of rendered pixels, as opposed to

Table 3.1: Performance of SSS for different image resolutions. Each step is measured individually, Step 2 for different filter types. The total time is the sum of Steps 1–3 with the 4×4 kernel used in Step 2.

I / D Resolution	Step 1	Step 2				Step 3	Total
		Ours			Naïve		
		2×2	4×4	6×6	4×4		
1 MPx	0.3 ms	0.7 ms	1.1 ms	2.6 ms	1.0 ms	1.2 ms	2.6 ms
2 MPx	0.6 ms	1.2 ms	1.6 ms	4.6 ms	1.5 ms	2.1 ms	4.3 ms
4 MPx	1.2 ms	1.9 ms	2.6 ms	8.6 ms	2.3 ms	4.2 ms	8.0 ms

the quadratic growth in the screen-space reference solution. In addition, the measurements show that using the anisotropic filter incurs a negligible additional cost in comparison to the naïve solution, despite increasing the quality considerably. Finally, performance for different filtering kernel sizes has been evaluated; please refer to Section 3.4 for a discussion in this regard.

3.4 Discussion

The described approach is an addition to the increasing number of screen-space methods, the history, strengths and weaknesses of which are summarised in Section 2.8. An extended discussion is provided in Section 7.1.1.

Depth-of-field rendering As already mentioned, SSS closely relates to Lee et al. [2009], and some of the involved steps might be regarded as adaptations from this method. Further, similarly to depth-of-field methods, an explicit bilateral filtering could be used to limit the illumination leakage even more (although our anisotropic filter in fact acts comparably to a bilateral filter). Finally, an interesting possibility to examine is using the scattering MIP map to render depth-of-field effects as well. This should be well feasible, and the main step towards such modification would be to combine the degrees of blurring due to scattering and de-focus and use the result to query the filtering hierarchy.

Filtering kernel size Because the resolution of the MIP map levels halves with each additional level, the size of the filtering kernel must be even. Taking into account the results of Burt [1981], we found that the filter size which provides the best compromise between price and quality is 4×4. While it is possible to use a 2×2 filter (with an approximately 30 % performance gain in the filtering step, cf. Table 3.1), its application does not correspond to a Gaussian kernel anymore, as it is equivalent to a simple box filter. Its limited spatial support also causes disturbing temporal flickering artefacts, especially in scenes with a high dynamic range. On the other hand, using a 6×6 kernel increases the cost of the filtering step considerably, roughly in proportion to the increase of fetches required to compute it (i. e., more than twice). Though slightly reducing the temporal flickering, all experiments indicate that it does not compensate for the increased computational costs.

Parametrisation The method also requires the specification of one arbitrary parameter, in addition to the physical properties of the simulated medium. This is the threshold width ϵ (Section 3.2.2). However, it is an intuitive quantity and once configured for a given scene it can typically remain unchanged.

Chapter 4

Amortised Photon Mapping

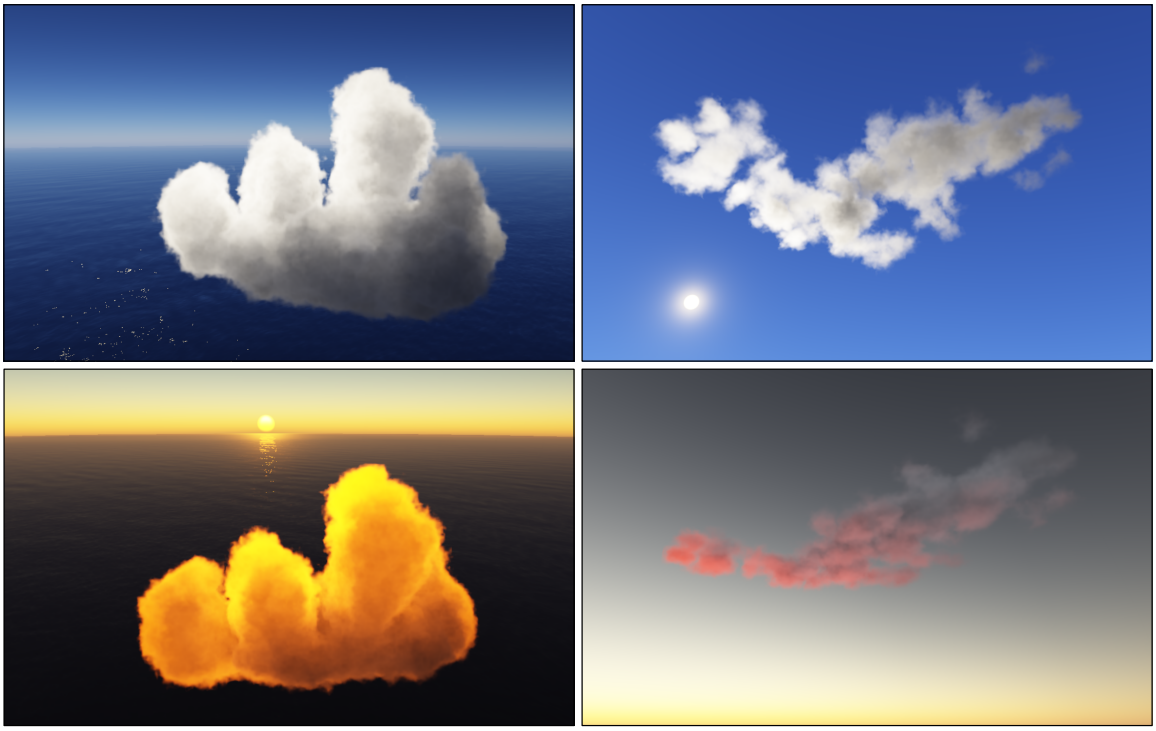


Figure 4.1: Cumulus congestus and Cirrocumulus altocumulus clouds rendered fully dynamically by the proposed approach at 50 and 90 FPS respectively. Shown in two different times of day.

4.1 Introduction

In contrast to the previous chapter, the focus of the following method is on efficient physically-based rendering of clouds (see Figure 4.1 for sample results). Interactive cloud rendering is often needed when an application includes interactive visualisation of outdoor environments. That includes serious applications such as flight and soaring simulators or meteorological visualisations, but also recreational ones, for instance 3D games and packages like Google Earth.

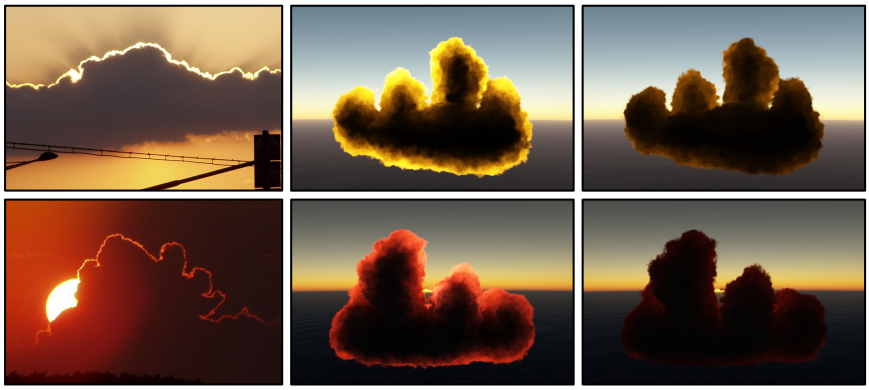


Figure 4.2: Anisotropic scattering is a defining visual property of clouds, causing above all the *silver lining* effect. Without correctly handling anisotropic scattering it is difficult to reproduce this phenomenon.

Even among participating media, clouds stand out as a particularly difficult case for state-of-the-art volumetric light transport techniques. This can clearly be seen from the volume of work related to cloud rendering, part of which is reviewed in Section 2.5.4. Their virtually unit albedo and small relative mean free photon path imply the necessity to simulate many orders of multiple scattering. The very high scattering anisotropy of cloud droplets (often in the order of $g = 0.95$, see Figure 4.2 for a comparison) in turn causes difficulties for stochastic methods (Section 2.5.1) and approaches based on discrete light propagation volumes (Section 2.5.2). The latter property in combination with the usual high complexity of cloud shapes also makes analytic approaches unsuitable (Section 2.5.3), mainly because it is usually difficult to fulfil the boundary conditions of the employed approximations. Finally, the high variability of morphological cloud types makes it difficult to establish a unified representation of their density distributions. Naturally, these issues are even more challenging for interactive rendering techniques.

The method presented in this chapter deals with these issues to a good extent by building on the following design decisions. Consequently, it is able to reach real-time rendering speeds on current consumer GPUs.

- Light energy is propagated via photon beam tracing [Jarosz et al. 2011a], which ensures a correct handling of scattering anisotropy. The basic similarity theory [Wyman et al. 1989], guided by a physically meaningful heuristic, is employed to avoid excessive tracing computation.
- To greatly decrease photon storage costs and accelerate the reconstruction of scattered radiance, the photon energy is accumulated in a compact grid structure with a fixed memory footprint. This structure is aligned with the cloud density field, but otherwise no coupling between them is maintained, which importantly enables dynamic changes to the density field (i. e., cloud shape).
- Instead of using a general spherical basis to represent the stored radiance directionality, we take advantage of the characteristic illumination in this environment (Section 4.2) and represent the radiance by simple unimodal distributions, namely the Henyey-Greenstein function (Section 2.3.3). This implies a very compact storage (usually using only a few lobes aligned with the respective reference illumination directions), but also an efficient radiance reconstruction.
- Using multiple grids to cache the photon energy, the tracing costs can be amortised across many frames (bounded only by available memory). This is feasible since cloud dynamics are typically much slower than the simulation update rate.

- Finally, to improve the apparent resolution of the solution, an enhanced version of the joint bilateral upsampling [Kopf et al. 2007] is used. This process relies on additional statistics calculated during the photon propagation. Upsampling is a very fast parallel operation which however significantly reduces the volume of photons that needs to be traced and stored.

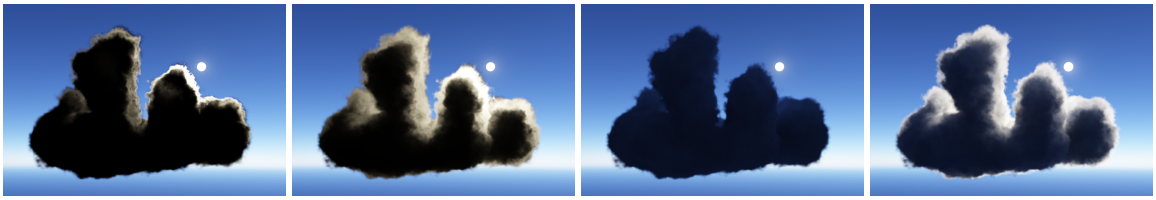


Figure 4.3: Decomposition of the three basic scattering categories in clouds. Note that the transition from single to second and higher orders of scattering is very fluent due to the high scattering anisotropy. For this reason methods that handle single and multiple scattering separately typically produce inconsistent results.

4.2 Amortised Photon Mapping

This section describes APM, first stating the method's assumptions and high-level structure, followed by a detailed exposition of each individual step (Sections 4.2.1–4.2.5). While the target platform are contemporary GPUs, the description is valid for any general fine-grained parallel machine. Such a machine executes many parallel threads, can coherently read from and write into buffers, read from special buffers ('textures') that provide efficient one-, two- and three-dimensional linear filtering, and can read and write a fast 'local' memory.

Assumptions First we need to establish some assumptions about the simulated environment, based primarily on actual observations. It is explained later how these are used to the method's advantage.

- **Illumination** Clouds are virtually always illuminated by a single strong yet slowly-moving light source with one *dominant direction* (i. e., the sun or the moon) and by additional slowly-changing environmental light sources (the sky, street lights in urbanised areas, etc.). Occurrences such as an airplane rapidly flying through a cloud are sufficiently rare and deemed acceptable to be neglected. In other words, clouds are illuminated by low-frequency light sources, and that both in temporal and spatial sense.
- **Density distribution** Clouds are indeed dynamic media, as their shape changes due to air convection, precipitation, etc. However, not only is this process never very rapid, it is seldom even noticeable for a human observer. Movement of whole clouds across the sky due to wind is usually relatively slow as well, and does not influence the shape in a significant way.

Apart from these, no other assumptions are made by the method. Multiple light sources can be handled, as long as there is one with a dominant radiant power and direction. Even local light sources are acceptable, as long as they move slowly. Similarly, all cloud shapes that can be reasonably represented by a 3D discrete density field are allowed. The source of this density field can be arbitrary – it can be stored as a series of volumetric animation frames in memory, or can result from a dynamic fluid convection simulation. Importantly, no precomputations need to be performed on these data. Finally, APM does not impose any limitations on the observer position or movement.

Overview The input to APM is a scalar density field of the simulated cloud $\mathsf{D}(i,j,k) : \mathbb{N}^3 \to \mathbb{R}$ (see

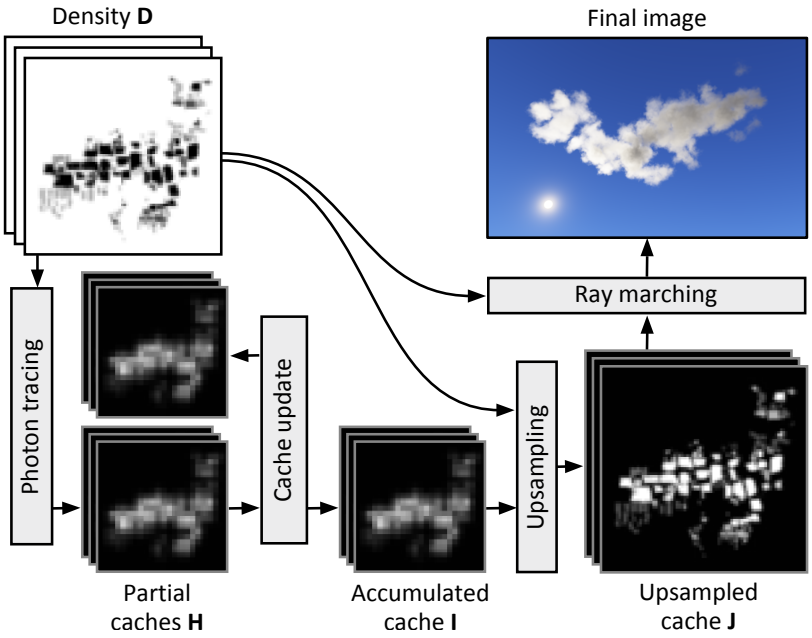


Figure 4.4: Schematic view of the algorithm's pipeline.

Figure 4.4). At the beginning of each frame D is updated (e. g., using a cloud dynamics simulation or by streaming the dataset from memory on-the-fly) and stored in a regular 3D texture.

Assuming a slowly changing environment, the illumination is cached in a separate (regular) grid $\mathsf{I}(i,j,k) : \mathbb{N}^3 \to \mathbb{R}^4$. Every cell in I stores the RGB flux as well as one anisotropy coefficient g (Equation 2.11) to model the light distribution (Section 4.2.2). The grid I is updated progressively, so that each cell stores a mix of the current and several increasingly outdated values – however, as long as the above assumptions hold, the difference from the ideal solution will be very small (as demonstrated in Section 4.3).

The back-end of the caching data structure is a circular buffer of m (we typically use $m = 100$) partial caches $\mathsf{H}_0(i,j,k), \ldots, \mathsf{H}_{m-1}(i,j,k)$ that combine as

$$\mathsf{I}(i,j,k) = \frac{1}{m} \sum_{l=0}^{m-1} \mathsf{H}_l(i,j,k) \tag{4.1}$$

to the full cache. The spatial resolution of the cache I is assumed to be much lower than the one of the density field D to keep the amount of consumed memory and the variance of I low. To improve quality, before reconstructing the final image from I and D using ray-marching, the solution I is upsampled to a second cache J, which has the same resolution as D.

The global energy state of the cloud is represented by n_t photons, which are uniformly divided into m generations, each containing $n_\mathrm{g} = n_\mathrm{t}/m$ photons. In each frame f the algorithm performs the following steps:

1. The flux of n_g photons that are shot and traced through D (Section 4.2.1) is stored into a *new* partial cache H_new (Section 4.2.2).
2. The partial cache $\mathsf{H}_\mathrm{old} = \mathsf{H}_{f \bmod m}$ in the circular buffer becomes the old partial cache and we replace it with the a partial cache H_new as: $\mathsf{I}^{(f)} = \mathsf{I}^{(f-1)} - \mathsf{H}_\mathrm{old} + \mathsf{H}_\mathrm{new}$ (Section 4.2.3).
3. The low-resolution cache I is upsampled to another cache J using the density field D as a guiding

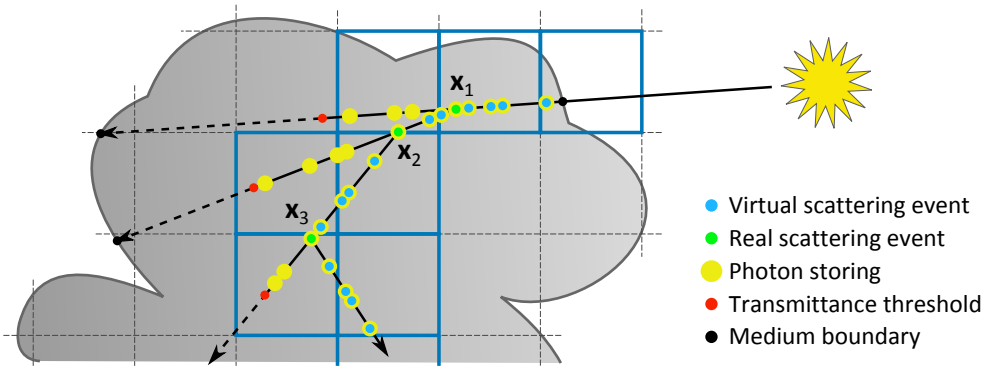


Figure 4.5: Scheme of the photon beam propagation procedure. Only the blue-framed grid cells receive photon energy.

signal (Section 4.2.4).

4. Ray-marching D and J together produces the final image (Section 4.2.5).

4.2.1 Light Propagation

Illumination is computed by tracing photon beams in cloud the density field D and storing them into a new partial cache H_{new} (Section 4.2.2). The spatial resolution of H is typically much lower than of D: as an example, in the presented implementation the number of cells in each partial cache should not exceed 2400 (for a cubic grid this limits the resolution to 13^3) in order to fit into the 48 kB of shared local memory available on tested GPU (see Section 4.3 for more details).

Photon tracing In every step, n_g photons are shot from the sun and the sky. Note that different numbers of photons may be traced in each frame, e. g., if there is a need to balance the computation time against other tasks performed by the GPU; the only requirement is that these photons carry the same flux, so that the energy state of the scene remains consistent. Each photon is firstly attenuated by atmospheric scattering using a tabulated model [Bruneton and Neyret 2008], [Elek and Kmoch 2010].

For the propagation, we adapt the *photon marching* technique presented by Jarosz et al. [2011a]: for every photon a parallel thread is started which runs in a while-loop that scatters and stores photons (Section 4.2.2) until each individually leaves the volume. Instead of depositing photons in constant [Jarosz et al. 2011a] or adaptively-sized [Krüger and Westermann 2003] steps, APM performs randomly-sized steps. This is because Woodcock tracking (that itself performs randomly-spaced steps to generate interaction events, Section 2.4.1) is employed for obtaining free photon paths through the medium. Consequently, photons are deposited at the locations where Woodcock tracking examines the currently generated event. This essentially helps to avoid two simultaneous stepping procedures along the propagation ray. See Figure 4.5 for an illustration of this process.

The downside of such an approach is of course the potentially higher variance of the solution, since the photon beams are discretised into points instead of being treated as full 1D primitives. However, as the photon contributions are aggregated in the grid cells (which are much larger than the average mean free path in the medium) this effect turns out to be small.

In addition, the original photon marching technique traces each photon beam up to the medium boundary. Such an approach is suitable for optically thin media such as fog and certain kinds of smoke which Jarosz et al. [2011a] typically use in their experiments. However, in optically

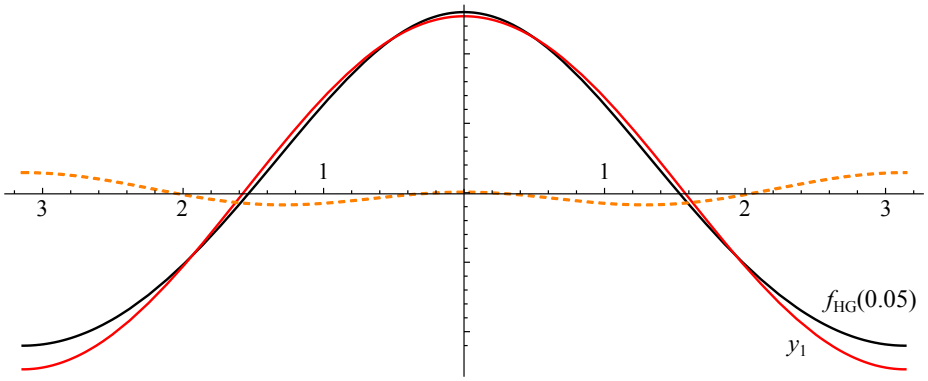


Figure 4.6: Comparison between f_{HG} with $g = 0.05$ and the first order spherical harmonic function y_1. The dashed orange plot represents the difference between these two functions.

thick environments such as clouds, this approach invests large amounts of computational effort into simulating and storing photons that—due to low transmittance—carry only very little energy. To overcome this problem APM simply stops tracing a beam if its transmittance gets below a small threshold. Although this of course introduces a slight additional bias, the resulting speed-up can reach and order of magnitude.

Further, we use the similarity theory [Wyman et al. 1989] to speed up the tracing process. A fixed threshold $t = 0.05$ is posed, and if the cumulative scattering anisotropy of a traced photon gets below t, the algorithm switches to the *reduced scattering coefficient* $\sigma_s' = \sigma_s \cdot (1 - g)$ and isotropic phase function. This can be determined heuristically, because the angular distribution of light scattered i times roughly corresponds to i self-convolutions of the phase function, which for the Henyey-Greenstein function in turn corresponds to just using g^i instead of g as its anisotropy parameter (see Section 2.3.3 for more details). Therefore for a given photon we switch to σ_s' and isotropic scattering after $i = \log(t)/\log(g)$ bounces, where g has to be positive.

The choice of t is not arbitrary; according to the derivation of the original similarity theory [Wyman et al. 1989] the equivalence relations are valid if the radiance anisotropy is linear, i. e., if it can be expressed by spherical harmonic functions with the maximum degree of one. Except for the case of $g = 0$ this is unfortunately not possible for f_{HG}. We can therefore aim at least for a difference within some error threshold (Figure 4.6). The traditional threshold of a 2 %-error corresponds roughly to $g = 0.1$; however, a lower error margin needs to be reached as the error accumulates with each scattering bounce. Using $t = 0.05$ ensures an approximately five times lower error margin, which is arguably sufficient for this application.

4.2.2 Photon Storing

The classic caching-based volumetric rendering methods, such photon mapping [Jensen and Christensen 1998] and irradiance caching [Jarosz et al. 2008a] (Section 2.5.1) have shortcomings that hinder their application to interactive cloud rendering: maintaining and querying of complex spatial data structures and a limited support for highly-anisotropic scattering due to spherical harmonic functions used for the caching. APM overcomes both these limitations by using a simple regular grid and a different, simpler directional representation.

Regular photon grids Classic volumetric photon mapping stores photons into hierarchical data-

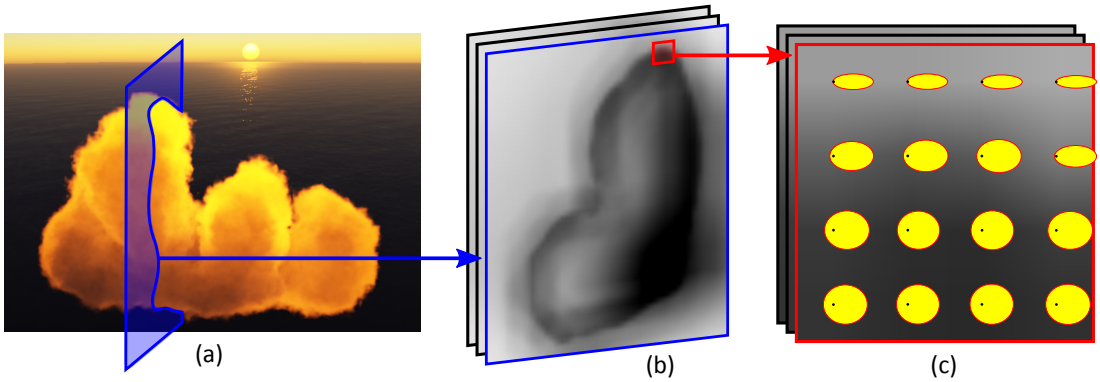


Figure 4.7: Local Henyey-Greenstein lobes – *(b)* shows a slice of the greyscale-encoded anisotropy coefficients of the illumination solution that correspond to the marked region in the cloud *(a)*. The exact shape of the lobes in the cache locations is shown in *(c)*. The light in *(c)* is coming from the left.

structures—such as kD-trees—that can adaptively resolve fine spatial details in the photon distribution as found, e. g., in occlusions or caustics in surface lighting. For clouds however, fine spatial details are usually induced by gradients in the density field rather than radiance itself. From another perspective, inserting photons into a complex structure and performing adaptive density estimation is less suitable for the targeted massively-parallel processors.

We therefore suggest to trade adaptivity for simplicity and revert to a plain regular grid in which the illumination is accumulated, similar to irradiance volumes [Greger et al. 1998] (illustrated in Figure 4.5). This grid H is sufficiently small to fit into local memory shared by a block of threads executed on the same multi-processor. Storing a photon into such a three-dimensional grid is now as efficient as splatting it in two dimensions [McGuire and Luebke 2009]. Also, since atomic operations are necessary to perform the accumulation without access collisions, for the fast local shared memory we found them not to cause virtually any additional overhead.

The resolution of such cases is naturally limited, and with it also the illumination features that can be reproduced. Section 4.2.4 explains how a specifically tailored upsampling scheme can be used to improve the effective spatial resolution of the cache.

Henyey-Greenstein basis Approaches that cache the illumination information inside the volume [Jarosz et al. 2008a; Moon et al. 2008] typically use spherical harmonics to approximate the radiance directionality. This is suitable for isotropic or moderately anisotropic scattering media, but not for strongly forward-scattering media such as clouds, where a very high number of SH coefficients is required. This is problematic, as the strong forward scattering in clouds is visually important for their appearance, mainly causing the well known *silver lining* phenomenon (Figure 4.2).

To overcome this limitation we propose to represent caches using the Henyey-Greenstein function (Equation 2.13). Evidence shows [Ihrke et al. 2007] that in situations where most energy in scene comes from a single direction then also a majority of scattered light will intuitively propagate in a very similar direction, which will be especially true for media with high scattering anisotropy. Thus, when storing a photon into a cache cell, in addition to its radiant flux the algorithm also accumulates its cosine with respect to the dominant light direction. This only requires storing four values (an RGB-triple for flux and a single cosine value) per cell and the projection consists of evaluating just a single dot product per photon. In addition the number of photons arriving at each cell is counted, and after the tracing stage finishes, the accumulated cosines are divided by the number of contributing photons for a proper normalisation. The value obtained this way is exactly the anisotropy factor for

the given cache cell (Equations 2.14 and 2.15). The Henyey-Greenstein function can then be directly parametrised by this value to represent the per-cell angular illumination distribution (Figure 4.7).

Photons which are not emitted by the sun but arrive from the sky are not projected into the HG basis. As they—in difference from the sun photons—arrive from a wide distribution of directions, their angular contribution to the caches will likely be close to uniform. In case this does not hold (for instance if the sky is much brighter than the ground) it is easy to add an additional HG lobe and obtain the reference direction by weighting the sphere of directions by the relative energy contributions of the environmental lighting.

4.2.3 Cache Update

After resolving the illumination via photon tracing, the partial cache H_{new} is added to the global solution I and the outdated cache H_{old} is removed. This is done using a simple parallel addition and subtraction over all cache cells.

4.2.4 Solution Upsampling

At this point the cache I could already be used for rendering. However, as described above its spatial resolution is usually much lower than the one of D. This is required to fit it into shared local memory and to reduce variance of the illumination solution (see the discussion in Section 4.3), as well as the overall memory footprint. This, however, has two problems. First, the low resolution of I can cause a substantial blur of the stored light energy and cause interpolation artefacts. Second, as the relation between the anisotropy factor g and the resulting lobe shape is quite non-linear, linearly interpolating it during the cache fetching would sometimes produce incorrect angular distributions of the scattered radiance.

To avoid these limitations, we propose to upsample I to J, a cache with the same spatial resolution as D (Figure 4.8). One possible approach to this is to apply the joint bilateral upsampling [Kopf et al. 2007]. The idea of the joint bilateral upsampling is to use a bilateral filter consisting of two parts: a traditional domain filter, e. g., Gaussian, and a range filter operating above a *guiding signal*, which provides an additional regulatory mechanism over the filtering process. In our case it is very natural to use D as the guidance signal. This approach produces empirically good results, which is not surprising as it can be assumed that the density field D and the scattered radiance function L_{out} (Section 4.2.5) will spatially correlate on a local scale (roughly within one mean free path distance).

The main issue with the basic joint bilateral upsampling is that it employs Gaussian filters, as mentioned above. This requires specifying their parameters – standard deviations and, in case of their truncated versions, also effective radii. It is not necessarily clear what these values should be, and moreover, they can vary not only for different clouds, but also for a single cloud during its evolution.

To overcome this problem it is useful to exploit the additional knowledge available here, both in terms of the working data and the fact that the upsampled quantities have direct physical meanings. Based on this knowledge a more suitable physically-plausible upsampling filter can be designed.

At the end of the tracing process I contains two different quantities that need to be upsampled: the ‘concentration’ of radiant energy inside the cloud and its angular distribution represented by the HG anisotropy factor.

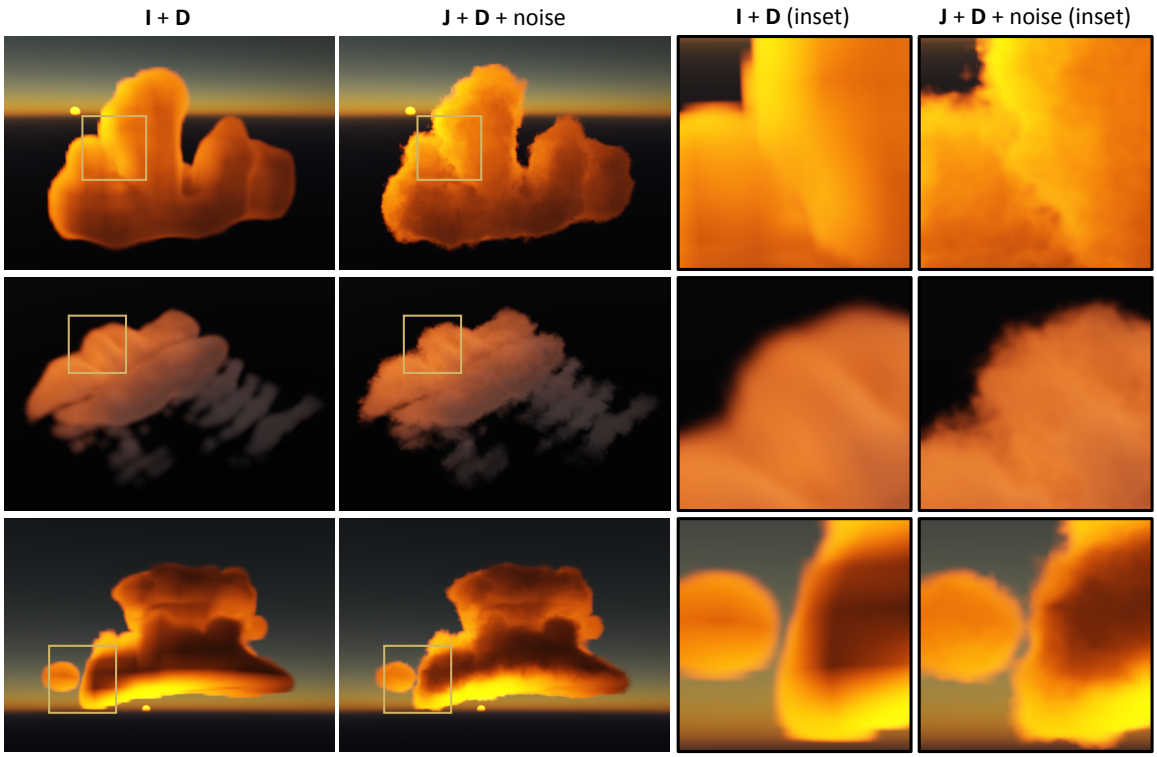


Figure 4.8: Upsampling of the coarse illumination solution I to J removes the artefacts caused by linear interpolation and incorrect handling of the radiance anisotropy. The additional noise perturbation helps to break the unnatural smoothness of the underlying density field D.

Radiant energy From the RTE we can see that the radiant energy L is locally proportional to the medium scattering coefficient, which in turn depends linearly on the density field D. Our proposition for its upsampling is using a joint bilateral filter f_L as follows. The domain filter needs to take into account that L is exponentially attenuated in space due to the medium transmittance. The range filter, which naturally uses D as the guiding signal, has to incorporate the linear dependence of the scattered L on the medium density. Mathematically this can be written as follows:

$$f_L(d_\mathrm{D}, d_\mathrm{R}, \tau, r) = f_\mathrm{texp}(d_\mathrm{D}, \tau, r) \cdot f_\mathrm{lin}(d_\mathrm{R}), \tag{4.2}$$

where d_D and d_R are the distances in the spatial and range domain respectively, τ is the optical thickness in the medium and r is the effective filtering radius. Please refer to the end of this section for the definition of these parameters.

The functions f_lin and f_texp are the linear and truncated exponential distributions respectively:

$$f_\mathrm{texp}(d,\lambda,r) = \max\left(\frac{f'_\mathrm{texp}(d,\lambda,r) - f'_\mathrm{texp}(r,\lambda,r)}{1 - r \cdot f'_\mathrm{texp}(r,\lambda,r)}, 0\right) \quad \text{with} \quad f'_\mathrm{texp}(d,\lambda,r) = \frac{\lambda \cdot e^{-\lambda d}}{1 - e^{-\lambda r}},$$
$$f_\mathrm{lin}(d) = \max(1-d, 0).$$

The truncated exponential distribution f_texp is defined by normalising the thresholded exponential distribution[1] f'_texp. The reason for this formulation is the infinite support of the exponential distribution (which should correctly be used). However, since only a local neighbourhood is used for filtering, using the exponential distribution would introduce discontinuities in the upsampled solution J.

[1] f'_texp parametrised by r is the basic exponential distribution modified to have a unit integral on the interval $[0, r]$.

Radiance anisotropy As already mentioned the upsampling of the anisotropy g' has to take into account the nonlinear relation between its value and the distribution it represents. There is however no straightforward way to compensate for this nonlinearity without an additional information about the medium.

Section 4.2.1 describes how the radiance anisotropy changes by scattering – when light progresses into the medium, its anisotropy decreases with the increasing number of scattering orders. Hence the anisotropy at a given position will likely well correlate with the average scattering order at that point. This quantity, dubbed *penetration depth* γ, can easily be obtained during the tracing process; we track the number of times a photon scatters, and store this value along with the photon's flux and average cosine. In contrast to anisotropy, γ already is close to linear, and thus can be linearly interpolated from the low-resolution grid during the upsampling. The upsampling filter for g' can then be defined as

$$f_{g'}(\delta, g) = g^{\delta}, \tag{4.3}$$

where δ is the difference of penetration depth γ between the upsampled point and the point from which we the anisotropy value is being extrapolated, and g is the medium scattering anisotropy parameter. The logic behind this filter is exactly the same as in the similarity heuristic in Section 4.2.1. Note that δ can also have a negative value, which means that if the upsampled point has a lower penetration depth than the extrapolated point, its anisotropy will be higher.

Final upsampling The upsampling procedure itself then iterates through a local neighbourhood Δ of the upsampled point $\mathbf{p}$, extrapolating the contributions from the neighbouring points to its radiant energy and anisotropy:

$$\mathsf{J}_L(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in\Delta} \mathsf{I}_L(\mathbf{q}) \cdot w_L}{\sum_{\mathbf{q}\in\Delta} w_L} \qquad \text{with} \qquad w_L = f_L(d_\mathrm{D}, d_\mathrm{R}, \tau, r), \tag{4.4}$$

$$\mathsf{J}_{g'}(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in\Delta} \mathsf{I}_{g'}(\mathbf{q}) \cdot w_{g'} \cdot w_L}{\sum_{\mathbf{q}\in\Delta} w_{g'} \cdot w_L} \qquad \text{with} \qquad w_{g'} = f_{g'}(\delta, g), \tag{4.5}$$

and $d_\mathrm{D} = \|\mathbf{p} - \mathbf{q}\|$, $d_\mathrm{R} = |\mathsf{D}(p) - \mathsf{D}(q)|$, $\delta = \mathsf{I}_\gamma(\mathbf{p}) - \mathsf{I}_\gamma(\mathbf{q})$, $\tau = \sigma_\mathrm{s} \cdot d_\mathrm{R} \,/\, 2$ and r is the shortest distance from the centre of Δ to its boundary.

It should be pointed out that the anisotropy weights $w_{g'}$ are multiplied by w_L as well. This is because the points in the cache which contribute more energy to the upsampled point should also have a higher overall impact on its resulting anisotropy.

The upsampling procedure maps well to parallel execution environments – it is a purely local operation executed exactly once per each upsampled solution cell. According to our experiments using a 3^3 neighbourhood already produces good results, although ideally it should be in the same order as the resolution ratio between D and I. The upsampling could also be performed directly during the final ray-marching stage (Section 4.2.5), but it is much faster when done in an intermediate step, as this allows a coherent reading of I and D and no work is duplicated.

4.2.5 Cloud Visualisation

To visualise clouds APM relies on the standard, front-to-back, early-exit ray marching [Krüger and Westermann 2003]. For every pixel a thread is started that linearly marches through D and J, accumulates radiance from J (see below) and transmittance T from D, applies compositing accordingly and terminates when T is less than 2 %.

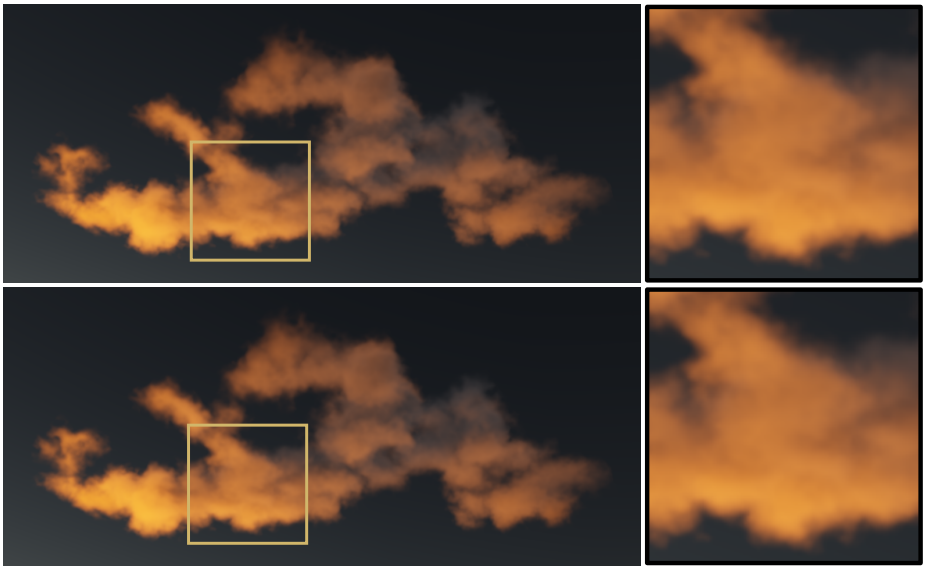


Figure 4.9: Cirrocumulus altocumulus rendered at 1920×1080 (HD) resolution (*top*, ray-marching time 77 ms) and half-HD resolution (*bottom*, ray-marching time 21 ms).

To keep the memory consumption in affordable levels only moderate resolutions for D are recommended (see Table 4.1). The amount of detail can then be increased by procedurally perturbing the texture coordinates for fetching D and J by several octaves of simple vector noise stored in a compact 3D texture, similarly to Kniss et al. [2002] (Figure 4.8). Typically we use three octaves, although this can be regulated depending on the cloud's projected size.

For most clouds it is not necessary to ray-march them in full screen resolution, as the frequency of the applied noise is still lower than the screen sampling frequency (see Figure 4.9). Therefore in all cases we opted to render the clouds in half of screen resolution.

In addition, because of the good temporal coherence of the environment, it is wasteful to conservatively ray-march the cloud volume in every frame. It is therefore beneficial to implement an impostor caching strategy to avoid this. Our proposition is to use several empirical thresholds, which, if any is exceeded, indicate that the cloud impostor should be updated:

1. Angular criterion – if the relative angle under which the cloud is seen changes by more than 5°.
2. Distance criterion – if the relative distance from the cloud changes by more than 20 %.
3. Illumination criterion – if the photon map has been updated by more than 4 % (with respect its update period m).
4. Animation criterion – if the cloud shape has changed significantly.

Naturally, all these criteria are dependent on the cloud type, evolution speed, camera movement pattern and so on, and therefore need to be tuned individually. As Section 4.3 details, the two above optimisations (resolution reduction and impostors) increase the ray-marching speed by about an order of magnitude, and that without causing any visible degradation in quality.

Reconstruction Aside from its compactness, there are additional benefits of the proposed Henyey-Greenstein representation. Not only is the illumination directionality represented well for the highly anisotropic scattering characteristic in clouds, but at the same time is the representation smooth. This prevents high-frequency noise that could possibly emerge if another basis was used, which in clouds

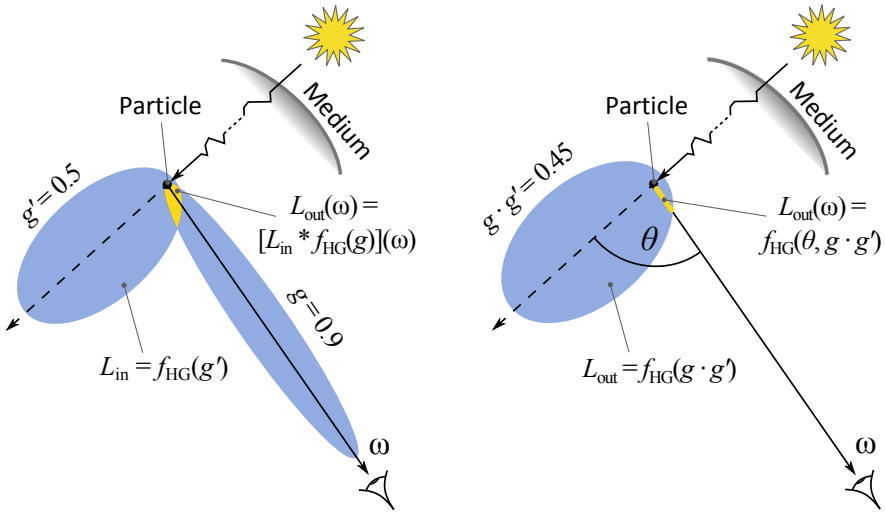


Figure 4.10: Two possible ways of reconstructing the illumination information represented by the Henyey-Greenstein function. *Left:* the yellow area represents the value of $L_{out}(\omega)$ obtained by convolving L_{in} and $f_{HG}(g)$. *Right:* the yellow segment is directly the value of $L_{out}(\omega)$.

would appear visually disturbing.

The illumination reconstruction itself can be done efficiently during the ray marching though J. Each cell of J contains RGB intensities of the directional and environmental illumination energy and the local directional anisotropy factor g', which represents the angular distribution of this energy.

It is however important to note that since APM uses the incident photon direction for the projection step during the photon tracing procedure (Section 4.2.1), the stored value g' represents the *incident* radiance function L_{in} at the cache location. To obtain the desired *outgoing* radiance value $L_{out}(\omega)$ for a given view direction ω it is necessary to convolve L_{in} (represented by $f_{HG}(g')$) with the medium phase function centred around the viewing direction (Figure 4.10, left). Instead of computing the convolution directly (e. g., numerically) we can once more utilise the self-convolution property of the Henyey-Greenstein function (Section 2.3.3), and simply evaluate $f_{HG}(\theta, g\cdot g')$, where g is the HG anisotropy coefficient used for the scattering simulation (Figure 4.10, right).

This is mathematically equivalent. Recalling the RTE (Equation 2.25), it can be seen that its scattering contribution factor (Equation 2.26) is a spherical convolution between the incident radiance (hereinafter L_{in}) and the medium phase function f at the scattering location $\mathbf{x}$.

APM approximates the incident photons' distribution L_{in} with the f_{HG} basis function by averaging their cosine values in respect to the dominant (sun) direction, thus producing an anisotropy factor g' representing the angular distribution of scattered photons at $\mathbf{x}$ (Section 4.2.2). Therefore $L_{in} \approx f_{HG}(g')$ and the integral term of Equation 2.26 becomes

$$\int_\Omega f(\omega', \omega)\cdot L_{in}(\omega')\,\mathrm{d}\mu(\omega') \approx \int_\Omega f(\omega', \omega)\cdot f_{HG}(\omega', g')\,\mathrm{d}\mu(\omega'),$$

omitting the spatial dependence from the notation. Note that the orientation of $f_{HG}(g')$ in space is given by the dominant light direction, which is constant for the entire cloud (although nothing prevents the orientation from varying in space by storing the direction separately for each cell).

Since in APM the used medium phase function is also f_{HG} parametrised by g, utilising the self-

Table 4.1: Performance of the algorithm across various stages. All other steps not listed in the table took less than 1 ms to execute.

Dataset	D resolution	D size (km)	σ_s (m^{-1})	g	I resolution
Altostratus undulatus	70×22×71	1.4×0.4×1.4	0.03	0.96	15×10×15
Cumulus congestus	48×62×90	0.9×1.2×1.8	0.03	0.96	10×12×18
PBRT smoke (anim.)	100×40×100	2×0.8×2	0.03	0.96	15×10×15

Dataset	Photon tracing	Upsampling	Ray-marching (conservative)	Ray-marching (impostors)	Total (cons. / imp.)
Altostratus undulatus	6.1 ms	2.3 ms	22.2 ms	7.4 ms	30.6 ms / 15.8 ms
Cumulus congestus	9.1 ms	3.0 ms	20.7 ms	7.9 ms	32.8 ms / 20.0 ms
PBRT smoke (anim.)	7.8 ms	3.3 ms	33.2 ms	9.5 ms	44.3 ms / 20.6 ms

convolution property of f_{HG} the above can be rewritten as

$$\left[\int_{\Omega} f_{\mathrm{HG}}(\omega', \omega, g) \cdot f_{\mathrm{HG}}(\omega', g')\, \mathrm{d}\mu(\omega')\right](\omega) = f_{\mathrm{HG}}(\omega, g \cdot g').$$

Thus the *outgoing* radiance function is directly represented by the HG distribution as $L_{\mathrm{out}} \approx f_{\mathrm{HG}}(g \cdot g')$, allowing an efficient closed-form evaluation during the ray marching.

4.3 Results and Discussion

The presented experiments have been conducted on a laptop PC with 2 GHz Intel Core-i7 CPU, NVidia GTX 485 Mobile GPU and Windows 7 64-bit. All tests use the following global settings: 819k photon beams, $m = 100$ partial caches, ray-marching step size of $1/300$-th of the cloud bounding box diagonal, and a screen resolution of 1920×1080. All density fields beside the PBRT smoke were modelled by hand as meshes and voxelised into 3D grids. We have performed three kinds of tests – method performance measurements, analysis of the illumination out-dating during scene state changes, and the rendering quality in dependence on the photon map resolution. An extended discussion is provided in Section 7.1.2.

Performance We have used three datasets for timing measurements: two static datasets of Cumulus congestus (dense concentrated cloud) and Altostratus undulatus (sparse wavy cloud) and the animated smoke dataset available in PBRT renderer (pbrt.org). The respective scene settings and measured timings are summarised in Table 4.1. Figures 4.1, 4.13 and 4.14 show several results of the method, including the measured datasets.

As Table 4.1 shows, the photon tracing costs are in the order of milliseconds, since only a low number of photon beams is traced per frame (about 8.2k in our tests). Under such conditions, the ray marching becomes the most expensive step, if performed conservatively in each frame. The impostor caching decreases the ray-marching costs by a factor of three, on average, which then also decreases the total rendering time significantly.

Illumination obsolescence Since APM progressively amortises the illumination computation, the partial solutions in the circular buffer are increasingly obsolete. To show that the update process is sufficiently fast we compare clouds rendered under constant light source movement against clouds under static illumination. Figure 4.11 shows the resulting images. The maximum luminance

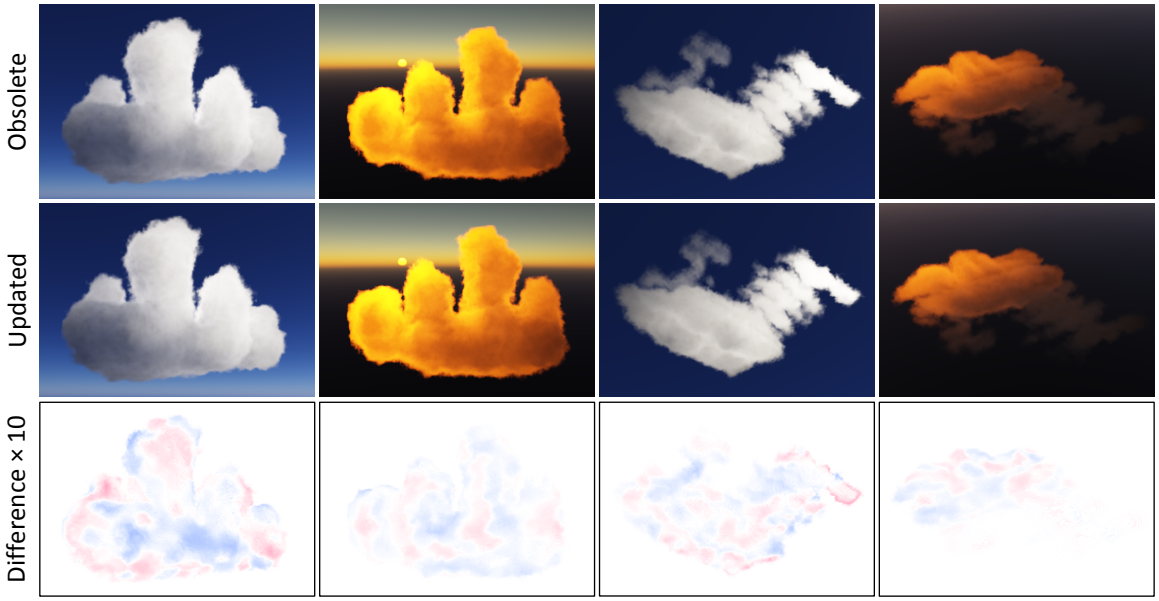


Figure 4.11: Illumination obsolescence in Cumulus congestus and Altostratus undulatus in two different times of day. The out-dated cases were captured during a continuous sun movement of 0.5 deg/s.

Figure 4.12: Quality comparison for different photon map resolutions, $m = 100$. *Left:* 270 cells (0.5 MB video memory), 102k photon beams, 4.8 ms per frame. *Middle:* 2080 cells (4.2 MB video memory), 819k photon beams, 9.1 ms per frame. *Right:* 16640 cells (33.8 MB video memory), 4096k photon beams, 33.4 ms per frame. The settings used to generate the middle image are used in all the other results for the particular dataset.

difference was about 1.5 % throughout all cases, mostly due to a residual low-frequency photon noise. Please note that although the angular speed of sun was only 0.5 deg / s, this is still 120 times faster than the real speed and several times faster than is commonly used in interactive 3D applications (for instance the common time ratio in games is 1 game-time hour to 2 real-time minutes, which corresponds to an angular speed of 0.125 deg / s).

Photon map resolution Figure 4.12 shows the Cumulus congestus dataset rendered with three different resolutions of the photon map: the resolution used in all other measurements, and two others, one with eight times less and one with eight times more cells. The photon tracing times naturally differ as well, because (as discussed below) different amounts of photon beams have to be used to obtain coherent results. It can however be seen that for lower numbers of photon beams the simulation time does not scale down linearly. This is because if too few beams are traced simultaneously, the GPU utilisation becomes too low. The conclusion is that although the quality degrades rather gracefully with decreasing photon map resolution (mainly thanks to the upsampling stage), from a certain point doing so does not pay off anymore.

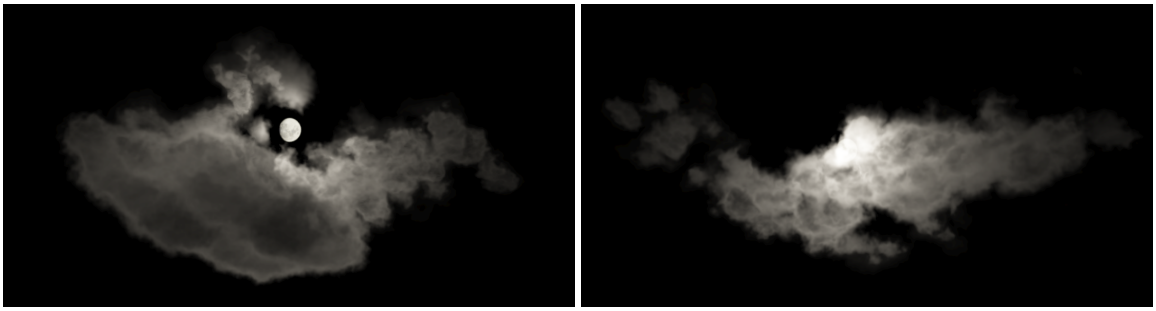


Figure 4.13: Examples of clouds rendered at night time.

Temporal coherence In order to avoid low frequency temporal noise in the rendered images it is necessary to maintain a low variance of the illumination solution I. Since the convergence rate of a Monte Carlo estimate is $O(n^{1/2})$, to maintain 2 % error in each grid cell the total number of photons in the scene n_t needs to be set so that each cell in I receives at least around $n \approx 2.5$k photons (together from all partial caches $\mathsf{H}_0, \ldots, \mathsf{H}_{m-1}$). Because the photon budget is roughly fixed, the grid resolution needs to be adjusted accordingly to fulfil this criterion (in addition to the limitation imposed by the available shared memory of GPUs, see Section 4.2.1).

Note however that it is necessary to distinguish photon beams and individual photons – a single photon beam can dispose many photons into the grid (Figure 4.5). There is of course no exact way to determine the ratio between these two for a particular non-trivial dataset, but a good initial approximation can be derived from the scattering coefficient σ_s, the average density of the dataset D and its spatial dimensions.

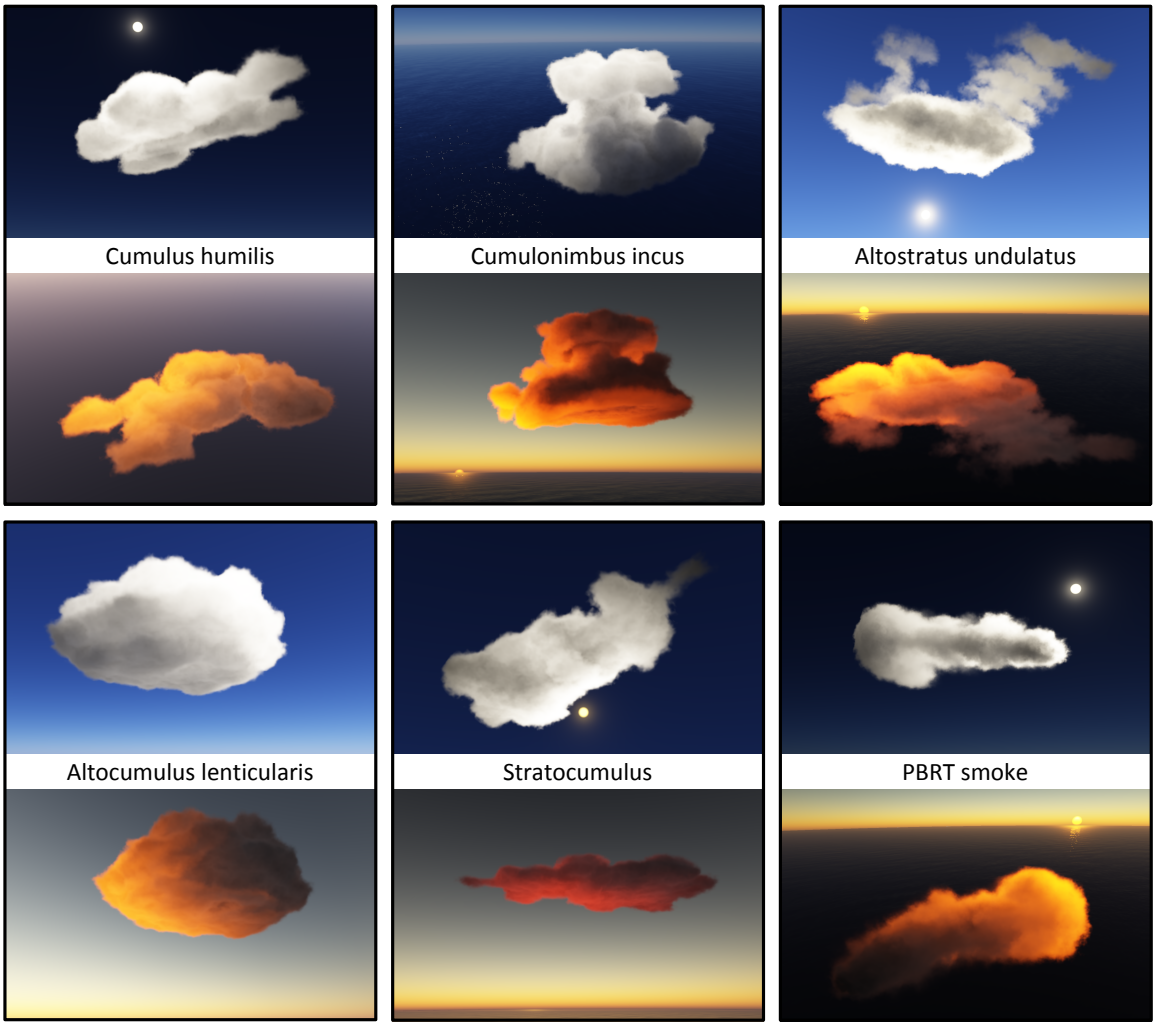


Figure 4.14: Additional results of the method. The apparent darkness of the sky in some images is caused by the used global tone mapping operator, the exposure of which is determined from the (very high) brightness of the clouds.

Chapter 5

Principal-Ordinates Propagation

Figure 5.1: Dense smoke exhibiting strong multiple anisotropic scattering produced by a steam locomotive under complex environment illumination. The presented method renders it dynamically without any precomputations at 25 Hz (NVidia GeForce GTX 770). Notice, e. g., the correct reproduction of the characteristic anisotropic illumination features *(top row)*.

5.1 Introduction

This chapter proposes *principal-ordinates propagation* (POP) – a novel, physically plausible method for interactive rendering of general heterogeneous participating media. The main focus is on optically thick, translucent media with arbitrarily anisotropic scattering, which, recalling the discussion in previous chapters, is one of the most difficult combinations to simulate. As in the other approaches presented in this work, POP has no reliance on any sort of precomputation, and therefore is fully suitable for simulating dynamic media. Several sample results are shown in Figure 5.1.

The core of the approach is to transport light in discrete propagation volumes oriented along a set of approximate *principal ordinates* of the source illumination. For this POP typically uses multiple rectilinear grids to propagate environmental (distant) lighting, and spherical grids to account for

point light sources. In both cases, the critical feature of the method is that one dimension of the grids is aligned with the prominent directional part of the source radiance for which the grid has been created.

In contrast to previous methods based on similar propagation paradigms (e. g., [Kaplanyan and Dachsbacher 2010; Billeter et al. 2012]), discretising the illumination into directional and point light sources enables POP to approximately describe the anisotropy (directionality) of light transport by a single scalar value per grid cell and per light source. Specifically, this anisotropy value corresponds to the unimodal Henyey-Greenstein distribution (Section 2.3.3) implicitly aligned with the respective principal ordinate (similarly to the representation used in Chapter 4). In addition to exploiting data locality and the parallelism of GPUs, the benefit of these decisions is a significant reduction of the *false scattering* (numerical dissipation) and *ray effects* (misalignment errors). These artifacts arise in many finite-element methods as a consequence of representing the propagated radiance by, e. g., spherical harmonics or piecewise-constant functions (see discussion in Section 2.5.2). POP also shares similarities with the finite-difference time domain method by Musbach et al. [2014], although only the radiance amplitude is considered in the propagation as the overall focus is on efficiency.

5.2 Principal-Ordinates Propagation

Virtually all existing variants or extensions of DOM (on which this work builds; also see Section 2.5.2) use a single scene-aligned propagation grid, where every cell stores a representation of the directional radiance function using spherical harmonics (SH) or piecewise-constant functions. This representation is then used to iteratively calculate energy transfer between nearby cells, typically within a local 6-, 18- or 26-neighbourhood. However, this approach is only suited for moderately anisotropic scattering at best – especially for anisotropic media under complex (high-frequency) illumination it causes prominent ray effects and false scattering artifacts.

The core aim of POP is to reduce these drawbacks, which are intrinsic to all iterative FE methods and stem from the fact that the propagation domain is generally not aligned with the prominent light transport directions. We in contrast propose to identify the most important light propagation directions (principal ordinates) in the scene and then use *multiple propagation grids* explicitly aligned with these directions, instead of a single one.

Additionally, this enables using a single scalar value per grid cell to describe the local anisotropy of the directional light distribution. This value is again the *anisotropy coefficient* which is used to parametrise the unimodal HG function in each grid cell, in a fashion similar to how the method from Chapter 4 represents radiance distributions in clouds. Using principal directions implies that for more complex lighting scenarios, it is necessary to use multiple grids that approximate their directionality sufficiently well; for local light sources we propose to use spherical grids centred around them.

These choices inherently assume that the principal directions can be derived from the initial radiance distribution and do not change strongly when light travels through the medium. Such variation certainly might occur if the density of the simulated medium changes abruptly. Still, we deem this to be a necessary compromise if speed is the priority, and as discussed in Section 5.3.5, violating this assumption does not cause our algorithm to fail, but only leads to a gradual decrease of accuracy.

The method is exposed as follows. First, the concept of principal-ordinates propagation itself is explained for a single directional source (Section 5.3). Then we describe how to extend this scheme to environment illumination (Section 5.4) and local light sources (Section 5.5) by using multiple, importance-sampled, rectilinear and spherical propagation volumes respectively.

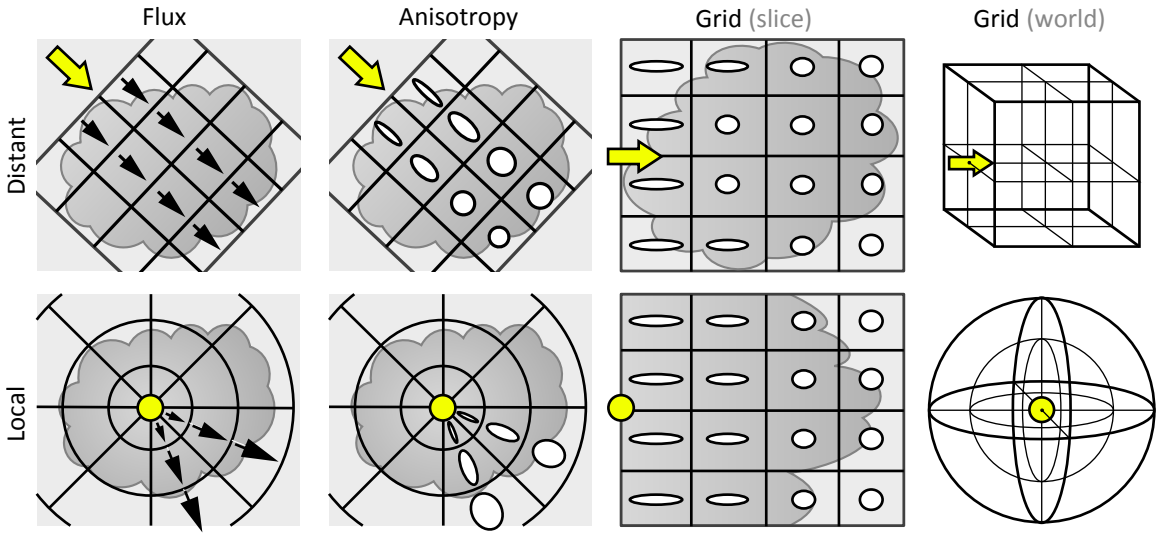


Figure 5.2: For distant (parallel) light POP uses rectilinear grids aligned with its principal direction, and spherical grids for point light sources. Every grid cell stores only radiance magnitude and anisotropy. The propagation scheme is almost identical for both cases.

The propagation scheme is explained using radiance as the radiometric quantity; it is assumed that all other quantities (such as irradiance from environment maps or flux from point lights) be converted accordingly. All frequently used notation is summarised in Table 5.1.

5.3 Rectilinear Propagation Grids (Directional Illumination)

The concept as well as the theory behind the proposed propagation scheme can be best explained for parallel (distant) light travelling along a direction $\mathbf{d}$ through a region in space (Figure 5.2, top). For this case the space is discretised into a uniform rectilinear grid similar to DOM; however, one of its dimensions is explicitly aligned with $\mathbf{d}$. For every grid cell i, the radiance magnitude L_i and its directional distribution are stored (all computations are performed independently per-wavelength, which is omitted here for brevity).

The main difference to DOM is that POP represents both the *directional distribution* of light and the medium *phase function* using the HG distribution implicitly aligned with $\mathbf{d}$. To distinguish radiance anisotropy (directional distributions) from phase functions, we denote the HG anisotropy parameter for the former as $a_i \in [-1,1]$, and $g \in [0,1]$ for the latter (negative values of g are not considered because of the physical implausibility of dominantly backscattering media). That is, the directional radiance distribution of a grid cell centred at $\mathbf{x}_i$ is expressed as $L(\mathbf{x}_i,\omega) = L_i \cdot f_{\mathrm{HG}}(\vartheta, a_i)$, where f_{HG} is the HG function (Section 2.3.3) and $\vartheta = \omega \cdot \mathbf{d}$ is the cosine of the angle between a direction ω and the principal light direction $\mathbf{d}$.

It is further assumed that the medium is characterised by its spatially-varying scattering coefficient σ_s and absorption coefficient σ_a; these two quantities as well as the (potentially also spatially varying) anisotropy of the phase function defined by the HG parameter g are wavelength-dependent and stored for every cell of the medium volume (which stands independently of the light propagation volumes).

Conceptually, two grids are required in the propagation procedure. The first, *propagation grid*, stores the unpropagated (residual) energy. This will be denoted as L and its state at the iteration $m \in \{1..M\}$,

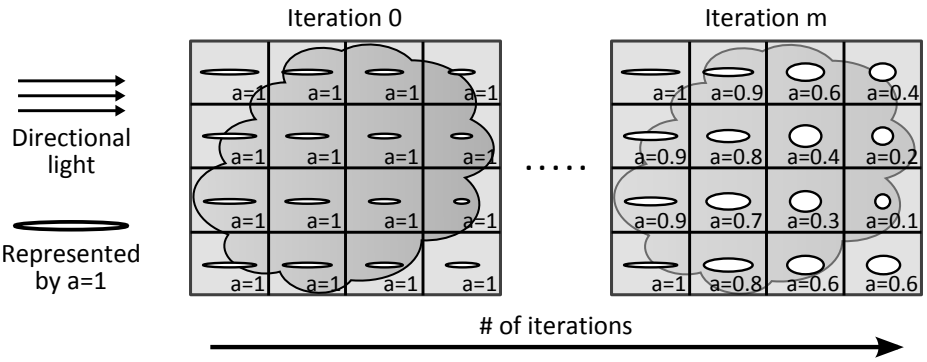


Figure 5.3: The propagation grid aligned with the direction of incidence is initialised with the attenuated radiance and anisotropy parameter $a_i = 1$ (which corresponds to perfectly forward Dirac distributions). During the propagation both radiance magnitude and anisotropy change towards lower anisotropy.

where M is the total number of propagation iterations, as L^m. The second, *accumulation grid* L_{acc}, is needed to accumulate the energy transported through the medium over the course of the computation.

Importantly, two options are available for implementing L_{acc}: it could either store the overall radiance distribution that has passed though each cell during the propagation, or alternatively store only the observer-dependent out-scattered radiance at each iteration. We opted for the second approach, because storing the entire directional radiance distribution at each cell is much more expensive than just accumulating the outgoing radiance (which is essentially a single scalar value per wavelength). Although this of course requires recomputing the solution on every observer position change, it is in agreement with our premise of a fully dynamic algorithm without relying on precomputations.

5.3.1 Grid Initialisation

At the beginning each propagation grid – which is scaled to span the entire medium (Figure 5.2, top) – needs to be initialised by the incident radiance at each cell. As no scattering has been accounted for yet, the anisotropy is set to the HG coefficient of $a_i = 1$, an equivalent of the Dirac function in the direction $\mathbf{d}$ (Figure 5.3). That is, for every cell, the algorithm computes the transmittance T_i (from the point where light enters the medium, travelling along $\mathbf{d}$ to $\mathbf{x}_i$) and sets the radiance magnitude to $L_i = L_{\text{in}}(\mathbf{d}) \cdot T_i$. Note that this can be efficiently computed using ray marching: as the grid is aligned with $\mathbf{d}$ the transmittance can be calculated incrementally along individual 'slices' of the grid in a single sweep along $\mathbf{d}$, accessing each cell only once.

5.3.2 Light Energy Propagation

This section describes the core step of the method: how to iteratively update the propagation grid to simulate light transport. The minimal propagation stencil is used, where the radiance of each grid cell is propagated to its six direct neighbours in every iteration. Specifically, akin to the other presented methods also POP uses the more GPU-friendly gathering-type computation of how much radiance flows *into* each grid cell from its neighbours based on their radiance distributions. These contributions are then combined to yield the new distribution at that cell (Figure 5.4, right). In the following we denote the neighbouring source cell with the subscript *src*, and the (current) target destination cell as *dst*.

Table 5.1: Table of frequently used symbols (in the order of appearance).

$\mathbf{d}$	(principal) direction
g	scattering anisotropy coefficient
σ_s, σ_a	scattering / absorption coefficient
$\mathbf{x}_i$	location of grid cell i
L_i, a_i	(per-cell) radiance magnitude and anisotropy
f_{HG}, F_{HG}	HG function and its cumulative distribution
ϑ	scattering angle cosine
$\llcorner$, $\llcorner_{\text{acc}}$	propagation and accumulation grid
M, m	number of iterations / iteration index
$L_{\text{in}}(\mathbf{d})$	incident radiance from direction $\mathbf{d}$
$\Delta L_{src\to dst}$	*src* to *dst* radiance contribution
T_i, $T_{src\to dst}$	transmittance to cell i and between cells
Ω_i, Ω_n	solid angle subtended by cell i or ordinate n
N, n	number of principal ordinates / ordinate index

Radiance magnitude contribution The first thing to determine is the amount of radiant energy that flows from cell *src* towards *dst* according to the radiance distribution in *src*. To this end, we efficiently integrate $L(\mathbf{x}_{src}, \omega)$ over the solid angle subtended by *dst* (denoted as $\Omega_{src\to dst}$ below) using the closed form of the cumulative HG function (see Equations 2.18 and 2.19). The radiance contribution from *src* travelling towards *dst* is then computed using the inter-cell transmittance $T_{src\to dst}$ as

$$\Delta L_{src\to dst} = L_{src} \cdot T_{src\to dst} \cdot \Big[F_{\text{HG}}(\cos\theta_1, \phi_1, a_{src}) - F_{\text{HG}}(\cos\theta_2, \phi_2, a_{src}) \Big] \tag{5.1}$$

using the following approximate parametrisation for the subtended solid angle $\Omega_{src\to dst}$ (depending on mutual positions of *src* and *dst*):

$$\Omega_{src\to dst}(\theta_1, \theta_2, |\phi_1 - \phi_2|) = \begin{cases} (0, \frac{\pi}{4}, 2\pi) & dst \text{ in front of } src \\ (\frac{\pi}{4}, \frac{3\pi}{4}, \frac{\pi}{2}) & dst \text{ next to } src \\ (\frac{3\pi}{4}, \pi, 2\pi) & dst \text{ behind } src \end{cases} \tag{5.2}$$

(see Figure 5.4, left, for a sample illustration of the second case of Equation 5.2). Since the HG distribution is rotationally-symmetric (Figure 5.4, middle) only the absolute value of the difference of the azimuthal angles $|\phi_1 - \phi_2|$ is required.

Note that here the transmittance $T_{src\to dst}$ accounts *just for absorption* that affects the radiance propagation on its way from *src* to *dst*. This is because POP treats scattering as a decrease of anisotropy and not as an extinction process, as explained below. For practical purposes, it is usually sufficient to take the averaged absorption coefficients σ_a at the source and destination cells and the distance between their centres t, and apply the Beer-Lambert-Bouguer law (Equation 2.21). To avoid aliasing if the resolution of the propagation grid is much smaller than the medium volume resolution, the medium parameters need to be sampled from a downscaled version of the volume. On GPU, downscaling is a very fast operation, as it corresponds to building a small number of MIP map levels (depending on the ratio between the grid resolutions, but usually 1 or 2 in the presented results). Alternatively, ray marching with a small number of steps might be used if for some reason building a MIP map is not desired.

Radiance anisotropy contribution Similarly to absorption attenuating the radiant energy flowing between neighbouring cells, the anisotropy of the energy propagated from *src* to *dst* will decrease

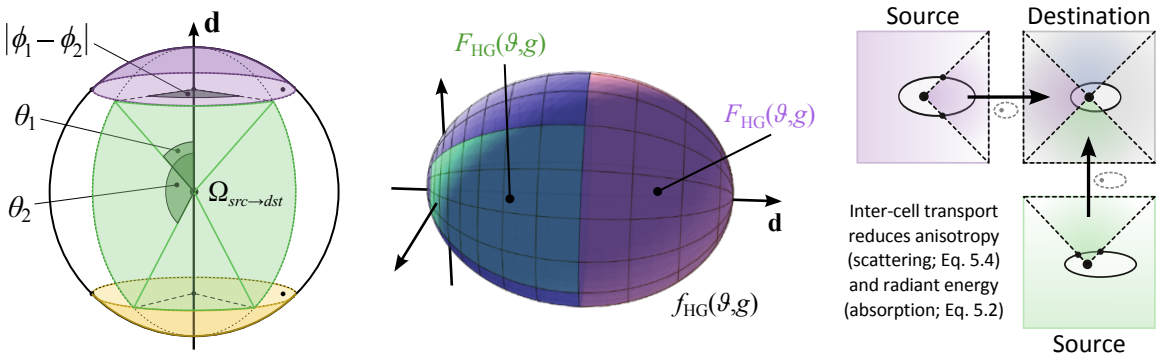


Figure 5.4: *Left:* Polar parametrisation of the solid sphere used in the propagation. The coloured patches correspond to the approximate solid angles subtended by the cells next to (green), in front (purple) and behind (orange) *src*. *Middle:* The HG cumulative function F_{HG} is used to integrate the radiance from the source cell flowing towards the destination cells (depicted as coloured patches of f_{HG}, for $g = 0.5$). *Right:* On the way the light undergoes scattering and is possibly reduced by absorption.

due to scattering. In agreement with the RTE, within the presented propagation scheme this can be computed exploiting the self-convolution property of the HG distribution. Recalling Equation 2.20, in a medium with scattering anisotropy of g the radiance anisotropy reduces to $a' = a \cdot g^{s \cdot \sigma_s}$ after travelling a distance s (assuming a constant σ_s along this path). Here σ_s and s are obtained the same way as for computing $T_{src \to dst}$ above. The change of radiance anisotropy from *src* to *dst* is therefore

$$\Delta a_{src \to dst} = a_{src} \cdot g^{s \cdot \sigma_s}. \tag{5.3}$$

We can easily see that this formula cannot lead to an increase of anisotropy, since $g \in [0, 1]$. Additionally, in non-scattering media ($\sigma_s = 0$) the directionality will be preserved perfectly, which is the sought-for behaviour.

Combining contributions from neighbours Updating the radiance distribution at the cell *dst* now entails accumulating the contributions from its six neighbours (indexed by *src*) as

$$L_{dst} = \sum_{src} \Delta L_{src \to dst}, \tag{5.4}$$

$$a_{dst} = \frac{\sum_{src} \Delta L_{src \to dst} \cdot \Delta a_{src \to dst}}{\sum_{src} \Delta L_{src \to dst}}. \tag{5.5}$$

While the radiant energy contributions simply need to be added up, the anisotropy is a weighted average of its neighbours, since the update has to yield an anisotropy value a_{dst} within the valid range $[0, 1]$. Figure 5.4, right, shows a sketch of this procedure.

This concludes the exposition of the elementary propagation operation. The advantage of the proposed formulation is its simplicity and speed, which is due to the fact that all integration is performed analytically. Moreover, the data access is coherent thanks to the propagation locality and the actual amount of data read per each cell is very small due to the compactness of the HG basis. However, such formulation of course bears certain implications, which are elaborated on in Section 5.3.5.

5.3.3 Iterating the Solution

The update procedure defined by Equations 5.4 and 5.5 is performed for every cell of L^m to yield L^{m+1} for every iteration m. Implementation-wise, this requires maintaining a second grid identical to the propagation grid and swapping these at each iteration.

Additionally, the results of every propagation iteration need to be accumulated in L_{acc} by evaluating the updated distributions in L^{m+1}:

$$\begin{aligned} L^{m+1}_{\text{acc},i} &= L^{m}_{\text{acc},i} + L^{m+1}(\mathbf{x}_i, \mathbf{e} - \mathbf{x}_i) \\ &= L^{m}_{\text{acc},i} + L^{m+1}_{i} \cdot f_{\text{HG}}(\vartheta, a^{m+1}_i) \end{aligned} \tag{5.6}$$

for every cell i. Here $\mathbf{e}$ is the observer position and μ is therefore the dot product of $\mathbf{d}$ and the view direction.

5.3.4 Upsampling and Rendering

When the solution has converged after a sufficient number of iterations (discussed later), using it for rendering is relatively straightforward. Again, ray marching is employed to integrate the incoming radiance for every camera ray using the common front-to-back emission-absorption model [Max 1995]. In this case the emission term corresponds to the scattered radiance accumulated in L_{acc}, which in each cell already contains a view dependent outgoing radiance value (see Equation 5.6).

As discussed in Section 5.6, for performance reasons the typical resolutions of the propagation volumes need to be kept small (most of the shown examples use 20^3 grids or less). In order to improve the rendering quality with such low grid resolutions it is desired to upsample them prior to their visualisation. For this the density field of the medium D (in other words the spatially varying scattering coefficient) is used as a guiding signal and the same 3D upsampling technique is applied as in the previous chapter (Section 4.2.4). Typically, the density field is significantly more detailed than the propagation volumes; this detail is 'transferred' to the solution by the upsampling procedure. According to our experiments, low-resolution propagation grids are usually sufficient for plausible results.

5.3.5 Discussion of the Propagation Scheme

Using pre-aligned unimodal HG functions to represent directional radiance distributions obviously means that there are distributions in a cell that cannot be represented well. On the other hand, this is compensated for this by using multiple grids (Section 5.4), each of which can handle anisotropic scattering significantly better than previous work thanks to the design of the propagation scheme. For a comparison, an exceedingly large number of SH coefficients is required to represent highly anisotropic distributions, and this still does not prevent false scattering issues if a local propagation scheme is employed.

The most approximate step of our scheme is arguably the recombination of reduced anisotropies from neighbouring cells (Equation 5.5). The logic behind this formulation is that the radiance distribution at dst will result from superposing the neighbouring distributions according to how much energy they contribute to dst. The main limitation of this approach lies in the fact that combining multiple HG distributions with different anisotropy values cannot generally be represented by any single HG distribution. Although experimenting with fitting the resulting HG distribution to the combination

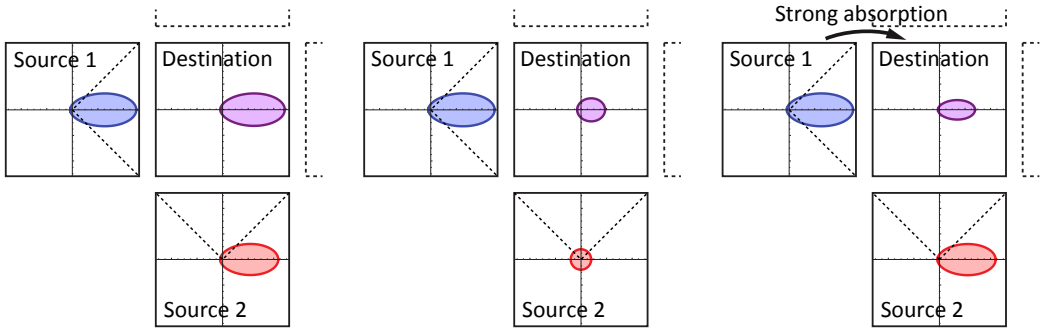


Figure 5.5: Three examples of local propagation behaviour. *Left:* all source cells exhibit strong forward scattering which is preserved well by our propagation scheme. *Centre:* radiance anisotropy is reduced due to in-scattering from Source 2 which has isotropic radiance distribution. *Right:* light from Source 1 to destination is almost entirely absorbed. Light from Source 2 should then be deviated 'upwards', which our scheme cannot represent.

of its neighbours in terms of the least-square error, we found that the simple weighted arithmetic average produces comparable results while keeping the computational cost of this core operation minimal. In addition, Equation 5.5 preserves the anisotropy of light transported along the principal direction very well, thus greatly reducing any apparent false scattering effects.

Note that there are cases of very heterogeneous media where POP might locally become inaccurate (see Figure 5.5). If light travelling along the principal direction undergoes strong absorption, while light from other directions does not, the resulting distribution could potentially become skewed, which cannot be represented within the proposed framework. Although this is obviously a failure case of this representation, occurrences of such strong absorption fluctuations are comparatively rare, and more importantly, the resulting radiance magnitude in these cases is typically very small (therefore having little impact on the resulting image). Also note that multiple propagation volumes can actually faithfully reproduce complex multimodal radiance distributions, despite each grid being composed of unimodal HG distributions.

In the remainder of this section, two key points of the method are analysed: the proof of energy conservation of Equation 5.4 and its convergence to an energy equilibrium, and a more detailed connection of Equation 5.5 to the underlying light transport physics (as given by the RTE).

HG propagation convergence An important concern in an iterative light transport solver is to ensure its convergence to an energy equilibrium in a finite number of iterations. The following demonstrates that this is indeed the case for POP.

Recalling Equation 5.1 for computing the radiance contribution $\Delta L_{src\to dst}$ from cell *src* to cell *dst*,

$$\Delta L_{src\to dst} = L_{src} \cdot T_{src\to dst} \cdot \Big[F_{\mathrm{HG}}(\cos\theta_1, \phi_1, a_{src}) - F_{\mathrm{HG}}(\cos\theta_2, \phi_2, a_{src}) \Big], \tag{5.7}$$

we need to verify that this formula does not distribute more radiant energy to the six neighbouring *dst* cells that the source cell *src* contains. For this, first consider an empty propagation domain, i. e., $\sigma_t = 0$ in the entire considered portion of the scene. Then the transmittance $T_{src\to dst} = 1$ and Equation 5.7 simplifies to

$$\Delta L_{src\to dst} = L_{src} \cdot \Big[F_{\mathrm{HG}}(\cos\theta_1, \phi_1, a_{src}) - F_{\mathrm{HG}}(\cos\theta_2, \phi_2, a_{src}) \Big], \tag{5.8}$$

If we first consider propagating the energy from *src* to the entire sphere Ω, so that $|\phi_1 - \phi_2| = 2\pi$,

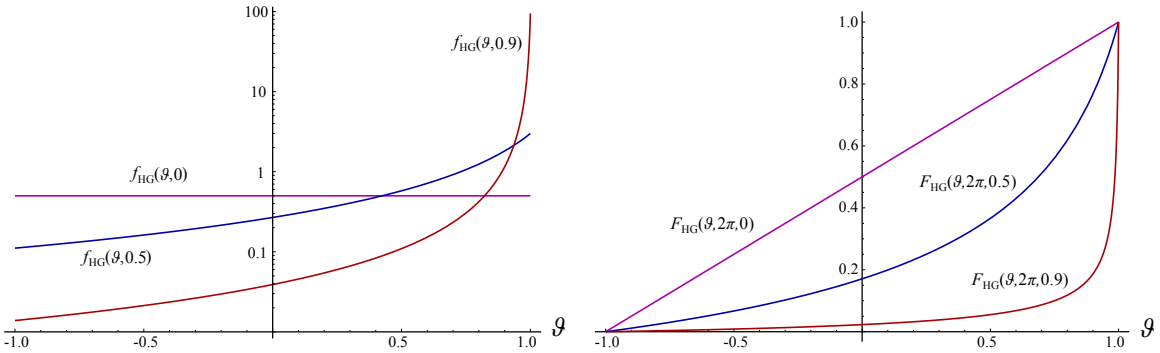


Figure 5.6: Plots of f_{HG} (left) and F_{HG} (right) for parameter g values of 0, 0.5 and 0.9. Please note that the plots of f_{HG} have a logarithmic scale and the values of F_{HG} are evaluated for the full azimuthal range, i. e., $|\phi_1 - \phi_2| = 2\pi$.

$\theta_1 = 0$ and $\theta_2 = \pi$ and the resulting propagated energy will be

$$\begin{aligned}\Delta L_{src\to\Omega} &= L_{src} \cdot \Big[F_{HG}(\cos 0, 2\pi, a_{src}) - F_{HG}(\cos\pi, 2\pi, a_{src})\Big] \\ &= L_{src} \cdot (1-0) = L_{src},\end{aligned} \tag{5.9}$$

as F_{HG} by definition (Equation 2.18) peaks at 1 regardless of its anisotropy parameter (see Figure 5.6, right). At this point it is only necessary to ensure that our parametrisation of the solid angles $\Omega_{src\to dst}$ subtended by the six neighbouring cells divides the sphere of directions into 6 mutually exclusive solid angles without any gaps. Using the approximate parametrisation in terms of polar angles from Equation 5.2 (for illustration see Figure 5.4, left) we can rewrite Equation 5.9 as (extracting the azimuth ϕ from inside of F_{HG} and leaving out the anisotropy parameters a_{src} of F_{HG} for better clarity)

$$\begin{aligned}\Delta L_{src\to\Omega} &= L_{src} \cdot 2\pi \cdot \Big[F_{HG}(\cos 0) - F_{HG}(\cos\pi)\Big] \\ &= L_{src} \cdot 2\pi \cdot \Big[F_{HG}(\cos 0) - F_{HG}(\cos\frac{\pi}{4})\Big] + \\ &+ 4 \cdot L_{src} \cdot \frac{\pi}{2} \cdot \Big[F_{HG}(\cos\frac{\pi}{4}) - F_{HG}(\cos\frac{3\pi}{4})\Big] + \\ &+ L_{src} \cdot 2\pi \cdot \Big[F_{HG}(\cos\frac{3\pi}{4}) - F_{HG}(\cos\pi)\Big].\end{aligned} \tag{5.10}$$

It is now easy to see that this decomposition corresponds to an expansion of the following equation

$$\Delta L_{src\to\Omega} = L_{src} = \sum_{dst} \Delta L_{src\to dst} \tag{5.11}$$

when the corresponding parameters for each of the six *dst* cells are used according Equation 5.2. This, however, is exactly the scattering equivalent of the energy accumulation defined in Equation 5.4, since it iterates over the same neighbourhood.

It can now be seen that propagating radiant energy either *from* a cell (Equation 5.11) or gathering *into* a cell (Equation 5.4) by using $\Delta L_{src\to dst}$ as defined in Equation 5.1 conserves energy perfectly. Therefore, unless a cyclic propagation domain is used, even at this point the algorithm will eventually converge, since a small portion of the energy contained in the propagation volume will dissipate away from its outermost cells in every iteration. Furthermore, in an absorbing medium ($\sigma_a > 0$) the transmittance becomes $T_{src\to dst} < 1$, which will lead to an additional attenuation of the propagated energy, increasing the rate of convergence.

Radiance anisotropy contribution In spite of the fact that POP utilises some heuristic steps to achieve better performance in exchange for bias, it is still based on the underlying mathematical description of light propagation in participating media. The following shows the connection between the radiative transport equation and the way POP treats scattering as an iterative convolution of radiance distributions and the medium phase function.

The first part follows the argumentation of Premože et al. [2003; 2004] in their work on formulating light scattering within the path integral framework. Recalling the *time-dependent* version of the RTE [Chandrasekhar 1960],

$$\left(\frac{\partial}{\partial s} + \omega \cdot \nabla + \sigma_t\right) L(s,\mathbf{x},\omega) = \\ L_e(\mathbf{x},\omega) + \sigma_s(\mathbf{x}) \int_\Omega f(\mathbf{x},\omega',\omega) L(s,\mathbf{x},\omega')\, \mathrm{d}\mu(\omega'), \tag{5.12}$$

one can already see that the change of the directional radiance distribution L at the position $\mathbf{x}$ is a spherical convolution with a general phase function f. Here the dependence on time t is expressed in terms of travelled distance $s = ct$, with c being the speed of light, and L_e is the (initial) source radiance distribution.

Equation 5.12 can be expressed as a convolution of L_e with the Green's function G (see Section 2.5.3):

$$L(s,\mathbf{x},\omega) = \int\int G(s,\mathbf{x},\mathbf{x}',\omega,\omega') L_e(\mathbf{x}',\omega')\, \mathrm{d}\mu(\mathbf{x}')\, \mathrm{d}\mu(\omega'), \tag{5.13}$$

Recall that the Green's function $G(s,\mathbf{x},\mathbf{x}',\omega,\omega')$ represents the radiance distribution at $(\mathbf{x},\omega)$ and time s, originating at $(\mathbf{x}',\omega')$ and zero time. Mathematically, G is the solution of the homogeneous version of Equation 5.12 when the initial boundary condition is expressed as

$$G(s=0,\mathbf{x},\mathbf{x}',\omega,\omega') = \delta(\mathbf{x}-\mathbf{x}')\delta(\omega-\omega'), \tag{5.14}$$

where δ is the Dirac function. Premože et al. [2003; 2004] have shown that G can be used to formulate light propagation in non-scattering and single-scattering media, and described an approximate analytical solution to general multiple scattering formulated via G (Section 2.5.3, Equation 2.36).

At this point we can use the expansion of G into a superposition of 'partial' Green's functions. Consider a propagation from an initial source position $\mathbf{x}_I$ to a final destination position $\mathbf{x}_F$. The propagator $G(s,\mathbf{x},\mathbf{x}',\omega,\omega')$ can be expressed as

$$G(s,\mathbf{x}_I,\mathbf{x}_F) = \\ \lim_{N\to\infty} \int \cdots \int G(s/N,\mathbf{x}_I,\mathbf{x}_1) \cdots G(s/N,\mathbf{x}_{N-1},\mathbf{x}_F)\ \mathrm{d}\mu(\mathbf{x}_1) \cdots \mathrm{d}\mu(\mathbf{x}_{N-1}) \tag{5.15}$$

omitting the directional parameters ω_I and ω_F for clarity. This expansion is called a *path integral* and its idea is to break the path into infinitesimally small segments and propagate energy within each segment separately. Now, if s is equal to the distance between $\mathbf{x}_I$ and $\mathbf{x}_F$ then all the auxiliary vertices $\mathbf{x}_1 \ldots \mathbf{x}_{N-1}$ will lie on a straight path. Then, during the propagation, the change of the directional distribution of energy will correspond to a spherical convolution with the partial propagator along each linear segment.

This is exactly the notion utilised in the proposed propagation scheme. In POP, a propagation volume corresponds to a discretisation of Equation 5.15 where each pair of neighbouring cells is virtually connected by a linear path segment. The propagation then approximates the change of the directional radiance distribution along this segment.

Since the radiance distributions at each cell are aligned, we can define $\omega = \omega_{\text{src}} = \omega_{\text{dst}}$. An additional assumption is that the scattering coefficient σ_s can be considered constant along the path between neighbouring cells *src* and *dst* (calculated as an average of the neighbouring cells, see Section 5.3.2). In this context, the partial propagator can be defined as

$$G(s, \mathbf{x}_{src}, \mathbf{x}_{dst}, \omega) = \sigma_s \int_s \int_\Omega f(\omega, \omega') \; \mathrm{d}\mu(\omega') \, \mathrm{d}s' \tag{5.16}$$

where $s = \|\mathbf{x}_{dst} - \mathbf{x}_{src}\|$ and f is a general phase function. However, since we describe the directional distribution of radiance at *src* by a HG distribution parametrised by an anisotropy parameter a_{src}, and also model the medium phase function by a HG function with an anisotropy g, the change of the directional radiance distribution can be expressed as

$$\Delta a_{src \to dst} = a_{src} \cdot g^{s \cdot \sigma_s}, \tag{5.17}$$

which is obtained by utilising the self-convolution property of HGF. We can now see that this is exactly Equation 5.3, which POP uses to describe the radiance anisotropy change when propagating radiance between two neighbouring cells.

5.4 Multiple Propagation Grids (Environment Illumination)

The previous section describes the HG-based light propagation for a single directional light source. In order to account for environmental lighting (typically modelled by an environment map), multiple grids oriented along different principal directions will be necessary.

This section discusses how to choose these directions (Section 5.4.2) and, as every grid accounts for light from a finite solid angle, how to prefilter the respective incident radiance to avoid singularity artifacts, which can arise for highly anisotropic media when using the proposed propagation scheme (Section 5.4.1). Further, Section 5.4.3 describes how multiple propagation grids are combined together for rendering. Finally, Section 5.4.4 presents an additional (optional) step in the POP pipeline, which allows splitting the propagation into two stages, anisotropic and isotropic, greatly improving the convergence for media with high albedo values.

5.4.1 Prefiltering

A straightforward approach is importance-sampling the environment map to obtain N directions, $\mathbf{d}_n$, each carrying an energy corresponding to its associated portion of the directional domain Ω_n. To avoid weak singularities arising from the discretised illumination representation (see Figure 5.7), POP can account for the size of Ω_n when determining the initial directional radiance distributions (parameter a_i in Section 5.3.1). Recall that the anisotropy parameter of f_{HG} represents the average cosine of the distribution. We can therefore approximate the initial anisotropy in a cell i as

$$a_{n,i} = \frac{\int_{\Omega_n} \langle \mathbf{d}_n, -\omega \rangle \; \mathrm{d}\mu(\omega)}{\|\Omega_n\|}, \tag{5.18}$$

i. e., the average cosine between $\mathbf{d}_n$ and the directions coming from within Ω_n, and use this value for the grid initialisation.

In practice, $a_{n,i}$ can be estimated without the integration over Ω_n for each ordinate or without even knowing the exact shape of Ω_n. As POP importance-samples the environment map, the importance

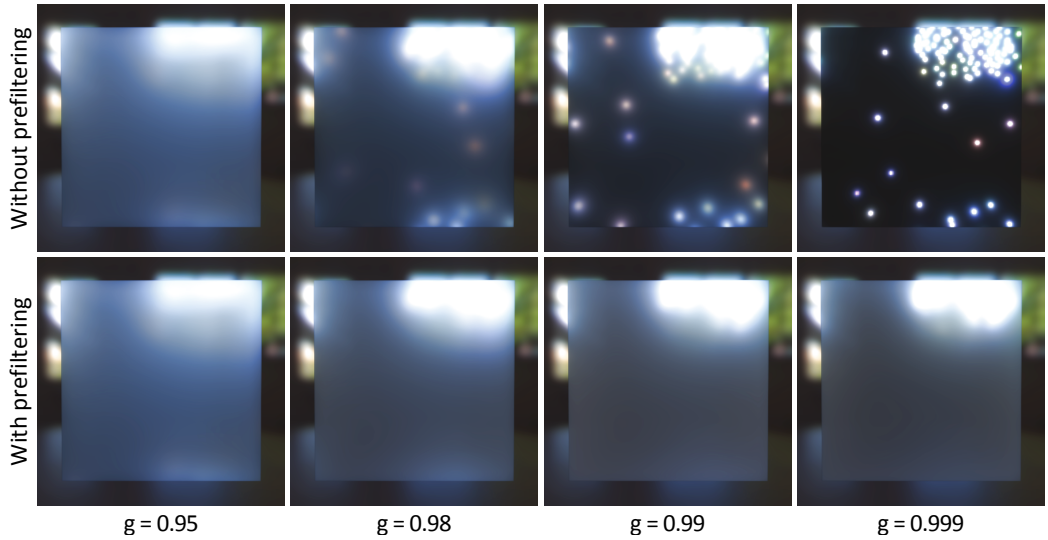


Figure 5.7: The effect of prefiltered initialisation on a thin, strongly-scattering medium with increasing anisotropy *(left to right)*. Without prefiltering the individual ordinates become apparent. Using prefiltering the resulting images become much smoother and yield an expected appearance (more anisotropic slabs appear more transparent). Note that the prefiltering is energy-conserving (as opposed to, e. g., singularity clamping in instant radiosity methods).

of the ordinate n is proportional and (up to a factor) very similar to the actual solid angle of Ω_n. Therefore, we propose a heuristic that maps the importance $w_n \in [0,1]$ to anisotropy as

$$a_{n,i} \approx (1 - w_n/N)^{\beta}, \tag{5.19}$$

so that important ordinates are denser in the directional domain and will have small solid angles and high anisotropy, and vice versa, less important ordinates are sparser, and will have larger solid angles and low anisotropy. The scalar factor $\beta > 0$ defines the proportionality and currently needs to be tuned manually once for each environment map; from our experience this is a simple and quick task.

5.4.2 Importance Propagation

The described sampling scheme can be further improved by considering how much illumination from different directions actually contributes to the image. To this end, an additional *importance propagation* step can be introduced before sampling the environment map: using a regular grid (perspective-warped into the camera frustum and oriented along the view direction) importance from the camera can be propagated through the medium. Thanks to the duality of light transport this is equivalent to the radiance propagation as described before.

The result of this propagation is a *directional importance distribution* stored in the grid cells. By ray-marching this grid the importance can be projected into the directional domain, creating a directional *importance map* that aligns with the environment map. POP then samples the environment map according to its product with the importance map (see Figure 5.8). We demonstrate that in certain situations this step improves the sampling result, mainly when a low number of propagation grids is used (Figures 5.8 and 5.15). It is also quite a cost-effective operation, since the directional importance function is typically very smooth and therefore only low resolutions for the propagation grid and the directional map are required (all examples use the resolutions of 16^3 for the former and 32×16 for the latter).

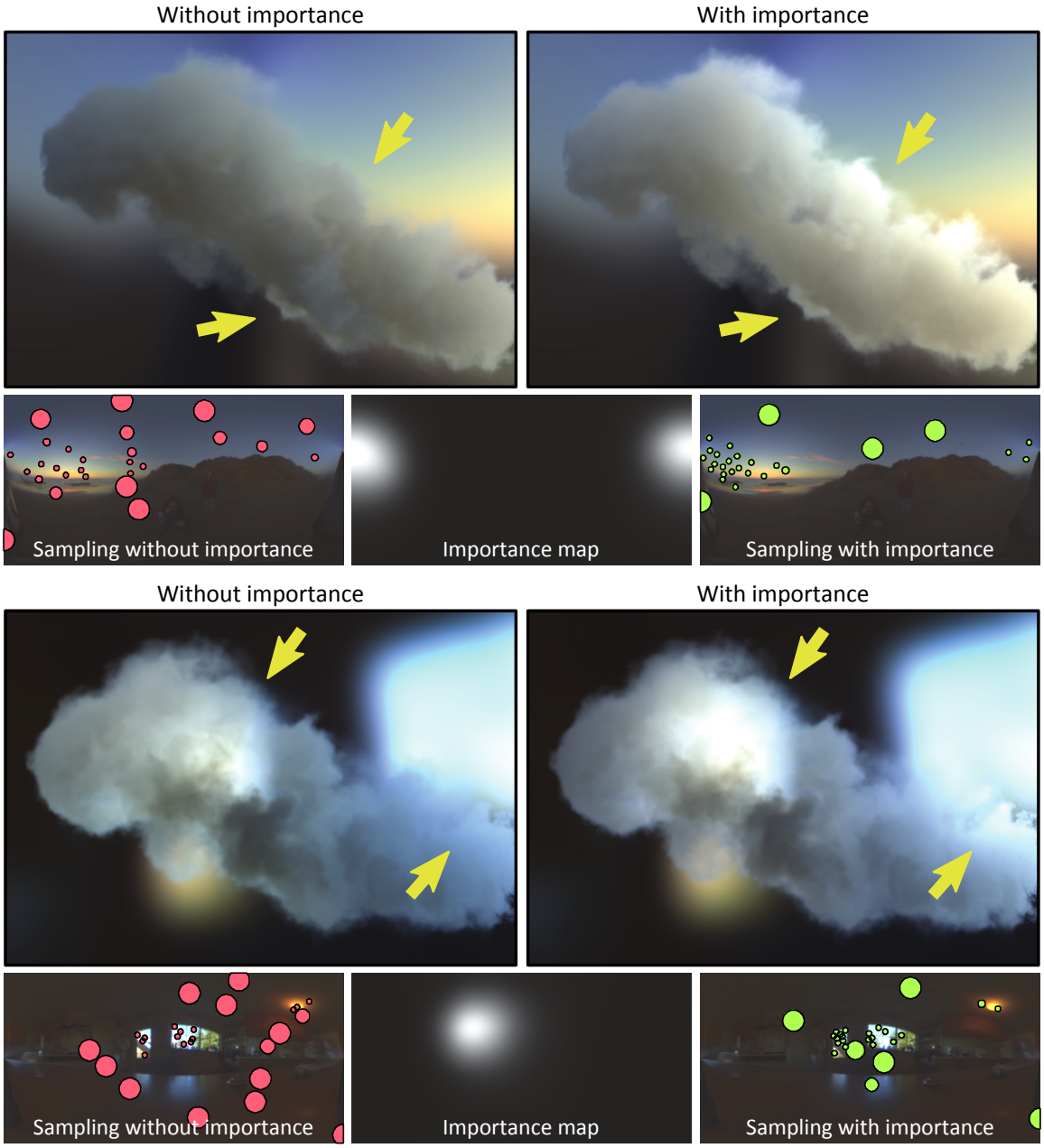


Figure 5.8: Importance propagation improves overall radiance distribution across the medium and visibility of bright regions behind *(yellow arrows)*. This especially holds for high-albedo media with strong scattering anisotropy (here $g = 0.96$) and when using a low number of ordinates (27 here). The sampling visualisations compare sampling patterns with and without incorporating the importance map. Here the sample disk sizes are proportional to the corresponding sample's solid angle Ω_n (as described in Section 5.4.1), although the sizes are clamped to prevent obscuring of the images. It is clearly visible that including importance into the sampling process concentrates the samples to the areas that have a bigger impact on the resulting appearance, which in most cases will be in the direction of the camera's main axis.

5.4.3 Merging Multiple Grids

Computing the transport for each of the N principal ordinates (as detailed in Section 5.3) yields a separate, view-dependent accumulation grid $\mathsf{L}_{\mathrm{acc},n}$ for each. Although it is possible to visualise these directly, this would be very inefficient as each grid in the set would have to be accessed at every ray-marching step.

Because of this we instead opt to combine all $\mathsf{L}_{\mathrm{acc},n}$ into a single medium-aligned grid, prior to upsampling and visualisation (Section 5.3.4). This merged grid typically has double the resolution of the individual propagation grids, since these are oriented arbitrarily in space (and therefore not increasing the resolution would result in undersampling). This is however still a very fast step which, as an additional benefit, also allows the remainder of the pipeline to stay virtually identical to the single-ordinate setting (even when propagation from local sources is included, as detailed in Section 5.5).

5.4.4 Isotropic Residuum

The solution is considered converged when the residual energy in all L_n^m is below a small threshold ϵ_{L}. This can however take a large number of iterations for high-albedo media, a problem inherent to all finite-element transport methods. On the other hand, we can observe that scattering reduces the anisotropy of the radiance distribution, so the propagation can be treated as (near-)isotropic as soon as $|a_{n,i}| < \epsilon_{\mathrm{a}}\ \forall i, \forall n$, again for a small anisotropy threshold ϵ_{a}.

As soon as all propagation grids fulfil this criterion the energy from them can be merged into a single grid aligned with the medium, as there is no directionality present anymore. This is similar to merging the accumulation grids (Section 5.4.3), except that here the propagation grids are merged as well and the propagation process switches to isotropic scattering (i. e., the anisotropies $a_{n,i}$ need not be maintained anymore). This decreases the propagation costs tremendously, as from this point it is performed just for a single global grid instead of one grid per principal ordinate (independently on the number of ordinates N).

In practice, we can determine the iteration m' when performing the switch to the cheaper isotropic propagation based on the maximum radiance anisotropy, $\widehat{a}$ (for a directional light $\widehat{a} = 1$, but pre-filtering can lower it, to our benefit), and phase function anisotropy g. Both of these parameters are already determined at the initialisation. Once again, making use of the the HG self-convolution property (Section 2.3.3), one can estimate the distance s that light has to travel such that its anisotropy falls below ϵ_{a} from the implicit equation

$$\widehat{a} \cdot g^{\overline{\sigma_{\mathrm{s}}} \cdot s} = \epsilon_{\mathrm{a}}, \tag{5.20}$$

where $\overline{\sigma_{\mathrm{s}}}$ is the average scattering coefficient in the medium. Furthermore, knowing that the travel distance depends on the average grid spacing Δx and the number of iterations, i. e., $s = m \cdot \Delta x$, m' can be obtained as

$$m' = \frac{\log \epsilon_{\mathrm{a}} - \log \widehat{a}}{\log g} \cdot \frac{1}{\overline{\sigma_{\mathrm{s}}} \cdot \Delta x}. \tag{5.21}$$

It is however usually a good idea to use m' at least equal to the grid resolution along the propagation direction, to allow for light even from the first 'slice' of cells to sufficiently penetrate into the rest of the volume (cf. [Kaplanyan and Dachsbacher 2010]). The subsequent propagation then operates on the residual isotropic radiance in the merged grid, iterating until the residual energy falls below ϵ_{L}.

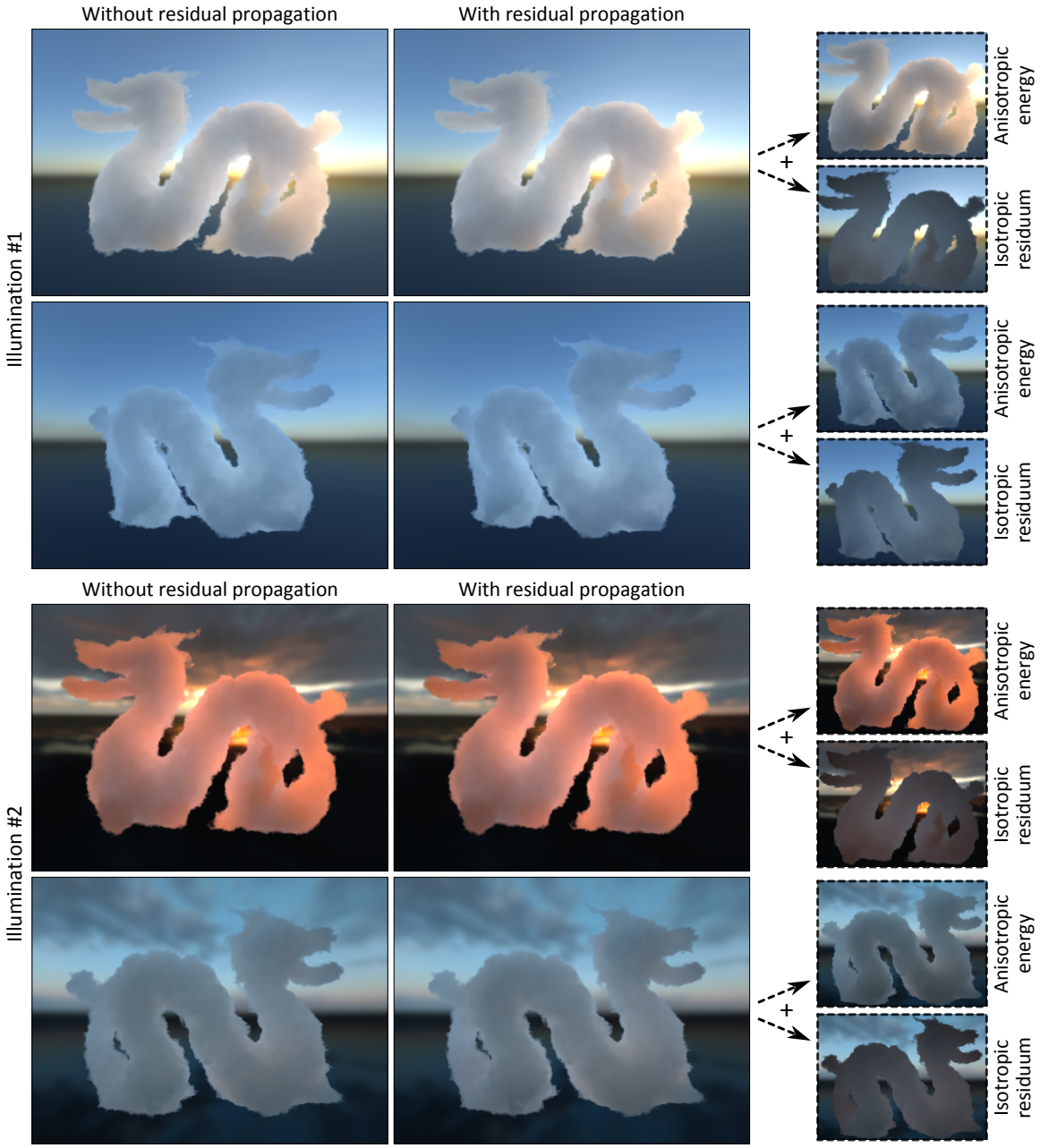


Figure 5.9: Evaluation of the isotropic residuum (IR) propagation for a dense high-albedo medium. The dragon dataset is rendered from two opposing viewpoints to demonstrate the view dependence of the resulting light distributions. The used medium (milk [Narasimhan et al. 2006], $\sigma_s = \{0.91, 1.07, 1.25\}\,\mathrm{m}^{-1}$, $g = 0.95$) has an albedo over 99.9 % across the visible spectrum, making it a difficult material to render because of the high number of iterations necessary to converge. The full propagation (64 ordinates, 16^3 grid each) requires about 100 iterations to converge, taking 120 ms (illumination #1) and 90 ms (illumination #2). In comparison, the proposed heuristic (Equation 5.21) switches to isotropic propagation after $m' = 20$ full anisotropic iterations ($\epsilon_a = 0.1$, $\Delta x = 0.25\,\mathrm{m}$). The IR propagation requires additional 20 iterations in a single 32^3 propagation grid. The combined propagation time for this case was, respectively for #1 and #2, 28 ms and 24 ms (25 ms / 20.5 ms anisotropic part, 3 ms / 3.5 ms isotropic part). This is about a 4-fold speed-up compared to the full propagation, with negligible visual difference.

Reasonable values for ϵ_a are around 0.1, or even higher. In fact this decision is not unlike the one made in similarity theory [Wyman et al. 1989], in which after a certain number of scattering events the propagation switches to isotropic scattering, which is accompanied by a switch to a so-called *reduced scattering coefficient*. This is usually done on an empirical basis and despite the fact that using the Henyey-Greenstein phase function allows us to quantify the decision better (discussed in detail in Section 4.2.1) this approach is still an approximation. Similarity theory also does not apply easily to heterogeneous media [Zhao et al. 2014]. However, thanks to the fact that POP treats scattering as a gradual decrease of anisotropy, the transition to isotropic propagation can be done in a well controlled manner, and without changing the propagation parameters or compromising the solution accuracy (aside from small geometric misalignments caused by the grid merging). This is demonstrated in Figure 5.9.

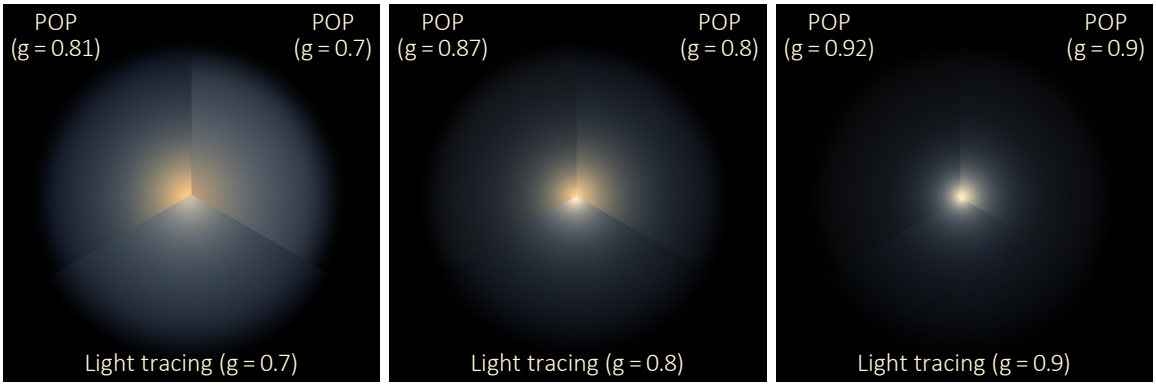


Figure 5.10: Comparison of the radial propagation to a Monte-Carlo reference for a uniform spherical medium (radius 2.5 m, $\sigma_s = \{0.8, 1, 1.3\}\,\mathrm{m}^{-1}$ and unit albedo). The resolution of the radial propagation grid was 32^3. Our solution differs from the reference mainly due to low (but for this propagation type still present) false scattering, in particular with lower anisotropy values. We found that this can be reduced by artificially increasing g, if a specific appearance is desired.

5.5 Radial Propagation Grids (Local Light Sources)

In order to extend POP to local light sources, spherical grids can be used, with two angular coordinates and a radial coordinate which is again aligned with the initial principal directions of the point source (Figure 5.2, bottom). To obtain more isotropic cell shapes, the spacing of shells along the radial coordinate grows exponentially (in proportion to the radial segment length at a given radius).

For parametrising the spherical domain, out of many possible options we opt for the octahedral parametrisation [Praun and Hoppe 2003] mainly as it is simple, provides reasonably uniform sampling, and above all, it discretises the domain into a 2D square where every cell has four natural neighbours (plus two along the radial axis), similarly to rectilinear grids. The resulting grid is thus topologically equivalent to a rectilinear grid (except for being cyclic in the two angular dimensions) and albeit not being uniform, it allows to approximately treat the space as locally Euclidean and obtain plausible results again using virtually the same propagation scheme as before. The main difference in the propagation is that the quadratic fall-off needs to be accounted for: although the propagation is based on radiance, it is necessary to explicitly compensate for the varying grid cell sizes resulting from the non-uniform shell spacing. To this end, POP scales the radiance when propagating along the principal direction in proportion to the radial coordinate spacing. A sample

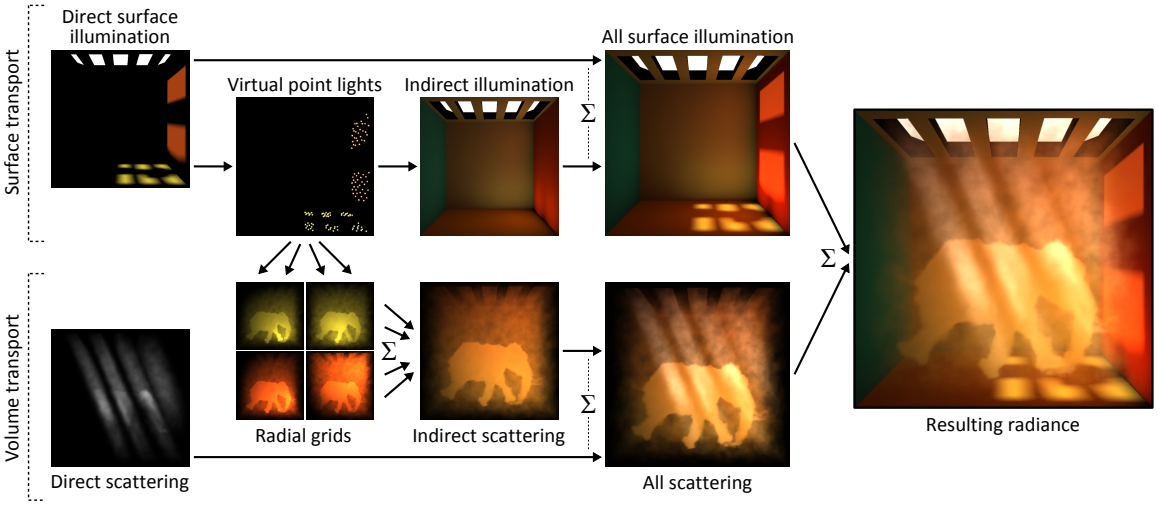


Figure 5.11: Workflow of the presented algorithm for a single directional light. For distant environment illumination the volumetric part of the pipeline is very similar, with the exception of rectilinear grids being used to propagate illumination from distant ordinates instead of the combination of VPLs and spherical grids.

'didactical' demonstration of this propagation type for a point light in a simple homogeneous spherical medium is shown in Figure 5.10, including a comparison to MC reference.

Instant radiosity Given the ability to use local point lights, we can take advantage of instant radiosity methods [Keller 1997; Engelhardt et al. 2010; Engelhardt et al. 2012], which represent complex illumination as a collection of virtual point lights (VPLs), to simulate surface-to-volume light transport. Normally these VPLs are obtained from random walks through the scene. In our interactive setting, VPLs can be generated using a reflective shadow map (RSM) [Dachsbacher and Stamminger 2005] for every primary light. These RSMs are importance-sampled according to surface albedo and (attenuated) irradiance, aiming at keeping the total number of VPLs low. The reflected radiance is then used to initialise the radial propagation grids.

Prefiltering can be done in the same way as for environment maps: VPLs with a large importance have a high initial anisotropy and vice versa. Similar to surface lighting, using clamping to reduce any remaining singularities [Engelhardt et al. 2012] is also desirable. Figure 5.11 depicts the pipeline of the algorithm when propagating scattering from one directional light and VPLs generated from its RSM.

5.6 Results and Analysis

All results were computed on a desktop PC with a 3.7 GHz Intel Xeon CPU, 16 GB RAM and NVidia GeForce GTX 770 GPU with 2 GB of VRAM. Our implementation is written in C++, using OpenGL and GLSL for the GPU code. In all presented measurements we use the framebuffer resolution of 800×600 in order to let the computation time be dominated by the propagation rather than ray-marching (which itself should be subject to optimisations in order to reach full efficiency). Resolutions of the medium density datasets are typically in the order of tens of voxels in each dimension (but effectively enhanced by procedural noise, similarly to Chapter 4).

Although the number of propagation iterations needs to be chosen empirically at the moment, in

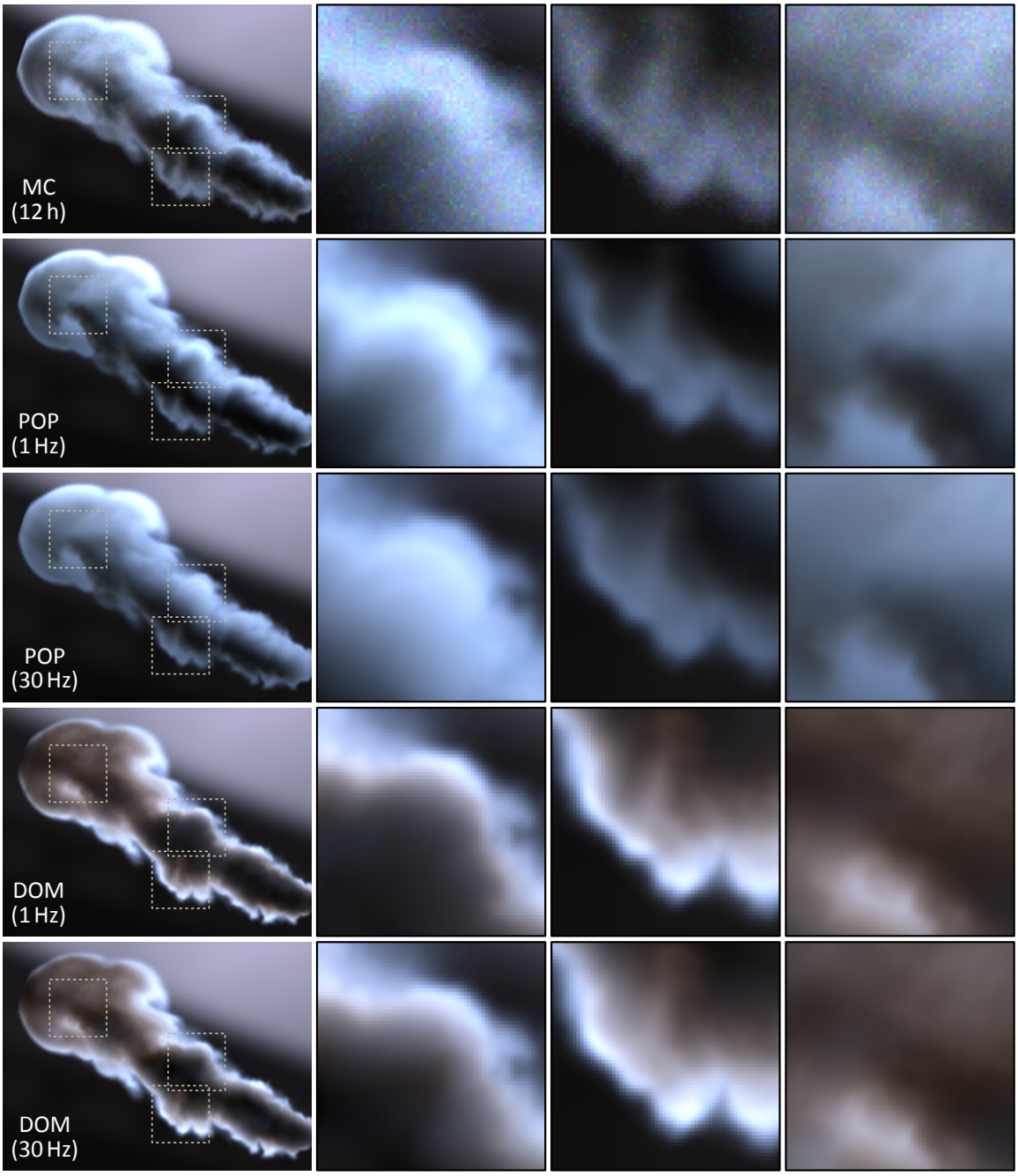


Figure 5.12: Comparison of principal-ordinates propagation (POP) to SHDOM and a Monte-Carlo reference (light tracing), for a smoke plume 10 m across with $\sigma_s = \{2.9, 3.6, 4.2\}\,\mathrm{m}^{-1}$, $\sigma_a = \{3.4, 3.35, 3.4\}\,\mathrm{m}^{-1}$ and $g = 0.9$ using the "Uffizi" environment map as illumination. Here, POP used 64 and 125 principal ordinates, grid resolutions of 20^3 and 50^3, with 10 and 30 propagation iterations, respectively. SHDOM used 5 and 10 SH bands to represent the directional radiance distribution in each cell and the same grid resolutions. SHDOM required a strong prefiltering to avoid the typical *ringing* issues and due to false scattering it fails to reproduce the high scattering anisotropy. Our method compares well to the reference solution, matching the overall appearance well even with the lower (interactive) settings.

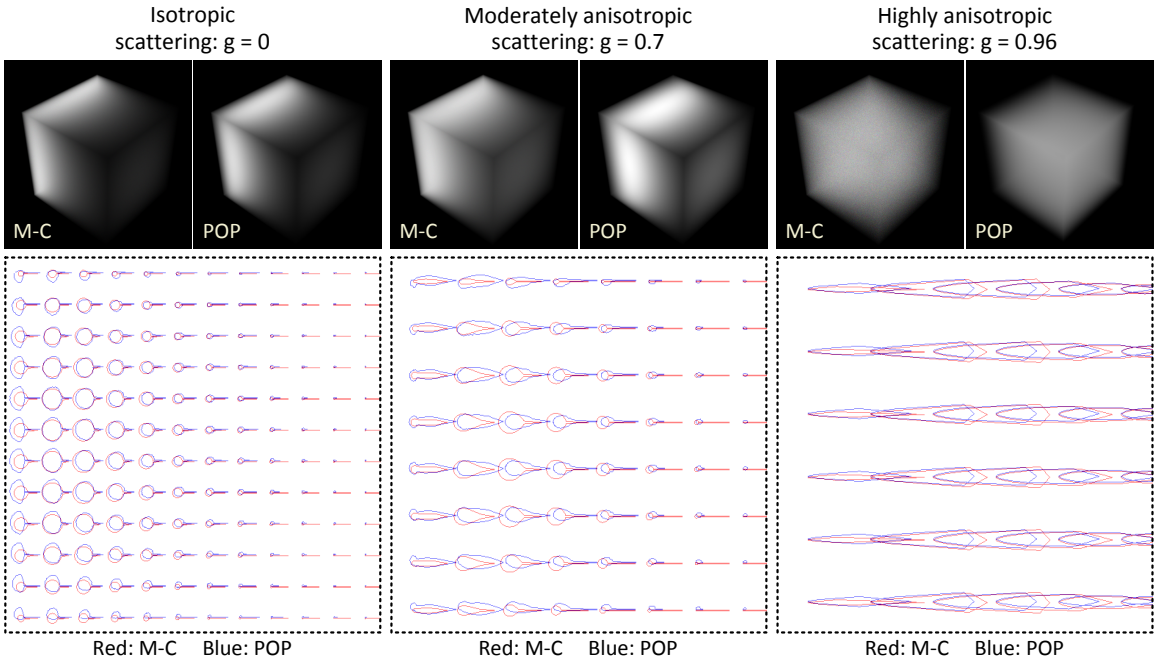


Figure 5.13: Detailed analysis of POP in comparison to Monte-Carlo reference (light tracing). Here a sample single-ordinate scenario is shown, with a 16^3 propagation domain aligned with the medium (homogeneous cube 1 m^3, $\sigma_s = 4\,m^{-1}$ and unit albedo), using three representative anisotropy values. The plots compare the converged radiance distributions within a 2D horizontal slice in the middle of the domain.

general we found that amounts similar to the propagation grid resolution along the propagation dimension is sufficient (around 10–20 in most examples). Please note that the environment maps are blurred only for presentation purposes (so that the medium lighting features can be examined better) – the actual illumination is in fact sampled from the full-resolution maps. Other specific scene details are provided in the caption of each discussed figure.

Reference comparisons We first compare POP to an unbiased Monte-Carlo reference (light tracing, Section 2.5.1), as well as spherical harmonics (SH) DOM (Section 2.5.2), in Figure 5.12. It is apparent that the described artifacts prevent SHDOM from handling anisotropic media correctly, despite being theoretically capable to do so. In contrast POP, despite being biased, reproduces the qualitative appearance well.

A simpler analysis for a single directional propagation is presented as well. Figure 5.13 shows comparisons to the reference for a single ordinate, propagating in a simple cubic medium. Again, despite some differences in the appearance, we can see the directional radiance distributions match well. We note that this is actually a difficult case for POP, because the medium regularity and the high density gradient on the faces parallel to the propagation direction violate the alignment assumption, as discussed in Section 5.3.5. However, observe that the directional distributions produced by POP have more complex shapes that the simple Henyey-Greenstein ellipsoid lobes, since they are constituted by a superposition of many such lobes (one per propagation iteration, M in total). In addition, Figure 5.10 shows a simplified analysis similar to this, but for a point light source and a radial propagation grid.

Propagation behaviour The convergence of POP is examined in Figure 5.14. The setting is identical to the second case in Figure 5.13. Notice that because of the absence of absorption the

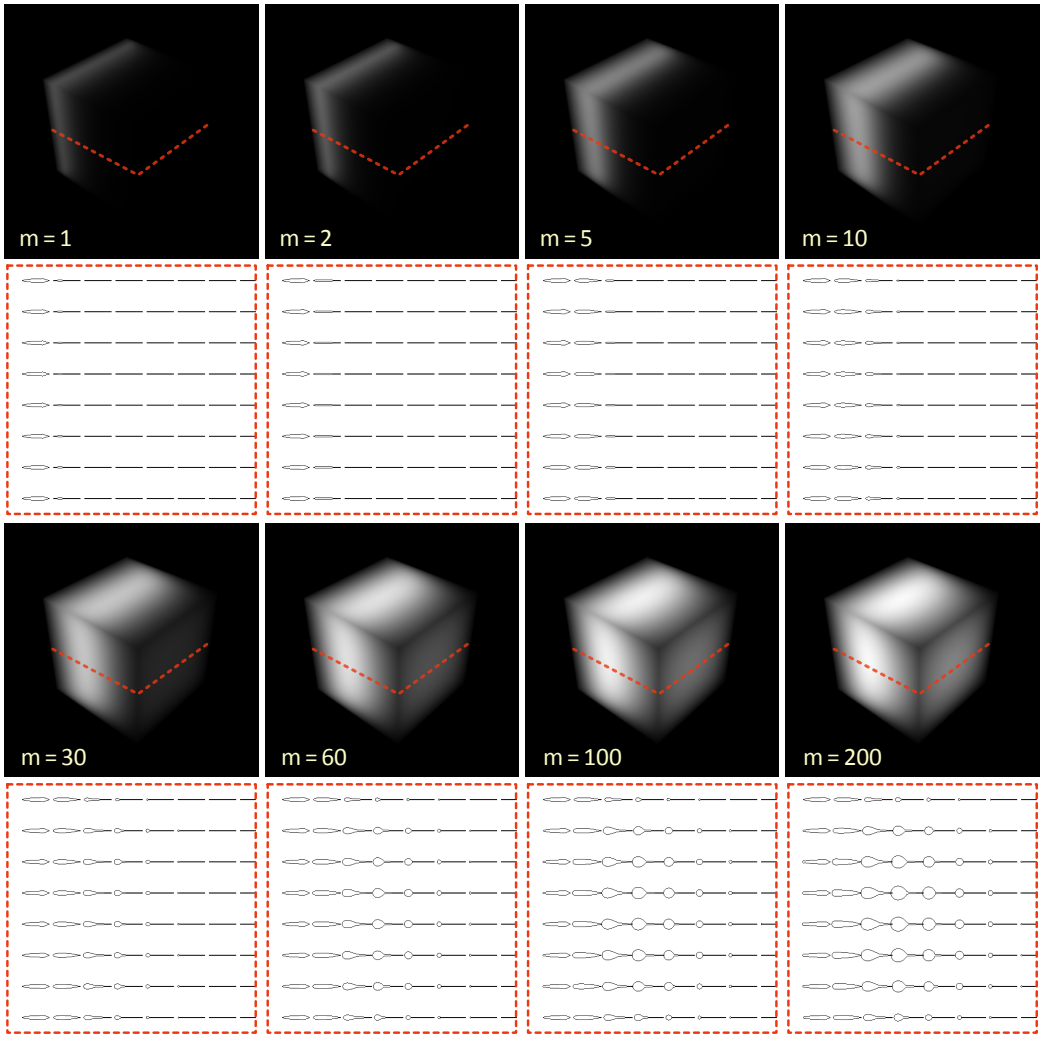


Figure 5.14: Convergence of the propagation scheme for the setting described in Figure 5.13, with scattering anisotropy $g = 0.7$. The plots show the respective incident radiance distributions within a 2D horizontal slice in the middle of the domain (marked by the red dashed line). The observed strong forward peaks represent the *unscattered* energy which did not (yet) interact with the medium.

propagation takes a significant number of iterations, even for the small 16^3 grid. That is the main motivation for introducing the isotropic residuum propagation (Section 5.4.4).

The effect of using different numbers of principal ordinates is shown in Figure 5.15. It can be seen that the discretisation becomes apparent only with very few ordinates. The importance propagation usually helps to alleviate this by sampling those directions which will influence the solution most significantly. As Figure 5.8 demonstrates, this is most likely the opposite side of the medium, suggesting that a simpler empirical heuristic could potentially work in certain cases.

One of the main shortcomings of the importance propagation is its potential temporal incoherence, primarily manifested by temporal flickering. For this reason we filter the importance map both spatially and temporarily, which, however, is not a fully robust solution to the issue. One of our main targets for future work is therefore improving this by explicitly enforcing temporal coherence when the sampled light sources relocate due to camera movement or illumination changes, similarly to, e. g., Prutkin et al. [2012] (see further discussion in Section 7.1.3).

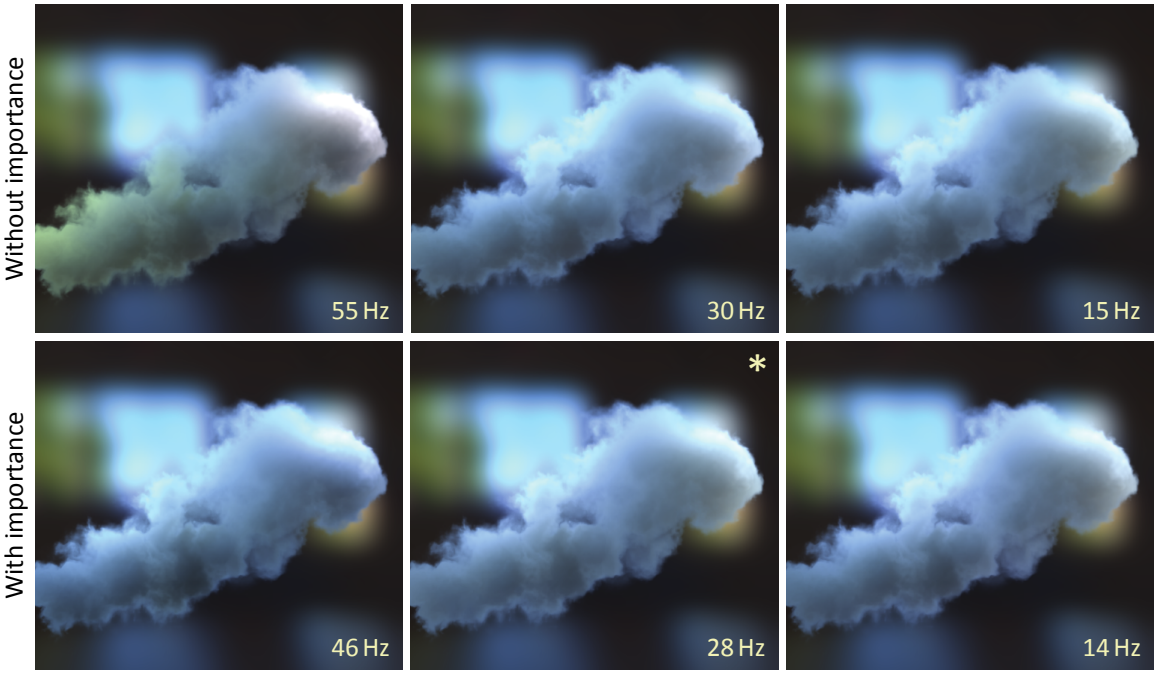


Figure 5.15: The smoke dataset with an increasing number of ordinates using the "kitchen" environment map ($g = 0.9$, 20^3 grid resolution, 10 propagation iterations). Accounting for importance improves the results, mainly if low numbers of principal ordinates are used. The typical configuration we use (in the *bottom-centre*, asterisk-marked) takes 5 ms for importance propagation, 2 ms for determining the ordinates, 2 ms for grid initialisation, 12 ms for propagation, 2 ms for residuum propagation, 4 ms for grid merging and upsampling and 5 ms for ray-marching.

Prefiltering (Section 5.4.1) helps to improve the rendering quality in most scenarios and has been used to generate all the presented results. It is particularly indispensable for media with an optical thickness too low to sufficiently 'redistribute' the sampled illumination, as for instance in Figure 5.7, where weak singularity artefacts would appear otherwise. The prefiltering removes these artifacts but still allows perceiving some features of the background illumination, thanks to its adaptivity (as opposed to a naïve 'flat' prefiltering of the source illumination).

Scattering anisotropy The shortcomings of current methods in handling highly anisotropic scattering were the main motivation for this work, since by far the majority of both natural and artificial media exhibit anisotropic scattering (cf. [Narasimhan et al. 2006]).

Akin to Chapter 4 we tested POP for clouds, which are media with naturally very high scattering anisotropy, and compared to their isotropic versions (Figure 5.16). It can be seen that the proposed propagation scheme handles both cases well, and that correctly handling anisotropic scattering is a key to reproducing such media. The same can be observed in Figure 5.1, since steam has properties similar to clouds. Interestingly, grid resolutions as well as computation times required to render plausible participating media are rather insensitive to their anisotropy, i. e., anisotropic media render roughly as fast as isotropic media. Although a larger number of ordinates might be required to reproduce high-anisotropy effects, this additional effort is usually compensated by a decreased complexity of the spatial radiance distribution, which enables using coarser propagation grids.

The dual propagation scheme (i. e., including the separate step to propagate residual isotropic energy)

Figure 5.16: Clouds rendered at 35 Hz using 64 ordinates and 20^3 grid resolution for each of them, with 15 propagation iterations. In media like clouds the scattering anisotropy plays a crucial role in forming their appearance, thus the common assumption of isotropic scattering *(left)* prevents a believable rendition of such media. The scattering anisotropy for the *right* column has been $g = 0.96$, otherwise all settings and medium parameters were kept identical.

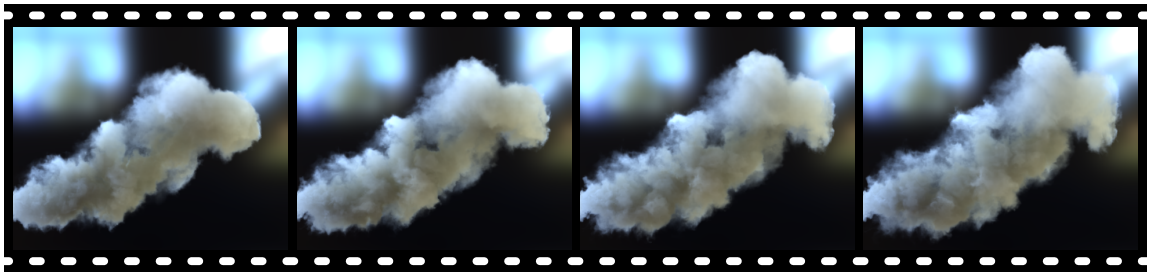


Figure 5.17: Animated sequence of an expanding smoke plume, rendered dynamically at 30 Hz by the presented method, while maintaining good temporal coherence.

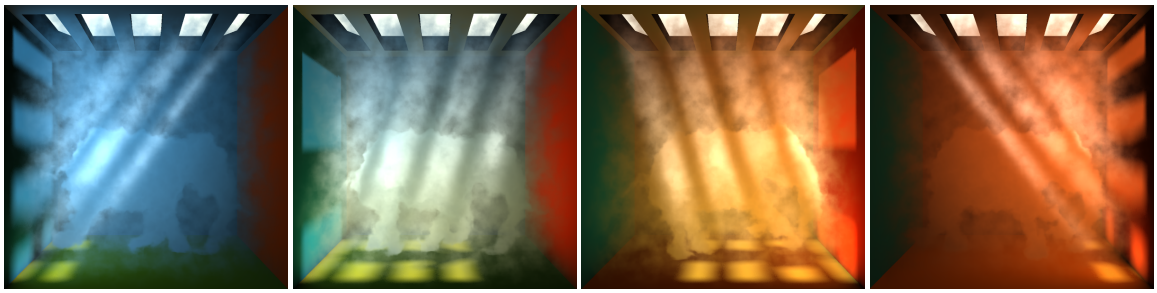


Figure 5.18: Scene with both animated medium and illumination, combining scattering from directional and local virtual light sources, running at 16 Hz (including the generation of the 125 VPLs and rendering the indirect illumination from surfaces, which alone takes 30 ms). The medium has a size of 20^3 m with $\sigma_s = \{3.2, 3.3, 3.4\}\,\mathrm{m}^{-1}$, $\sigma_a = \{1.15, 1.2, 1.3\}\,\mathrm{m}^{-1}$, $g = 0.7$, and is very heterogeneous. The grid size for the directional light is $128^2 \times 16$, with the 16-cell axis oriented along the light shafts (i. e., along the principal ordinate). The radial grids have a resolution of 8^3 each. These settings were used for local light sources in all presented examples; note how even this small resolution proves to be sufficient for plausible results.

also efficiently handles optically thick anisotropic media, as seen in Figure 5.9. The initial, full propagation handles the directionally-dependent portion of radiant energy, while the remaining isotropic residuum is rapidly propagated in the second stage.

Animation Thanks to the fully dynamic nature of the algorithm it can seamlessly handle animated media without any precomputations or performance penalty. Figure 5.17 shows several frames of an animated smoke plume coherently rendered at real-time framerates. In Figure 5.18 the dynamic interaction of POP with surfaces is demonstrated, as described in Section 5.5.

In general, we believe to have demonstrated the versatility of the method. The propagation along principal ordinates is capable of computing direct illumination and low-order scattering effects (light shafts), as well as arbitrary multiple scattering from directional, local and environment illumination. Orthogonally to this, POP is capable of simulating media with wide ranges of optical thickness, albedo and most importantly, scattering anisotropy. For an extended discussion and outlooks, please refer to Section 7.1.3.

Chapter 6

Spectral Ray Differentials

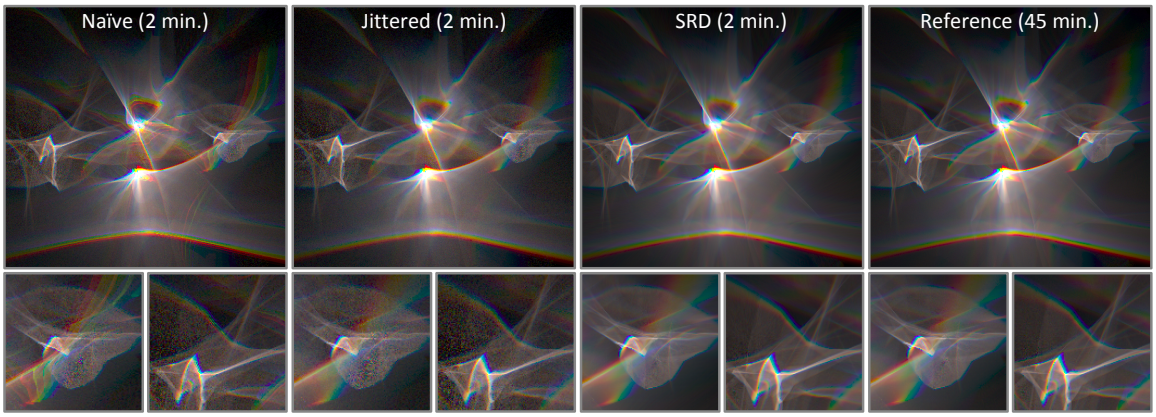


Figure 6.1: Complex dispersive caustic rendered with spectral light tracing using 7 spectral bands. The naïve solution exhibits strong aliasing artifacts due to regular sampling of the spectrum; stochastically jittering the spectral samples solves this problem, but the solution still contains significant chromatic noise. The proposed spectral ray differentials (SRD) allow an efficient reconstruction of the solution, yielding images visually identical to the reference (rendered with 35 spectral bands and 4 times more samples) at a fraction of time.

6.1 Introduction

When light refracts at a dispersive interface (such as a glass prism) its new direction is wavelength-dependent, resulting in colourful rainbow-like patterns (details are provided in Section 2.6). This is both true for caustics where light disperses before it hits a diffusive surface, as well as for objects when directly seen through dispersive interfaces (e. g., a low-quality lens system), exhibiting chromatic aberrations. Creating light or eye paths for rendering in the presence of dispersion then implies that a single unique wavelength has to be used for every path as different wavelengths take different paths. This results either in spectral under-sampling (banding) or chromatic noise, as demonstrated in Figure 6.1.

To address this problem, this chapter introduces the concept of *spectral ray differentials* (SRD). The

common ray, path and photon differentials [Igehy 1999; Suykens and Willems 2001; Schjøth et al. 2007] describe the wavelength-independent change of a ray, path or photon in the spatial domain with respect to the change of the initial coordinates in the respective domain where they originate from (see Section 2.7). The domain is defined by the sensor plane for rays and paths, or by the light source coordinates for photons. This information is then used primarily for improved texture filtering and photon density estimation. In contrast, spectral ray differentials analytically model the spatial change of a ray when interacting with dispersive interfaces with regard to change of wavelength. Intuitively, they describe how light transport changes, when wavelength-dependent interactions occur along an optical path.

Instead of making assumptions about the scene and trying to find approximate frequency bounds (as many contemporary reconstruction methods discussed in Section 2.7), SRD rely on tracing actual rays and their exact analytic derivatives, which makes them easily incorporated into existing stochastic rendering methods (Section 2.5.1). The application of this information is then based on a first-order Taylor expansion and is dependent how a particular rendering algorithm connects optical paths to the sensor (Figure 6.2). Typically, however, the resulting reconstruction can be as simple as a custom wavelength-dependent linear filter, either applied by a convolution in image space or splatting from world space (see Section 6.5).

One shortcoming of the first-order approximation is the bias introduced into the solution, due to the pre-determined finite size of the reconstruction region. In order to resolve this issue, the proposed method is formulated in a progressive manner (cf. Section 2.5.1). The basic idea is analogous to progressive photon mapping methods [Knaus and Zwicker 2011; Jarosz et al. 2011b] – to carefully decrease the reconstruction scale of SRD with increasing number of illumination samples that are traced though the scene. This allows SRD to consistently converge to a corresponding unbiased Monte-Carlo solution.

SRD can be used for reconstructing wavelength-dependent phenomena in offline as well as interactive rendering, which is demonstrated in Section 6.6 for dispersive ray and light tracing, as well as interactive GPU photon mapping. The improved reconstruction resulting from the application of SRD allows tracing and storing significantly less samples to obtain smooth solutions. Additional examples are presented of how spectral differentials open up new possibilities besides rendering, such as interactive local and non-physical artistic control over dispersion effects.

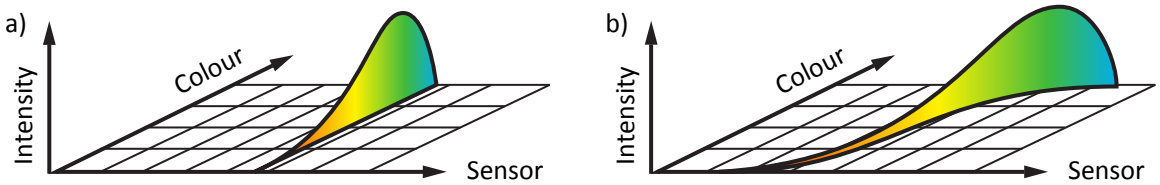


Figure 6.2: Interaction of spectral light sample colour and intensity as well as an abstracted 1D sensor in a classic *(a)* and spectrally dependent *(b)* image formation model. Commonly, the samples contribute all wavelengths to the same sensor location (alignment with the colour axis). In our case, samples at different wavelengths contribute to different sensor locations. This function can have a complicated shape (indicated by a curve), for which SRD use a first-order, piecewise linear approximation.

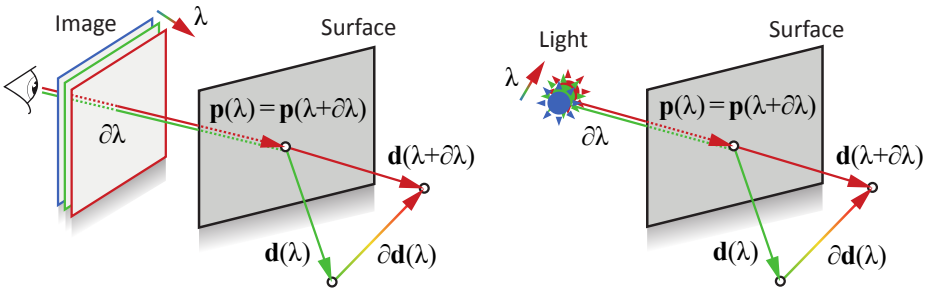


Figure 6.3: Concept and notation of spectral ray differentials, for eye paths *(left)* and light paths *(right)*.

6.2 Spectral Ray Differentials

The key observation is that in the presence of dispersion the positions and directions of optical paths (hereinafter simply referred to as rays) that start at the sensor are not only functions of sensor position but also of wavelength. This situation is naturally symmetric for paths starting at light sources. Therefore, for a ray $\mathbf{R}(\lambda) = (\mathbf{p}, \mathbf{d})$ associated with wavelength λ, we want to compute the partial differentials $\frac{\partial \mathbf{p}}{\partial \lambda}$ and $\frac{\partial \mathbf{d}}{\partial \lambda}$.

Initially, if the ray starts at the light or the sensor, its spectral differentials are zero (in contrast to classic ray differentials). When the ray is traced through the scene, the differential is updated for all transport, reflection and refraction events. For transport and reflection (which are wavelength-independent) the formulas derived by Igehy [1999] can be used directly by substituting λ as the differential variable instead of the sensor coordinates (Figure 2.11 in Section 2.7). However, for a dispersive refraction event (Figure 6.3) the new direction of the ray now changes with the wavelength-dependent refractive index ratio η (Equations 2.37 and 2.38).

Akin to the classic ray differentials, spectral ray differentials w. r. t. the associated wavelength λ are the following pair of partial derivatives

$$\frac{\partial \mathbf{R}}{\partial \lambda} = \left(\frac{\partial \mathbf{p}}{\partial \lambda}, \frac{\partial \mathbf{d}}{\partial \lambda} \right), \tag{6.1}$$

which are updated after an interaction with a dispersive interface as

$$\frac{\partial \mathbf{p}'}{\partial \lambda} = \frac{\partial \mathbf{p}}{\partial \lambda} \qquad \text{and} \qquad \frac{\partial \mathbf{d}'}{\partial \lambda} = \frac{\partial (\eta \mathbf{d} - \nu \mathbf{n})}{\partial \lambda}. \tag{6.2}$$

Here, $\mathbf{n}$ is the interface normal, η depends on the used refractive index model (Section 6.2.2) and ν is defined in Equation 2.38. Note that $\mathbf{p}$, $\mathbf{d}$, $\mathbf{n}$, η and ν are generally all functions of λ. However, since refraction does not change the *positional* differential, we are only interested in finding the *directional* spectral differential as described below.

6.2.1 Directional Spectral Differential

For better readability, the cosine of the angle between the incoming ray direction and surface normal will be denoted as $\vartheta = \mathbf{d} \cdot \mathbf{n}$, where '$\cdot$' will be used as a shorthand for dot product '$\langle\rangle$'. Using the product rule applied to the two vectors, its partial derivative $\frac{\partial \vartheta}{\partial \lambda}$ is computed by

$$\frac{\partial \vartheta}{\partial \lambda} = \mathbf{n} \cdot \frac{\partial \mathbf{d}}{\partial \lambda} + \mathbf{d} \cdot \frac{\partial \mathbf{n}}{\partial \lambda}, \tag{6.3}$$

noting that the differentials of $\mathbf{d}$ and $\mathbf{n}$ are vectors. The new spectral directional differential after refraction is then given by

$$\boxed{\frac{\partial \mathbf{d}'}{\partial \lambda} = \mathbf{d}\frac{\partial \eta}{\partial \lambda} + \eta\frac{\partial \mathbf{d}}{\partial \lambda} - \mathbf{n}\frac{\partial \nu}{\partial \lambda} - \nu\frac{\partial \mathbf{n}}{\partial \lambda}} \tag{6.4}$$

with

$$\frac{\partial \nu}{\partial \lambda} = \vartheta\frac{\partial \eta}{\partial \lambda} + \eta\frac{\partial \vartheta}{\partial \lambda} + \frac{\eta(\vartheta^2-1)\frac{\partial \eta}{\partial \lambda} + \eta^2\vartheta\frac{\partial \vartheta}{\partial \lambda}}{\sqrt{1-\eta^2+\eta^2\vartheta^2}}. \tag{6.5}$$

Equation 6.4 is the main theoretical contribution in this chapter, with derivation details provided below.

Note that the formulation of the normal differential $\frac{\partial \mathbf{n}}{\partial \lambda}$ by Igehy is still valid for spectral differentials because it is solely based on the surface properties and the positional differential w. r. t. a given variable. In Igehy's case this is the screen coordinate x or y and here it is the wavelength λ.

The refractive-index ratio η is now dependent on the wavelength. It is defined by $\eta = \frac{\mathrm{n}(\lambda)}{\mathrm{n}'(\lambda)}$ where $\mathrm{n}(\lambda)$ and $\mathrm{n}'(\lambda)$ are the index of refraction (IOR) of the medium through which the ray travelled and the IOR of the medium that the newly generated ray will enter (this is illustrated in Figure 2.10, right). Its derivative $\frac{\partial \eta}{\partial \lambda}$ depends on the model used to obtain η itself, and for the most frequently utilised models is derived later in Section 6.2.2.

Derivation Rephrasing the Snell's law (Equation 2.38), the refracted direction of the incident ray $\mathbf{R} = (\mathbf{p}, \mathbf{d})$ with respect to index of refraction η, and ϑ defined above, is given by

$$\begin{aligned} \mathbf{d}' &= \eta\mathbf{d} - \nu\mathbf{n} \quad \text{with} \\ \nu = \eta\vartheta + \beta \quad &\text{and} \quad \beta = \sqrt{1-\eta^2+\eta^2\vartheta^2}. \end{aligned} \tag{6.6}$$

Now, we seek the partial differential of $\mathbf{d}'$ with respect to λ:

$$\frac{\partial \mathbf{d}'}{\partial \lambda} = \frac{\partial(\eta\mathbf{d})}{\partial \lambda} - \frac{\partial(\nu\mathbf{n})}{\partial \lambda}$$

Solving each term separately by applying the product rule, we get Equation 6.4:

$$\frac{\partial \mathbf{d}'}{\partial \lambda} = \mathbf{d}\frac{\partial \eta}{\partial \lambda} + \eta\frac{\partial \mathbf{d}}{\partial \lambda} - \left(\mathbf{n}\frac{\partial \nu}{\partial \lambda} + \nu\frac{\partial \mathbf{n}}{\partial \lambda}\right)$$

Out of the above terms, the remaining one to calculate is the partial differential $\frac{\partial \nu}{\partial \lambda}$:

$$\frac{\partial \nu}{\partial \lambda} = \frac{\partial(\eta\vartheta)}{\partial \lambda} + \frac{\partial \beta}{\partial \lambda} = \vartheta\frac{\partial \eta}{\partial \lambda} + \eta\frac{\partial \vartheta}{\partial \lambda} + \frac{\partial \beta}{\partial \lambda}$$

Finally, the partial differential $\frac{\partial \beta}{\partial \lambda}$ is computed using the chain rule:

$$\frac{\partial \beta}{\partial \lambda} = \frac{1}{2}(1-\eta^2+\eta^2\vartheta^2)^{-\frac{1}{2}}\frac{\partial(1-\eta^2+\eta^2\vartheta^2)}{\partial \lambda} = \frac{1}{2\beta}\frac{\partial(1-\eta^2+\eta^2\vartheta^2)}{\partial \lambda}$$

$$\frac{\partial(1-\eta^2+\eta^2\vartheta^2)}{\partial \lambda} = -2\eta\frac{\partial \eta}{\partial \lambda} + \frac{\partial(\eta^2\vartheta^2)}{\partial \lambda}$$

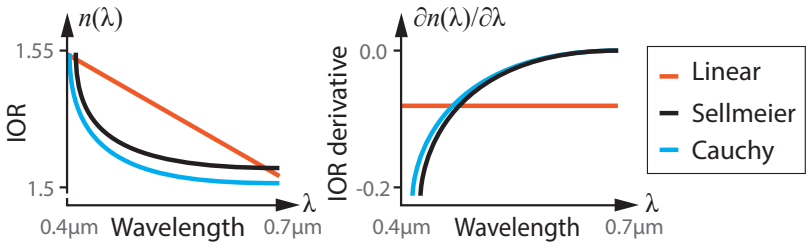


Figure 6.4: Dependency of the index of refraction (IOR) on wavelength for a typical material (glass) using different models, and their derivatives.

$$\frac{\partial(\eta^2\vartheta^2)}{\partial\lambda} = 2\eta\vartheta\left(\vartheta\frac{\partial\eta}{\partial\lambda}+\eta\frac{\partial\vartheta}{\partial\lambda}\right) = 2\eta\vartheta^2\frac{\partial\eta}{\partial\lambda}+2\eta^2\vartheta\frac{\partial\vartheta}{\partial\lambda}$$

$$\frac{\partial\beta}{\partial\lambda} = \frac{-2\eta\frac{\partial\eta}{\partial\lambda}+2\eta\vartheta^2\frac{\partial\eta}{\partial\lambda}+2\eta^2\vartheta\frac{\partial\vartheta}{\partial\lambda}}{2\beta} = \frac{-\eta\frac{\partial\eta}{\partial\lambda}+\eta\vartheta^2\frac{\partial\eta}{\partial\lambda}+\eta^2\vartheta\frac{\partial\vartheta}{\partial\lambda}}{\beta}$$

Putting everything together, we arrive to Equation 6.5:

$$\frac{\partial\nu}{\partial\lambda} = \vartheta\frac{\partial\eta}{\partial\lambda}+\eta\frac{\partial\vartheta}{\partial\lambda}+\frac{-\eta\frac{\partial\eta}{\partial\lambda}+\eta\vartheta^2\frac{\partial\eta}{\partial\lambda}+\eta^2\vartheta\frac{\partial\vartheta}{\partial\lambda}}{\beta}.$$

6.2.2 Differential of η

The function η is defined as the ratio between the index of refraction for the original medium $n(\lambda)$ and the entered medium $n'(\lambda)$. The differential of η is then computed by

$$\frac{\partial\eta}{\partial\lambda} = \frac{n'(\lambda)\frac{\partial n(\lambda)}{\partial\lambda}-n(\lambda)\frac{\partial n'(\lambda)}{\partial\lambda}}{n'(\lambda)^2}. \tag{6.7}$$

The function $n(\lambda)$ models the dependence between the wavelength (in vacuum) and the index of refraction for a certain material. It can be measured, but is often approximated by physically-based models, e. g., the Cauchy's and Sellmeier's formulas [Jenkins and White 2001], or even by a simple linear function (Figure 6.4), in which case the derivative of $n(\lambda)$ is constant.

Cauchy's model The formula of Cauchy is defined as

$$n_{\text{Cauchy}}(\lambda) = B+\frac{C}{\lambda^2}+\frac{D}{\lambda^4}+\ldots, \tag{6.8}$$

where B, C, D and so on are empirically measured, material-specific coefficients. Its derivative w. r. t. wavelength is given by

$$\frac{\partial n_{\text{Cauchy}}(\lambda)}{\partial\lambda} = -\frac{2C}{\lambda^3}-\frac{4D}{\lambda^5}-\ldots \tag{6.9}$$

Sellmeier's model Similarly, the generalised Sellmeier's formula and its derivative are

$$n_{\text{Sellmeier}}(\lambda) = \sqrt{1+\sum_k\frac{B_k\lambda^2}{\lambda^2-C_k}} \tag{6.10}$$

$$\frac{\partial n_{\text{Sellmeier}}(\lambda)}{\partial\lambda} = -\frac{\sum_k\frac{B_kC_k}{(\lambda^2-C_k)^2}}{\sqrt{1+\sum_k\frac{B_k\lambda^2}{\lambda^2-C_k}}}. \tag{6.11}$$

6.3 Tracing Spectral Differentials

Unbiased spectral rendering requires tracing monochromatic rays which define a unique path through the scene. To use spectral ray differentials when reconstructing a spectral image, the differentials need to be tracked and updated along the ray at all interactions that occur, both for camera and light rays. In any case, each ray $(\mathbf{p}, \mathbf{d}, \lambda)$ is given by two 3D vectors (position and direction) and a wavelength, and its spectral ray differential $(\frac{\partial \mathbf{p}}{\partial \lambda}, \frac{\partial \mathbf{d}}{\partial \lambda})$ is described by two additional 3D vectors (positional and directional differential).

Note that the spectral differential becomes non-zero only when a dispersive interface is hit along a path. It is also important that while dispersion is the only event that causes spectral differentials to become non-zero (and does so only in the directional component), the classic transport and reflection differentiation still needs to be applied. Hence, both the positional and directional differentials have to be maintained during the transport.

In practice, at every non-dispersive event (transport and reflection) the spectral differentials are updated using the classic equations from Igehy [1999]. On the other hand, at every dispersive event, Equation 6.4 is used to update the directional differential. Listing 6.1 shows the pseudo-code for updating the current differential for a refractive event. It is apparent that the computational overhead is small and only a relatively low number of arithmetic operations needs to be performed, that is, compared to ray-scene intersections and other tasks that are executed with comparable frequency.

Listing 6.1: C-style pseudocode for the directional spectral ray differential update (Equation 6.4).

```
vec3 updateDirectionalDifferential(vec3 p, vec3 ∂p, vec3 d, vec3 ∂d, vec3 n, float η)
{
    vec3 ∂n = normalDifferential(∂p);
    float ∂η = etaDifferential(η);

    float ϑ = dot(d, n);
    float β = sqrt(1 - sqr(η) + sqr(η)*sqr(ϑ));
    float ν = η*ϑ + β;

    float ∂ϑ = dot(∂d, n) + dot(d, ∂n);
    float ∂β = (-η*∂η + η*sqr(ϑ)*∂η + sqr(η)*ϑ*∂ϑ) / β;
    float ∂ν = ϑ*∂η + η*∂ϑ + ∂β;

    vec3 ∂d' = d*∂η + η*∂d - n*∂ν - ν*∂n;
    return ∂d';
}
```

6.4 Reconstruction

SRD can be used for image filtering and reconstruction in different settings, such as dispersive light tracing (Section 6.5.1), eye tracing (Section 6.5.2) or photon mapping (Section 6.5.3). As described in Section 6.3, tracing a spectral differential amounts to maintaining a pair of 3D vectors $(\frac{\partial \mathbf{p}}{\partial \lambda}, \frac{\partial \mathbf{d}}{\partial \lambda})$ associated with the ray $\mathbf{R}(\lambda) = (\mathbf{p}, \mathbf{d})$. The reconstruction then uses the differential to make predictions about the spatial ray changes in dependence on changes in its spectral coordinate λ by a small differential offset $\Delta\lambda$. In line with classic ray differentials [Igehy 1999], this spatial

displacement is obtained by applying the first-order Taylor approximation:

$$\mathbf{R}(\lambda+\Delta\lambda)-\mathbf{R}(\lambda)\approx\Delta\lambda\frac{\partial\mathbf{R}(\lambda)}{\partial\lambda}=:\Delta\mathbf{R} \tag{6.12}$$

This defines a linear 1D footprint of the sample, with the spatial extent between $(\mathbf{p}-\Delta\mathbf{p})$ and $(\mathbf{p}+\Delta\mathbf{p})$. Splatting of line primitives is one (and the simplest) way to reconstruct such samples.

The one-dimensionality of the footprint might seem surprising on the first sight (since the footprint of classic differentials is two-dimensional), but mathematically it is a consequence of the spectral domain's dimensionality. Intuitively, if a polychromatic pencil of light disperses on a (smooth) planar interface, the resulting footprint will indeed be a straight line, where each point receives light of only a single wavelength. Successive dispersive interactions can distort this footprint and even introduce discontinuities (Figures 6.5 and 6.6), but cannot increase its dimensionality.

An important observation is that since Equation 6.12 only provides a prediction of the true footprint, it carries an inherent bias. The reconstruction size $\Delta\lambda$ acts as a user-controlled parameter here. Increasing $\Delta\lambda$ will enhance the variance reduction of SRD (because each sample will contribute to a larger area) but also increase bias (since the prediction scale is extended). And, vice-versa, if $\Delta\lambda\to 0$ then the sample footprint shrinks, down to unbiased point estimates at $\Delta\lambda=0$. Performing this shrinking automatically and in a progressive manner is the key ingredient to enable SRD-based rendering that consistently converges to the true solution, similarly to the progressive formulations of photon mapping (discussed in Section 2.5.1) on which the following theory builds.

Estimate Regardless of the method-specific reconstruction procedure, the radiance L in the direction $\mathbf{d}_x$ within the sample footprint can be estimated as

$$L(\mathbf{p}_x,\mathbf{d}_x,\Delta\lambda)\approx k_{\Delta\lambda}(u)\cdot\gamma, \tag{6.13}$$

where $\mathbf{p}_x$ lies on the line between $(\mathbf{p}-\Delta\mathbf{p})$ and $(\mathbf{p}+\Delta\mathbf{p})$. Equation 6.13 expresses the reconstructed radiance as a kernel estimate from the sample contribution γ (which combines the flux carried by the sample modulated by the appropriate surface BRDF and geometry terms). The kernel $k_{\Delta\lambda}(u)$ with the support size of $\Delta\lambda$ depends on the distance $u=\|\mathbf{p}-\mathbf{p}_x\|$. Photon mapping traditionally uses the Gaussian kernel, although in the context of SRD a simpler box kernel $k_{\Delta\lambda}(u)=\frac{H(u+\Delta\lambda)-H(u-\Delta\lambda)}{2\cdot\Delta\lambda}$ proves sufficient.

The error ϵ_L of the estimate provided by Equation 6.13 can be stated explicitly as the difference between the estimated and the true radiance (at a given position and direction):

$$L(\mathbf{p}_x,\mathbf{d}_x)=L(\mathbf{p}_x,\mathbf{d}_x,\Delta\lambda)-\epsilon_L(\Delta\lambda). \tag{6.14}$$

Analysing the statistical behaviour of ϵ_L in dependence on $\Delta\lambda$ is the key step to derive the progressive update of $\Delta\lambda$.

Error analysis Thanks to the formulation of the estimate and error in Equations 6.13 and 6.14 we can now directly apply the results of the one-dimensional probabilistic framework of Jarosz et al. [2011b]. The primary result of their analysis are estimates for the expected value and the variance of $\epsilon_L(\Delta\lambda)$:

$$\mathrm{E}[\epsilon_L(\Delta\lambda)]\approx\Delta\lambda\cdot\mathrm{E}[\gamma]\cdot C_1, \tag{6.15}$$

$$\mathrm{Var}[\epsilon_L(\Delta\lambda)]\approx\frac{(\mathrm{Var}[\gamma]+\mathrm{E}[\gamma]^2)\cdot C_2}{\Delta\lambda}. \tag{6.16}$$

Here, C_1 and C_2 are constants that depend on the specific kernel shape and sampling strategy. These estimates are derived for reconstructing a single sample at a given query location, but it can be shown that they also generalise to a combined reconstruction from multiple samples with different size.

The most important conclusion to draw from Equations 6.15 and 6.16 is the linear relationship between the reconstruction scale $\Delta\lambda$ and the resulting error characteristics. Specifically if $\Delta\lambda$ *decreases* than the expected error will *decrease linearly* while its variance will *increase linearly*. The first relationship is necessary for the progressive solution to consistently converge, while the increasing variance will be compensated for by the increasing overall number of samples constituting the solution. As will be shown next the proper balance between these two relationships is achieved when $\Delta\lambda$ decreases *sub-linearly*.

Progressive update of $\Delta\lambda$ Under the conditions described previously, Knaus and Zwicker [2011] show that a progressive convergence can be achieved by imposing the following ratio of variance between successive passes:

$$\frac{\mathrm{Var}[\epsilon_{i+1}]}{\mathrm{Var}[\epsilon_i]} = \frac{i+1}{i+\alpha}, \tag{6.17}$$

where $\alpha \in [0,1]$ is a user parameter controlling how steep the progression should be. Due to the inverse proportion between the variance and the reconstruction scale $\Delta\lambda$ (Equation 6.16) we get an update rule for the reconstruction scale at pass i:

$$\frac{\Delta\lambda_{i+1}}{\Delta\lambda_i} = \frac{\mathrm{Var}[\epsilon_i]}{\mathrm{Var}[\epsilon_{i+1}]} = \frac{i+\alpha}{i+1}. \tag{6.18}$$

The intuition behind the parameter α is quite simple: for α close to 1 the reconstruction scale will decrease slowly, hence leading to only a slow variance growth. This implies that the solution becomes smooth quickly, but at the cost of only a slow decrease of bias. The opposite is true for small α.

Note that Equation 6.18 is independent on the number of samples traced per pass (i. e., per single batch of samples). This is obviously undesired, as different number of samples per pass will yield different results. Jarosz et al. proposed the addition of a parameter M (denoting the number of samples per pass) that compensates for this by applying the reconstruction scale update M-times after each pass (note however that the original equation in Jarosz et al. [2011b] contains a typo [Jarosz 2014]):

$$\frac{\Delta\lambda_{i+1}}{\Delta\lambda_i} = \prod_{j=1}^{M} \frac{(i-1)M+j+\alpha}{(i-1)M+j+1} \tag{6.19}$$

This way the parameter α becomes independent of the pass size and corresponds to changing $\Delta\lambda$ for every individual sample, albeit with a delayed application at the end of each pass. This is especially beneficial in GPU implementations where tracing small sample batches and frequent $\Delta\lambda$ updates would heavily under-utilise the available computational resources.

In addition, the recurrent relationship in Equation 6.19 can be reformulated to compute $\Delta\lambda_i$ directly:

$$\Delta\lambda_i = \Delta\lambda_1 \cdot \frac{1}{iM} \prod_{k=1}^{iM-1} \frac{k+\alpha}{k} \tag{6.20}$$

However, this is less efficient than Equation 6.19 as the number of evaluated operations grows with each successive pass.

6.5 Application

Tracing ray differentials (both classic and spectral) is independent on the specific Monte-Carlo method, and is described in detail Section 6.3. Different rendering algorithms then apply this information depending on their specific illumination reconstruction methods, using the methodology from Section 6.4.

6.5.1 Light Tracing

In light tracing, rays starting at the light are traced through the scene and eventually connected to the sensor (Section 2.5.1). All paths that contain a dispersive interface can result in a non-zero spectral differential. To utilise SRD for reconstruction, two differential-guided options are available: splatting of linear light samples, or a gathering-type image-space filtering approach.

Splatting This approach directly 'splats' the spectral positional differential (a world space offset) of every light path onto the sensor by drawing a line from $\rho(\mathbf{p}-\Delta\mathbf{p})$ to $\rho(\mathbf{p}+\Delta\mathbf{p})$, where $\rho \in \mathbb{R}^3 \to \mathbb{R}^2$ is a projective mapping, $\mathbf{p}$ is the ray's hit position, and $\Delta\mathbf{p}$ and $\Delta\lambda$ are the spatial and spectral neighbourhoods obtained according to Section 6.4.

The actual colours for each fragment (i. e., projected position) of the line are computed by Equation 6.13, with λ extrapolated in the spectrum based on the wavelengths $\lambda \pm \Delta\lambda$ that correspond to the start and end point. The spectral values are first converted to the XYZ colour space using the CIE matching functions [CIE 1932] and then into the RGB space of the output device. For efficiency a 1D lookup table is used to represent the $\lambda \to$ RGB mapping. This spectral conversion pipeline is further applied in all methods presented in this chapter.

Additionally, to preserve edges induced by the receiver geometry, each fragment along the splatted line is weighted by a term which gives higher weight to fragments with similar surface properties, comparably to geometry-aware joint-bilateral filtering [Eisemann and Durand 2004; Petschnigg et al. 2004]. These bilateral weights are obtained by storing the geometric properties (positions and normals) of the receiver surface where the connecting camera rays hit it.

As mentioned before, in reality the footprint over the entire spectrum might potentially not be a line or a connected curve, but can consist of multiple, disconnected segments due to the possible divergence of rays with different λ. The precise reproduction of this effect depends on how finely the spectrum is sampled and obviously is beyond the scope of the first-order approximation. However, the occurrence of this phenomenon is comparatively rare, and mainly, by progressively decreasing the filter footprint the reproduction consistently improves.

Gathering The other alternative is, along with producing a spectral radiance image $\mathsf{L}(\mathbf{x},\lambda)$, to create an additional image where the positional differential of each light path is averaged at each pixel. Practically, whenever a light sample is connected to the sensor, its positional differential is accumulated at the sensor location $\mathbf{x} = \rho(\mathbf{p})$. The distribution of offsets is meaningfully represented by a mean if a single caustic is projected onto a pixel and becomes gradually less precise if multiple divergent light paths are averaged, e. g., in the case of overlapping caustics.

Now, two assumptions can be used to filter the image L. First, the projected offset $\Delta\mathbf{x} = \rho(\Delta\mathbf{p}) \in \mathbb{R}^2$ should vary smoothly within the neighbourhood of each pixel at position $\mathbf{x}$. Second, we need to assume that the first-order approximation $\mathsf{L}(\mathbf{x},\lambda+\Delta\lambda) \approx \mathsf{L}(\mathbf{x}+\Delta\mathbf{x},\lambda)$ can predict radiance at one position and one wavelength from the differential and the radiance at different positions and different

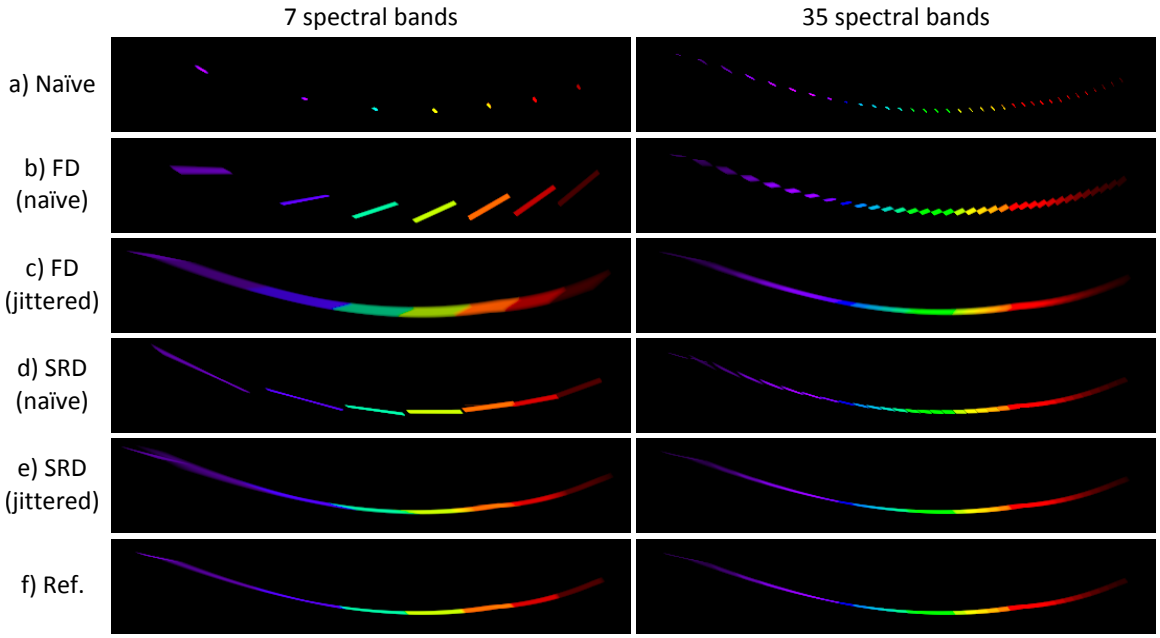


Figure 6.5: Narrow pencil of light dispersed by a curved object, comparing finite-difference reconstruction *(b,c)* and spectral differentials *(d,e)* with the basic naïve *(a)* and jittered *(f)* sampling approaches, for two different spectral resolutions. See the detailed discussion in Section 6.6.1.

wavelengths. If this holds, we can simply filter the radiance image with a 1D linear kernel with the size and orientation of $\pm\Delta\mathbf{x}$. At every location $\mathbf{x}$ the values between $\mathsf{L}(\mathbf{x}-\Delta\mathbf{x})$ and $\mathsf{L}(\mathbf{x}+\Delta\mathbf{x})$ are averaged, effectively corresponding to a convolution with a (spatially variant) oriented linear kernel. Same as in splatting the estimate employs the bilateral weights based on the receiver geometry.

6.5.2 Eye Tracing

Eye tracing operates similarly to light tracing except that the sensor (i. e., the image L) is sampled instead. In each pass, rays are started at sensor positions $\mathbf{x}_\mathrm{e}$ and traced through the scene, yielding (emitting) receiver intersections $\mathbf{p}$ and the corresponding spectral differentials. These again define 1D sensor footprints from $(\mathbf{x}_\mathrm{e}-\rho(\Delta\mathbf{p}))$ to $(\mathbf{x}_\mathrm{e}+\rho(\Delta\mathbf{p}))$ due to the reciprocity of light transport. Although splatting is still applicable, it is less practical here since the samples already correspond to sensor (and image) locations directly.

Same as in the gathering version of light tracing, it is alternatively possible to store the SRD statistics ($\Delta\mathbf{p}$ and optionally multi-lateral filtering weights) in separate images and after the tracing stage apply a 1D spatially-variant linear filter to the solution of each pass, before it is accumulated with the overall solution. Advantageously, in contrast to light tracing, here this approach does not suffer from the additional bias due to averaging of the differential statistics, as each progressive pass is filtered separately and only then accumulated. In principle this would be possible in both methods, however in light tracing the projected samples will have noisy and most likely also irregular distributions, decreasing the robustness of the filtering process significantly.

Further consideration is also necessary in defining the geometry-based filtering weights. In addition to the receiver geometry, the weights (based on positional and normal discrepancies between neighbouring samples) need to be derived from the dispersive object itself. However, these cannot

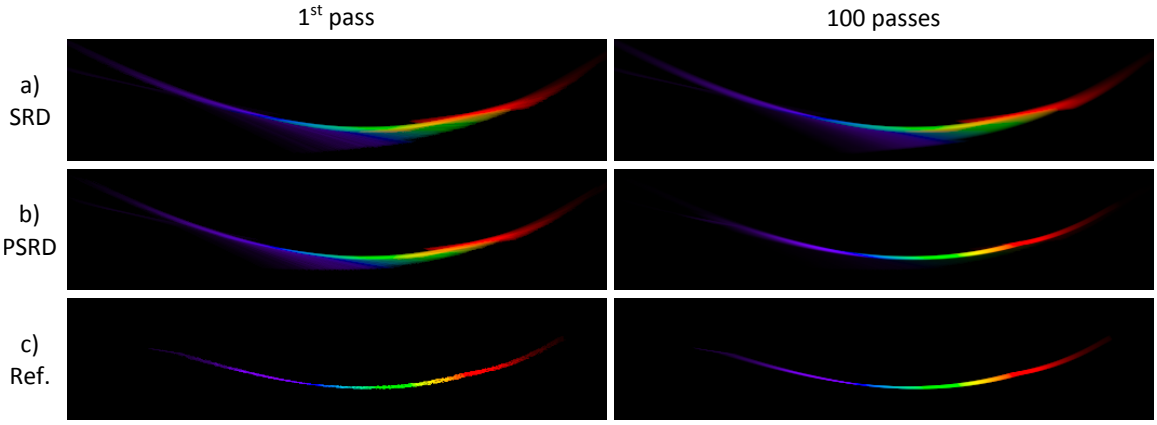


Figure 6.6: Narrow pencil of light dispersed by a curved object, comparing basic *(a)* to progressive SRD *(b)* and the reference *(c)*, across multiple passes. For details see the discussion in Section 6.6.1.

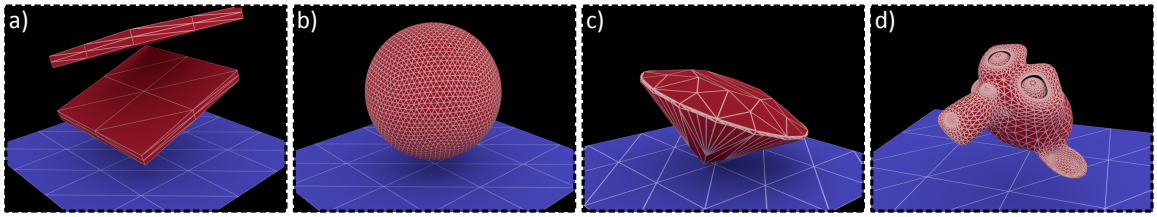


Figure 6.7: Schematics of the scene compositions used for Figures 6.8–6.11, as well as Figures 6.12 and 6.13. The resulting images in these figures are essentially caustic 'maps', representing the radiance reflected by the receiver. Refractive objects are drawn red, while blue corresponds to an opaque diffuse receiver, with a constant directional light field coming from the top.

be based on its first-intersected surface (as is normally done), because rays originally entering through the same interface can easily become divergent due to effects like internal reflection, causing them to exit the object in different locations and yield entirely different differentials as a result (cf. Figures 6.17, 6.18 and 6.19). The solution to this issue is to base these weights on the *exiting* locations' geometry, which correctly prevents the blurring of discontinuities between divergent sample rays.

6.5.3 Photon Mapping

Photon mapping combines the two above approaches, connecting them via a kernel density estimate (Section 2.5.1). Spectral differentials add an additional dimension to this estimate, which results in an anisotropic density estimation kernel as opposed to the standard isotropic (e. g., Gaussian) kernel (Equation 2.31). Essentially, the photon map becomes the differential of the entire illumination function: it describes where photons land, but also the first order approximation of where they would move if the wavelength was changed by a differential increment. A simpler alternative is again direct splatting of photons, which is used in the presented interactive application in Section 6.6.4.

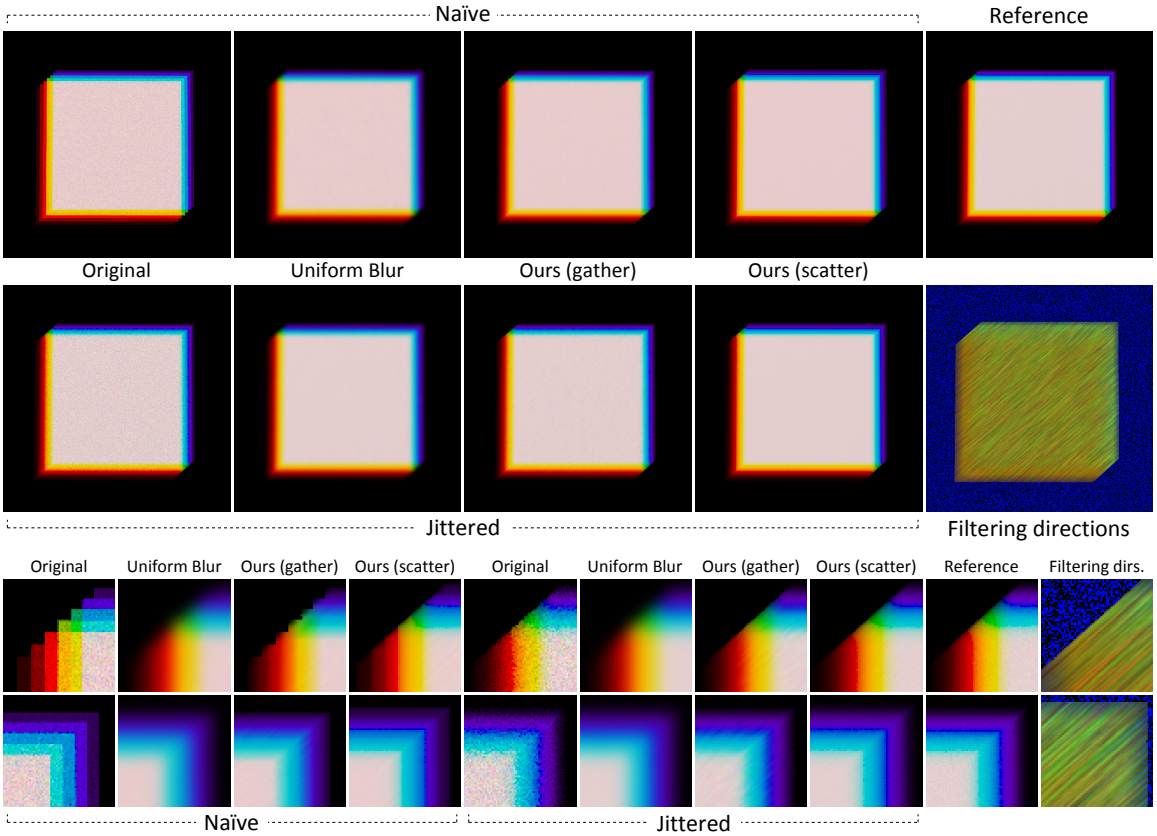


Figure 6.8: Analysis of the "slabs" caustic scene (Figure 6.7, a) discussed in detail Section 6.6.2. The rendering time was: 10 seconds (naïve and jittered) versus 2 minutes (reference).

6.6 Results and Analysis

The presented results use the three common image synthesis approaches discussed in Section 6.5 (offline light and eye tracing and interactive splatting-based photon mapping, all implemented on GPU using OpenGL and GLSL). Splatting is performed by rasterising line (light tracing) and quad (photon mapping) primitives procedurally generated from point samples in a geometry shader. Additionally we show how spectral differentials can be used for interactive artistic control of physically based rendering and conclude with an additional analysis.

Light tracing has been chosen as the primary representative algorithm to analyse SRD. This is because the effects of applying them can be well isolated and examined most clearly, as no other sources of illumination or bias are present (compared to the alternatives). This also facilitates the most direct numerical analysis.

Two ways of *sampling* the spectral domain are used throughout this section: "naïve" and "jittered". The naïve results are generated with discrete sampling of the spectrum at regular intervals. On the other hand, the jittered variants are produced by sampling the spectrum stochastically. The latter is naturally preferred (and is also used to produce all reference images), although we have decided to compare against the naïve approach as well, because, despite being overly simplistic, there can still be encountered cases when it is used in practice (typically due to software limitations).

Another important remark is that, when relevant, the analysis is done separately for non-progressive

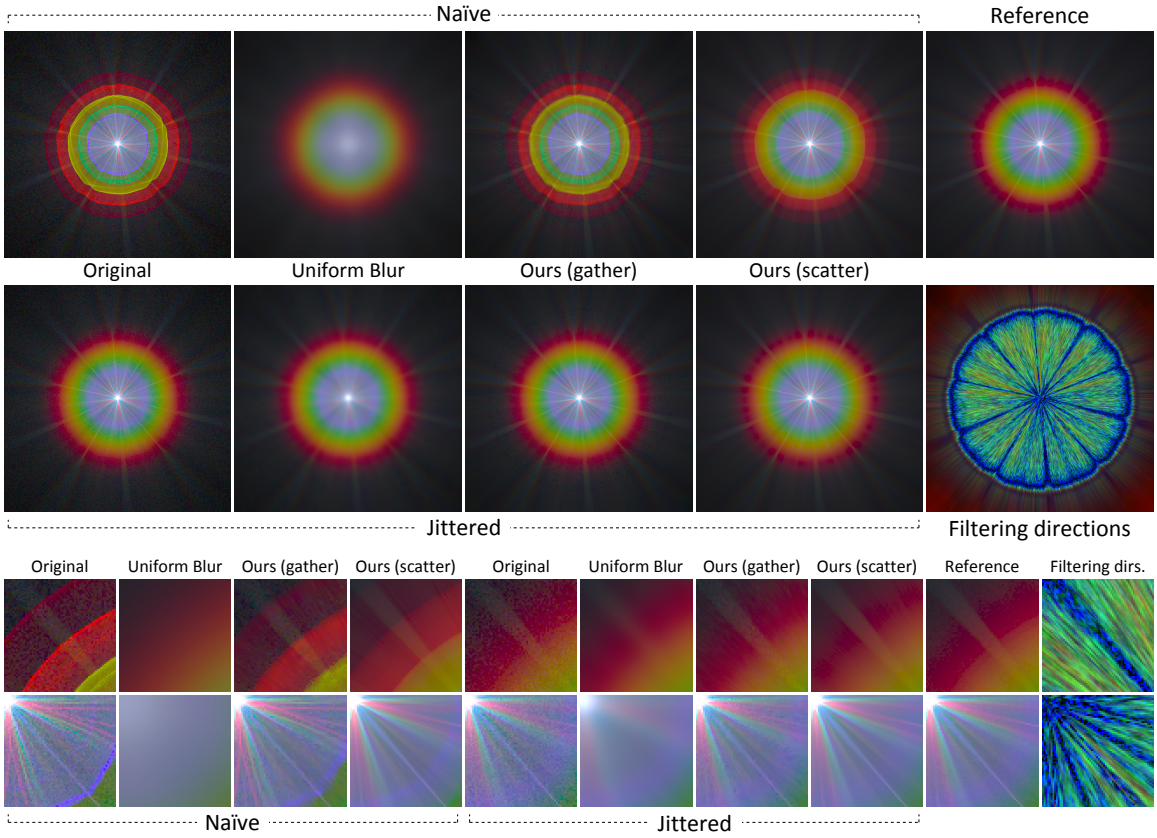


Figure 6.9: Analysis of the "sphere" caustic scene (Figure 6.7, b) discussed in detail in Section 6.6.2. The rendering time was: 15 seconds (naïve and jittered) versus 4 minutes (reference).

and progressive SRD (Section 6.4). This is to first understand and quantify the impact that using spectral differentials has on the resulting images. Therefore, in the non-progressive variants the $\Delta\lambda$ is kept constant, typically in relation to the spectral sampling granularity – for naïve sampling it is inversely proportional to the number of bands, and same for the jittered variants so that they remain comparable to the naïve results.

Performance is reported for an NVidia GeForce GTX 770 graphics card. All the presented images were rendered at the resolution of 512^2. For eye tracing, one sample is traced per pixel and per pass, while light tracing universally uses $M = 200\text{k}$ samples per pass, which in all experiments amounts approximately to 50 ms (meaning that tracing SRD does not incur an appreciable overhead).

6.6.1 Basic Analysis

To gain an insight about the behaviour of SRD, the first test consists of a single narrow beam of light dispersed by a curved object, resulting in a simple 'rainbow' caustic (Figure 6.5). The first row *(a)*, shows the naïve variant of sampling the spectrum, resulting in discrete discontinuous sampling of the caustic as expected. Second, rows *(b,c)* show results using (naïve and jittered) *finite differences*, which essentially correspond to tracing a second 'virtual' ray alongside the actual one, with the wavelength offset by $\Delta\lambda$. This approach works to an extent, although the curvature of the object is neglected, resulting in a poor approximation of the caustic's shape. Finally, rows *(d,e)* apply (again, with naïve and jittered sampling respectively) analytic spectral differentials, leading to more

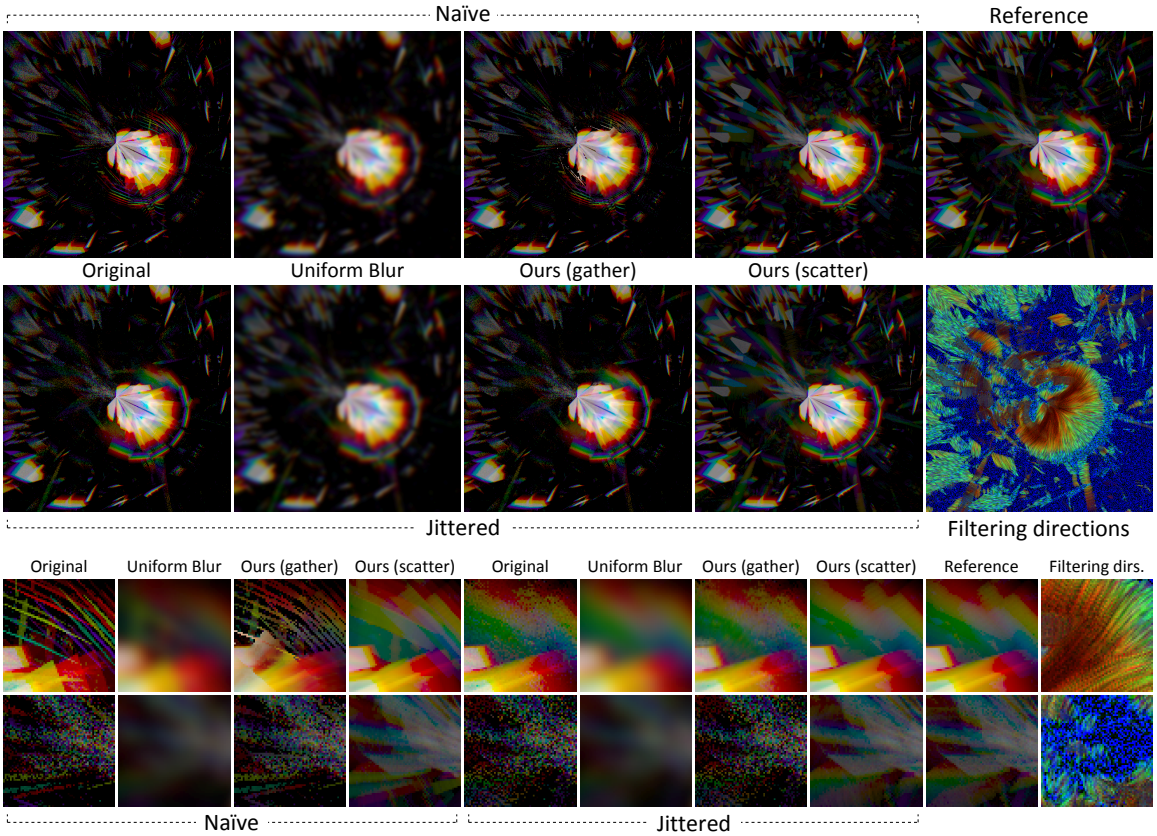


Figure 6.10: Analysis of the "diamond" caustic scene (Figure 6.7, c) discussed in detail in Section 6.6.2. The rendering time was: 2 minutes (naïve and jittered) versus 60 minutes (reference).

continuous approximation closer to the reference (row *(f)*) thanks to the fact that SRD take into account also normal differentials which are influenced by the object curvature.

In spite of that, due to the intrinsic limitations of the first-order approximation, the SRD variant still contains a certain bias. This is better visible in Figure 6.6, a, where the refractive index range has been expanded. The bias is manifested as a spatial blur, clearly visible in comparison to the reference (row *(c)*). On the other hand, the progressive SRD variant (row *(b)*), despite initially containing this bias as well, over multiple passes converges to the correct shape since the blur diminishes with decreasing $\Delta\lambda$.

6.6.2 Light Tracing Results

Basic SRD Figures 6.8–6.11 show results of light tracing in combination with spectral ray differentials. Four scenes are presented: simple "slabs" (for didactic purposes, Figure 6.8), and three more complex objects – tessellated "sphere" (Figure 6.9), faceted "diamond" gem (Figure 6.10) and finally "monkey" containing many high-curvature features and concavities (Figure 6.11). Both the "naïve" and the "jittered" results are generated using 7 spectral bands for the traced samples. The references are generated using 35 jittered spectral bands and 2–4 times more samples (depending on the scene). All scenes show caustics on a diffuse white receiver from the objects with an IOR of 1.6–1.4 lit by a white beam light source from the top (see Figure 6.7). Included in a line integral-convolution (LIC) visualization [Cabral and Leedom 1993] of the projected positional spectral ray differential, with a

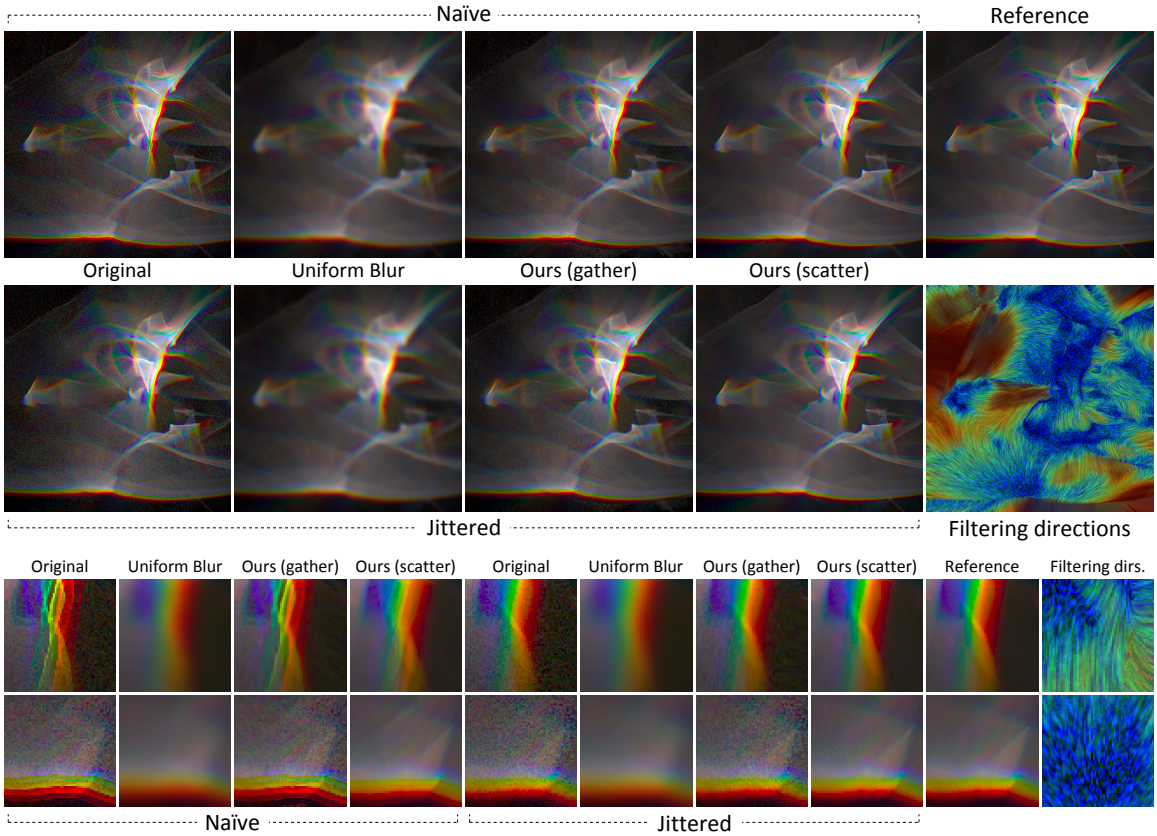


Figure 6.11: Analysis of the "monkey" caustic scene (Figure 6.7, d) discussed in detail in Section 6.6.2. The rendering time was: 1.5 minutes (naïve and jittered) versus 30 minutes (reference).

false-colour coding for its magnitude.

First, note that the naïve solutions suffer from visible spatial aliasing artifacts ('banding'). Jittering the spectrum removes the aliasing but generally contains more noise (i. e., the error is only distributed differently if an equivalent number of samples is used, as indicated by the quantitative analysis at the end of this section). For both of the sampling variants we first applied a spatially uniform Gaussian filter with a support just large enough to remove the respective aliasing and noise artifacts. As expected this removes most details in the caustics as well. We then applied the SRD-based splatting and gathering reconstruction techniques (Section 6.5.1). Gathering works well in case of a single structured caustic, such as Figure 6.8, but fails to remove noise or leads to over-blurring in cases where the accumulated positional differential estimate is noisy as well or when multiple divergent caustics overlap at a same location (e. g., Figures 6.10 and 6.11). Splatting, on the other hand, does not suffer from these problems as each sample is reconstructed independently, and consequently its application removes all visible noise and in the jittered variant leads to results visually indistinguishable from the reference. As a few examples, notice how details like the concentric streaks in Figure 6.9, individual facets in Figure 6.10, or curved features in Figure 6.11 are well preserved in contrast to the original unfiltered solutions.

Progressive SRD A separate analysis is provided for the "monkey" (Figure 6.12, with a different orientation corresponding to Figure 6.1) and "diamond" (Figure 6.13) scenes. The aim has been to examine the rate of convergence and resulting bias (visually and numerically, see below) of the progressive version of SRD in comparison to the basic one and to unbiased reference. The SRD

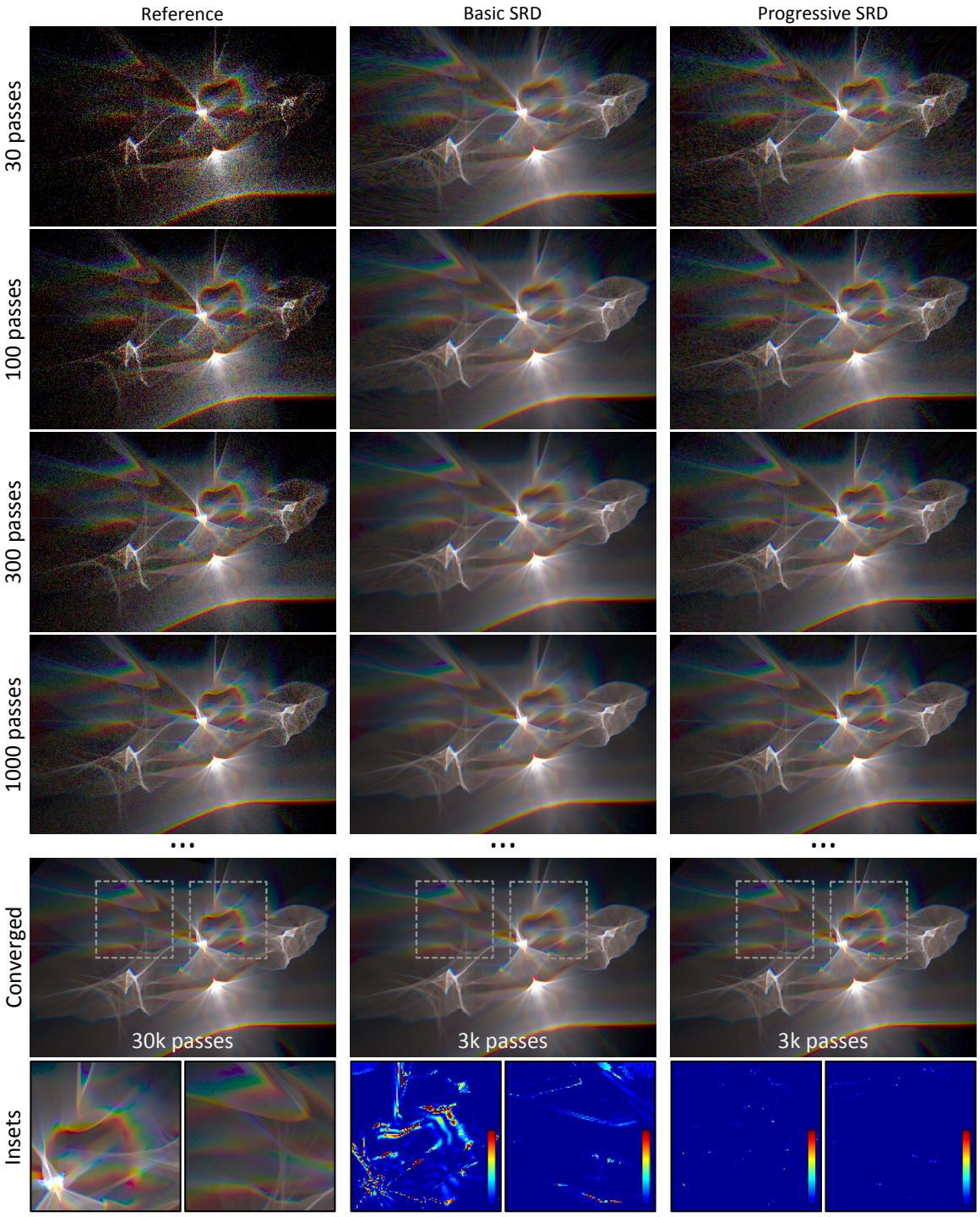


Figure 6.12: Progressive convergence of the “monkey” caustic (see Section 6.6.2 for a detailed discussion). The insets show false-colour-coded RMSE differences between the SRD / PSRD solutions and the reference.

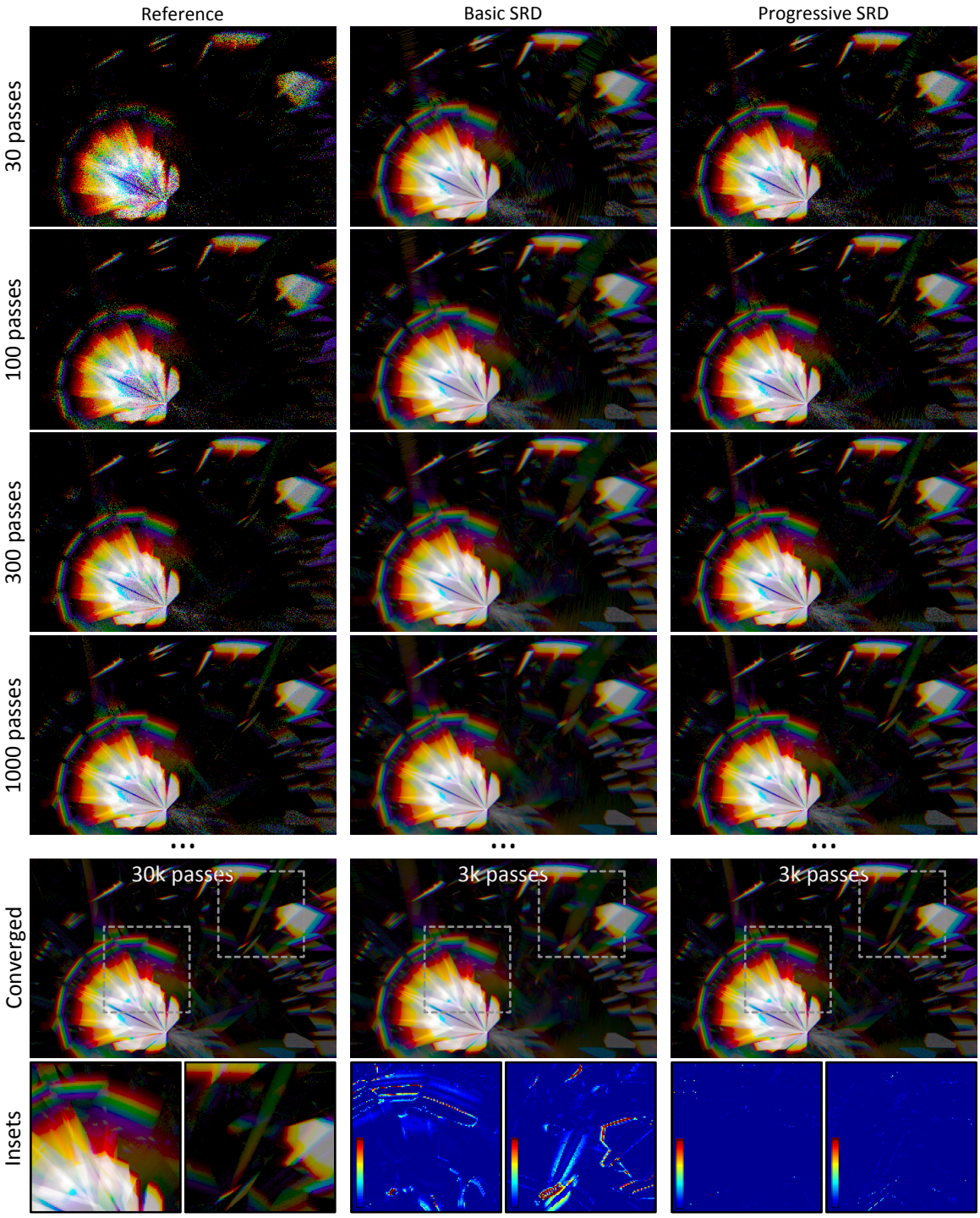


Figure 6.13: Progressive convergence of the "diamond" caustic (see Section 6.6.2 for a detailed discussion). The insets show false-colour-coded RMSE differences between the SRD / PSRD solutions and the reference.

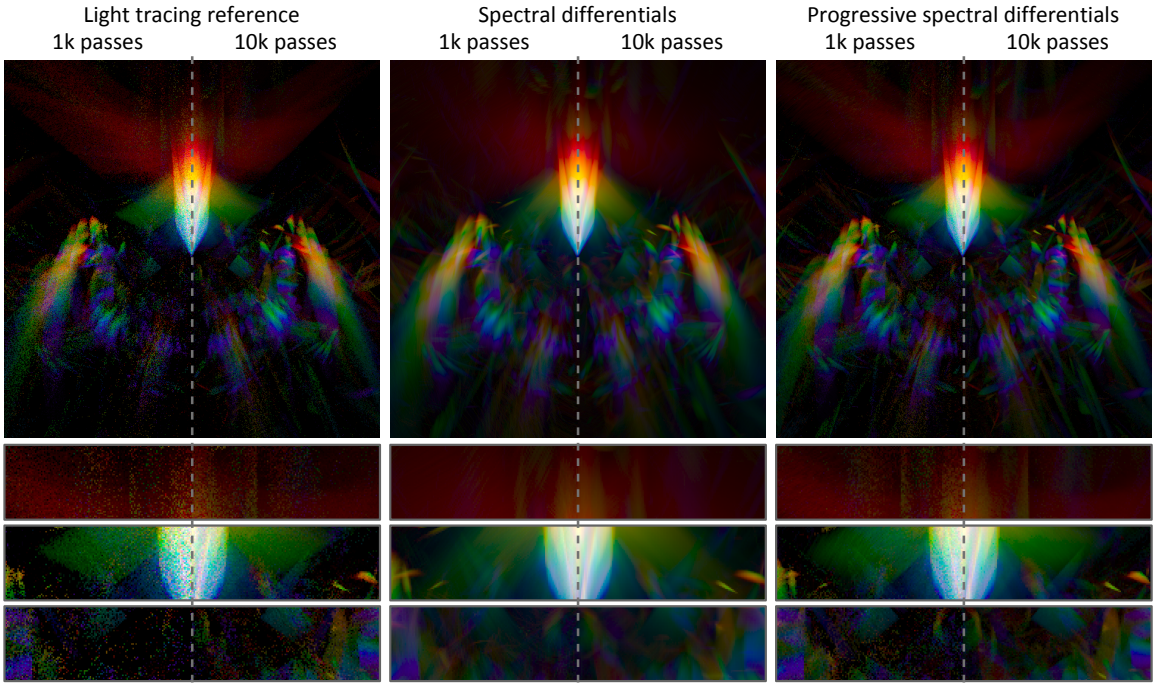


Figure 6.14: Intricate caustic resulting from chromatic dispersion on a brilliant-cut gemstone (Figure 6.7, c) but with a very wide refractive index range (from 2.2 for the violet end to 1.2 for the red end of the spectrum). The (jittered) light tracing reference produces very noisy results, due to the large dispersion magnitude, and does not fully converge even after a high number of passes. Basic SRD lead to a smooth solution and an order of magnitude faster convergence, the solution is however biased, which is manifested as blur of the caustic features (see insets below). The progressive SRD variant converges almost as fast and leads to the unbiased solution in the limit, so that the sharp features are correctly reproduced.

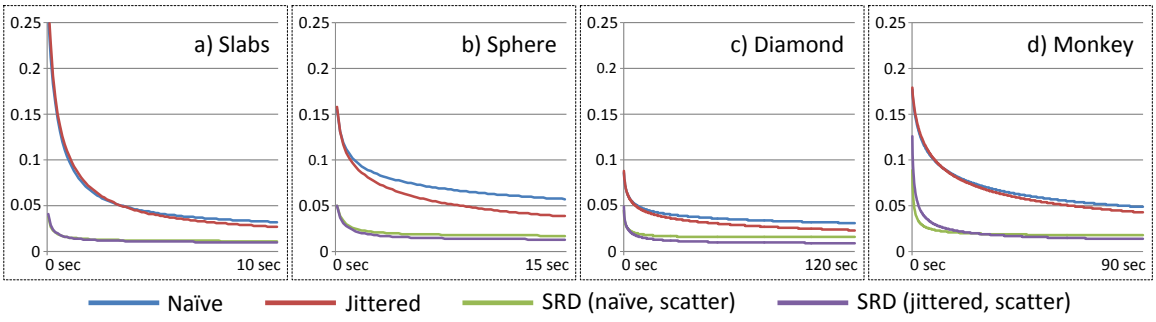


Figure 6.15: Convergence for the caustic result images Figures 6.8–6.11 using the splatting reconstruction (RMSE between the result and the reference images).

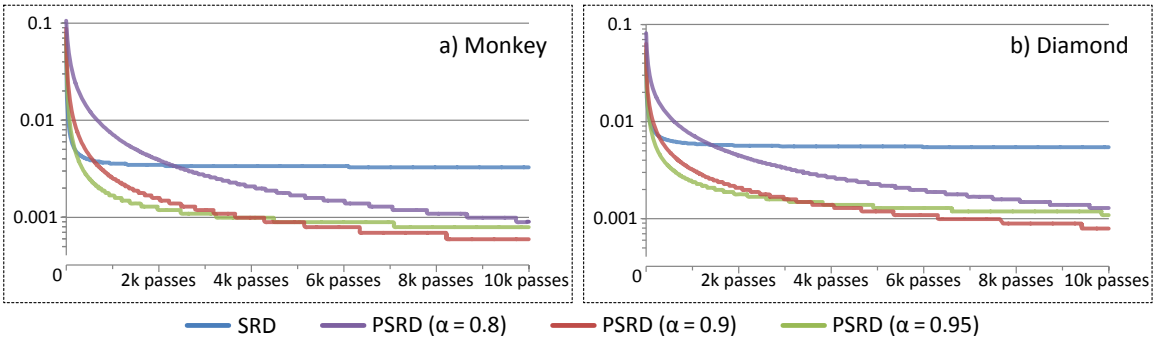


Figure 6.16: RMSE plots showing the progressive convergence over 10k passes of the monkey (Figure 6.12) and gem (Figure 6.13) scenes. SRD converge very fast but to a biased solution, while PSRD, regardless of the actual value of parameter α, approach the correct solution at the rate characteristic for other Monte-Carlo approaches.

solutions here use a constant $\Delta\lambda$ value corresponding to $1/6$ of the visible spectrum range. In contrast, the PSRD solutions start with the same $\Delta\lambda$ value, which is then progressively updated using Equation 6.19 with $\alpha = 0.9$. All the solutions are shown after selected numbers of passes, as well as converged to a stable state.

The conclusions to draw from this experiment are in agreement with the theory discussed in Section 6.4. The reference solution is very noisy, especially in areas with low sample density, and therefore requires a high number of passes to converge. SRD produce an exceptionally smooth solution, requiring about 1–1.5 orders of magnitude less passes (and samples thereof). The results however contain visible bias manifested as spatial blurring of features, similarly to Figure 6.6 (cf. difference images). In contrast the PSRD solution, as expected, has a slightly slower convergence rate but the converged results are visually indistinguishable from the reference and numerically converge to it.

Although the bias produced by SRD is often subtle it will generally increase with the dispersion magnitude, but also with distance the light travels after the interaction, as dispersion is an angular phenomenon. In Figure 6.14 this is demonstrated this by artificially increasing the IOR range, which leads to a much stronger dispersion. Here the bias is immediately apparent, causing blurring or even different shaping of the caustic features.

Quantitative analysis While the basic SRD reconstruction results in images with less noise, it might not be immediately obvious if they actually are better in relation to the reference, as any blur results in a biased solution. The plots of the root mean squared error (RMSE) in Figure 6.15 however show, that even with the blur, the results indeed are closer to the corresponding references and converge to a very similar solution. Here we analyse the convergence of the splatting reconstruction compared to the original solutions for both the naïve and jittered variants of Figures 6.8–6.11. It can easily be seen that the SRD-based solutions are not only numerically better in all test cases, but also converge much faster. This is thanks to the fact that the splatted samples are 1D lines as opposed to just point samples in the original solutions.

We have additionally measured the RMSE between the SRD and PSRD solutions, both in comparison to a converged reference solution. These measurements (corresponding to Figures 6.12 and 6.13) are plotted in Figure 6.16. For the PSRD variant, three different values of the parameter α have been tested. It is apparent that the basic SRD solutions converge the fastest, but not to the expected error value of 0, resulting in bias. The PSRD empirically turn to behave optimally for α values around 0.9.

For smaller α, such as 0.8 or less, the algorithm arrives to very small $\Delta\lambda$ values too quickly, leading to higher variance and slower convergence. And naturally, the opposite holds for α values close to 1.

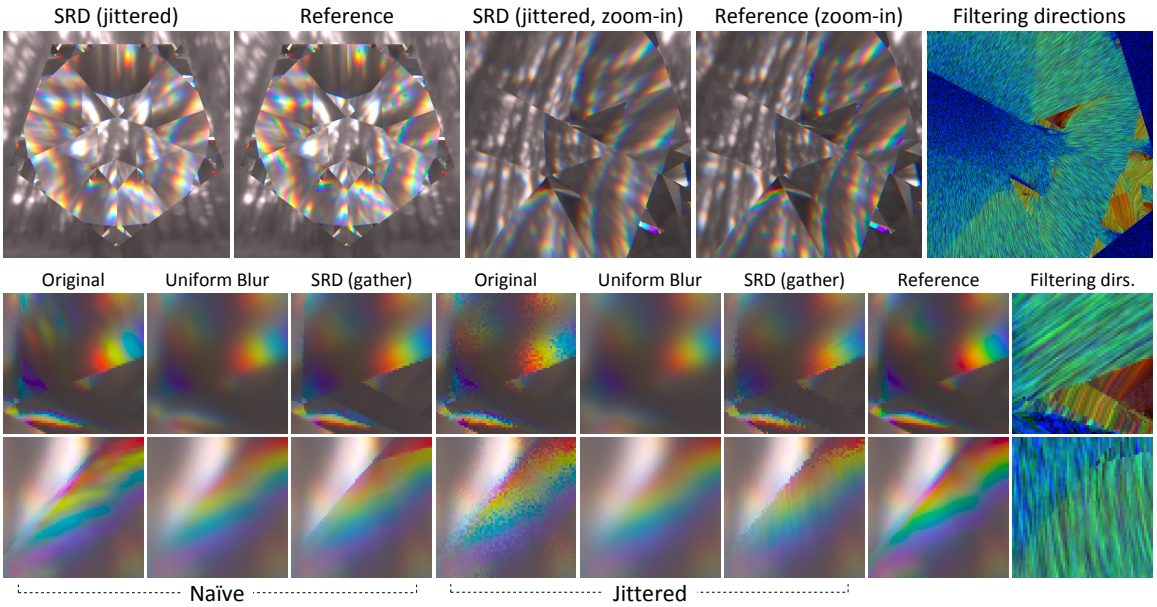


Figure 6.17: Icosahedron object made of glass. Computed using dispersive eye tracing and reconstructed using different methods (*insets*). Detailed discussion in Section 6.6.3.

6.6.3 Eye Tracing Results

Figures 6.17, 6.18 and 6.19 present eye tracing results for three sample objects and transparent materials (glass icosahedron, diamond gemstone and monkey composed of water). The corresponding references were again generated using 35 jittered spectral bands with 40 samples per pixel, while the original (unfiltered) naïve and jittered variants used 5 spectral bands and 4 samples per pixel. The variants containing SRD-based reconstruction were, again, about 1–1.5 orders of magnitude faster than the reference.

The comparison is similar to the light tracing case (Section 6.6.2), although only the gathering reconstruction is evaluated, since the rendering and the resulting reconstruction natively take place in the image space (see discussion in Section 6.5). The artefacts inherent to the naïve and jittered variants are present here in a similar form, as expected. Gathering also suffers from similar problems (noisy guide images of the bilateral filter and overlapping positional differentials), but in general produces results with less noise and closer to the reference. We have opted for rather conservative settings regarding filtering magnitudes and bilateral weighting, so that some noise can remain but the edges in the image are preserved. The remaining problems, however, are only visible at strong close-up views and the rendered objects under normal viewing conditions generally look very plausible.

6.6.4 Interactive Photon Mapping and Editing

Figure 6.20 shows image-space dispersion in an interactive rendering setting using SRD for eye and light paths at the same time. To generate the images with approximate reflection, refractions

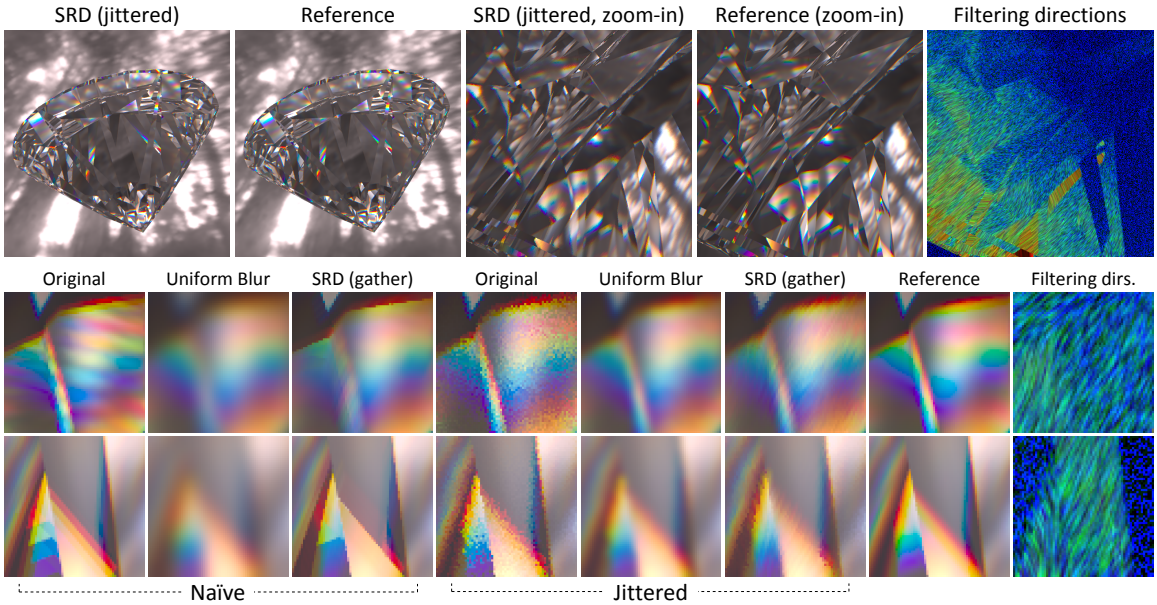


Figure 6.18: Brilliant-cut diamond. Computed using dispersive eye tracing and reconstructed using different methods (*insets*). Detailed discussion in Section 6.6.3.

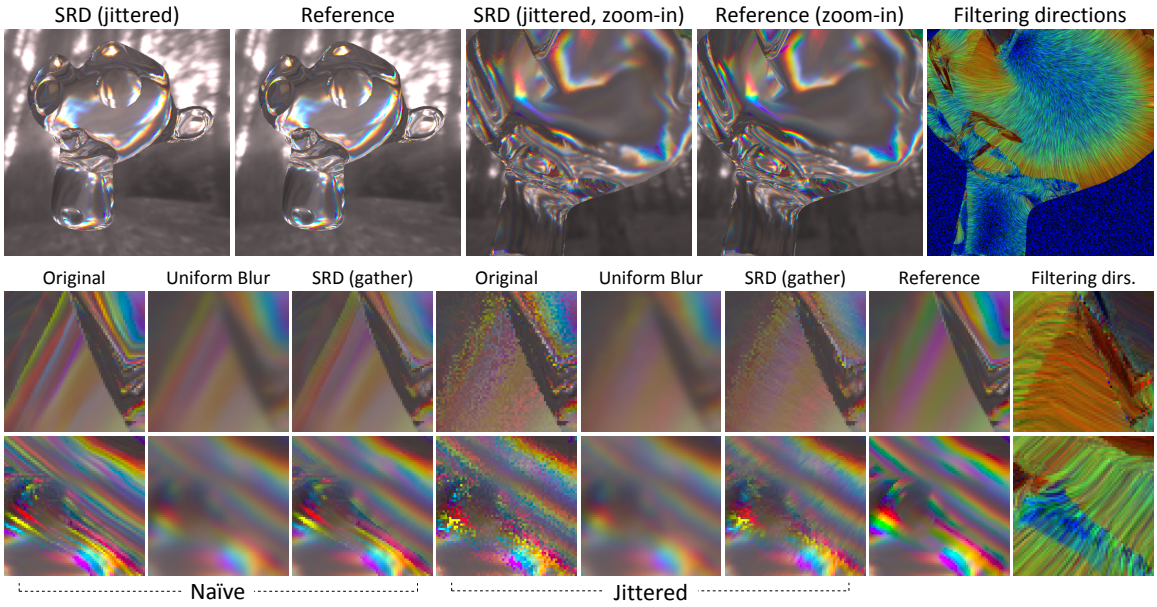


Figure 6.19: Monkey head composed of water. Computed using dispersive eye tracing and reconstructed using different methods (*insets*). Detailed discussion in Section 6.6.3.

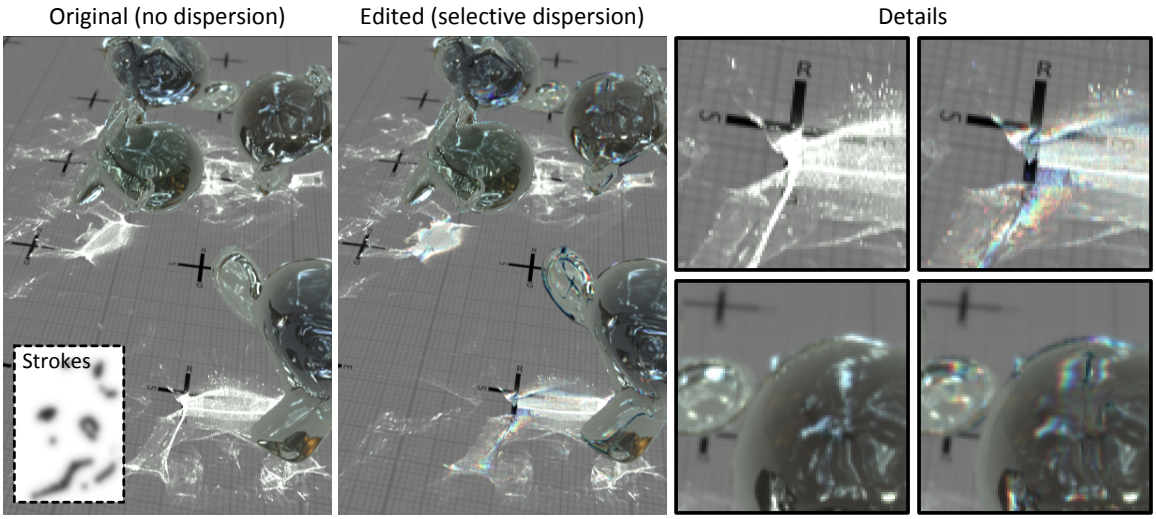


Figure 6.20: Interactive caustic rendering and editing (see Section 6.6.4) using photon splatting (2048^2 caustic map, 40 ms rendering). Dispersion is selectively generated by image space filtering in about 1 ms, using purely the information provided by spectral differentials. The strokes *(bottom left)* are not binary, but actually specify the dispersion magnitude, which—if artistically desired—allows consistently specifying a separation between the spectral components even beyond physical limits.

and caustics, we use a GPU-based photon mapper based on splatting [McGuire and Luebke 2009] in combination with caustic mapping [Wyman and Davis 2006]. Along with eye-refractions and caustics, the spectral differential is computed and averaged in screen space per-pixel. In the final pass, a 1D filter with 5 jittered taps in the direction defined by the positional differential is used to convolve the RGB image with RGB spectral response curves. This filter is very fast, requiring roughly 1 ms per megapixel.

Artistic control Dispersion can undoubtedly yield images with certain artistic qualities, such as shown in Figure 6.21 or by the work of Peter Erskine (example at www.thisiscolossal.com/2015/04/peter-erskine-prisms). However, artistic control over physically-based image synthesis is difficult to achieve. For dispersion, the proposed positional differential offset field can be used to guide artistic effects and can be controlled locally in image space by simple scaling. Starting from an image without dispersion (Figure 6.20, left) the user paints a multiplicative 2D control field (shown inverted) and the system interactively produces a plausible dispersed image by applying the filtering step using an offset field that is modified by the user guide, leading to the result in Figure 6.20, middle. Note how such local and interactive control would be very difficult—if not impossible—to achieve by modifying physical scene parameters such as IORs or geometry.

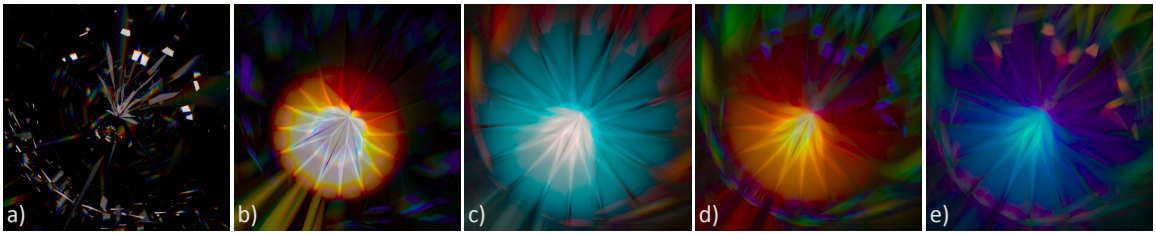


Figure 6.21: Caustic by a diamond *(a)* (violet-to-red IOR range 2.5–2.4) and multiple caustics resulting from different meta-materials with IOR ranges 1–1.5 *(b)*, 0.05–2.4 *(c)*, 0.5–2.5 *(d)* and 3.5–0.1 *(e)*. The scene composition is identical to Figure 6.10.

Chapter 7

Conclusion

The work presented in this thesis deals with physically-based rendering of various types of participating media. We have briefly exposed this problem in Chapter 1 and provided an overview of the relevant physical and mathematical bases, as well as some of the difficulties associated with the topic, in Chapter 2. Chapters 3–6 then presented several approaches to solve particular questions in the area.

This chapter summarises and concludes this effort. A general discussion is provided first, followed by a summary of the presented topics and possible future work stemming from each.

7.1 Summary and Outlooks

Light interacting with participating media has a behaviour that is complex to simulate. The most characteristic type of interaction is scattering, which is at the same time most difficult to reproduce properly in all conditions. On one hand, scattering is formulated as a highly multi-dimensional integration problem, and no analytic solutions are available to solve it except for the simplest of cases. From another point of view, scattering can be seen as a loss of information, since it locally corresponds to a directional convolution and globally leads to a decrease of frequency of an incoming illumination (proportionally to the medium's optical thickness). Yet again, due to its spatially varying nature it is difficult to characterise scattering by available statistical approaches except for most trivial (homogeneous and isotropic) conditions. Even transparent media which do not *visibly* exhibit scattering (due to their internal structure) still cause dispersion, which is a nonlinear, spectrally dependent phenomenon that has proven difficult for stochastic methods to sample.

All these interactions can theoretically be simulated by general Monte Carlo methods, but in practice this can be too cost-prohibitive due to the multitude of visual effects that media can have on scenes' appearance. This is especially true for interactive applications. Here it is virtually unavoidable to rely on specialized methods that utilise prior knowledge about the environment to be simulated.

The ordering of the thesis' core chapters is mirrored in the remainder of this section. The chosen order is not incidental, but reflects the degree of specialization of each of the presented methods and also its resulting efficiency. Chapters 3 and 4 make relatively restrictive assumptions about the simulated media (homogeneous media and clouds), but as a result reach fast execution speeds in the order of milliseconds. Chapter 5 focuses on the more general glass of optically thick media, relaxing

most of the preceding work's assumptions yet still achieving interactive speeds in around tens of milliseconds. Finally, Chapter 6 describes the specific interaction with dispersive transparent media but is applicable in most general MC methods, increasing the efficiency of reproducing dispersion by about an order of magnitude.

7.1.1 Screen-Space Scattering (SSS)

The approach from Chapter 3 allows approximating the distinct colouring and blur effects that can be essential to convincingly depict participating media such as water or foggy environments. It shares the usual shortcomings of screen-space approaches, in that it does not account for illumination not directly present in the framebuffer, but also all their beneficial properties, like high speed – taking only a few milliseconds for typical screen resolutions – and the ease of implementation and integration into an existing image-space / deferred-shading pipeline.

Approximation by a PSF Approximating scattering or any other global effect in this way is inherently limited. First, the derivation of the underlying PSF itself by Premože et al. [2004] (Equation 2.36) relies on many simplifying assumptions. Although the derivation considers multiple scattering, the PSF expression is limited to homogeneous media and a narrow incident light beam. It would still be possible to include heterogeneous media in an approximate way by using ray-marching to integrate the medium parameters for each pixel individually; this would however increase the cost of our approach significantly. An alternative could be using the PSF derived by Narasimhan et al. [2003; 2004] which is not limited to a simple Gaussian and is also valid for point light sources; the disadvantage of this model is its formulation as an infinite series, so further investigation would be required to see how this model could be implemented in an efficient way using, e. g., the presented pyramidal approach.

Also, as the employed model simulates multiple scattering as a blurring operation, high-frequency effects due to both on- and off-screen sources are not captured by our technique. These include phenomena such as light shafts and volumetric caustics, which in scenes with high dynamic range are often important to convey the presence of a participating medium. Given the character of our method, it is clear that simulating these requires employing a complementary approach for calculating low-order scattering effects, such as the epipolar-sampling technique by Engelhardt and Dachsbacher [2010].

Screen-space limitations Using only the HDR image of the scene as a starting point for approximating scattering is of course not entirely correct. The reason is that the calculation of the illumination even before considering the camera should already take the medium into account. A closely related limitation is that light sources not directly visible are not accounted for. The simple measure used in most of our examples is the addition of an emissive term derived from the intensity of the total illumination in the scene. A more interesting and systematic approach would be to use deep screen space [Nalbach et al. 2014a] which is a general framework aimed at overcoming the inherent limitations of screen-space techniques. The advantage of deep screen space is that it provides information about all scene objects, including the occluded and off-screen ones. Its volumetric extension [Nalbach et al. 2014b] however considers only single and double isotropic scattering. Combining this approach with a more general anisotropic PSF expression and a fast pyramidal gathering in image space could greatly extend the amount of reproducible effects. The main challenges in doing so are how to actually calculate the multiple-scattering contributions from source objects, but also how to efficiently access them (since, currently, deep screen space is limited

to a splatting-based accumulation of these data).

Decoupling scattering and absorption Despite the fact that absorption and scattering are intertwined effects the former needs to be taken care of before the latter in a screen-space method like ours. The reason is that absorption is a distance-dependent effect that can produce subtle but perceivable hue changes even for small distance variations. As such it needs to be calculated in the full-screen resolution where the distance information is pixel-accurate, and hence is already cannot be done during the lower-resolution MIP map construction. However, it can neither be calculated *after* the filtering phase, because the blurred radiance coming from different distances will exhibit different levels of absorption; if one tried to compute the absorption after first blurring the scattered radiance, multiple 'halos' from different distances could potentially overlap in any given pixel, but each pixel would be attenuated depending on *its own* distance, which might not correspond to distances the halos originate from.

Hierarchical convolution The approximation of an incremental convolution by a hierarchical one also bears some limitations, especially if an anisotropic filter is used as in our case. Although the lower-resolution MIP levels are sufficient to represent the blurred information from the perspective of its frequency, some spatial or temporal high-frequency artefacts can still appear – mostly due to the fact that the distance information in the higher MIP levels is averaged. This might become a problem in the presence of high-frequency illumination or very dense media, in which case increasing the filter support size might be necessary.

7.1.2 Amortised Photon Mapping (APM)

Chapter 4 presents an interactive cloud rendering method. The algorithm utilizes a temporally-coherent illumination caching process to amortize the simulation costs across multiple frames. Our novel representation of angular illumination distribution inside clouds enables reproducing the characteristic appearance of clouds, while keeping the computational and storage costs modest. The algorithm maps very well to the architecture of modern graphics hardware, and hence all its major steps can be evaluated in parallel on the GPU.

The proposed method improves on the existing techniques in multiple regards. Realistic observations about the typical environment of clouds were used in its design. The approach is not limited to any particular cloud type or any subset of possible light paths in clouds, and can handle the light sources important in the simulated environment. All physically plausible properties of clouds are considered and the entire method is physically meaningful.

Scaling In the current form the method's performance scales linearly with the number of simulated clouds, which is certainly not practical. It would be beneficial to investigate if, and what kind of a multi-resolution scheme can improve this. A suitable hierarchical representation of the photon cache could significantly reduce the number of photons required for rendering clouds that are distant from the camera.

Directional caching In addition, the algorithm could be improved and even adapted to other types of media by relaxing or even completely dropping the specific assumptions about the simulated environment. For example, multiple HG lobes could be used to represent more complex illumination distributions than the ones dealt with in the described approach. Such representation could be used not only for (approximate) direct rendering, but also to guide an unbiased algorithm such as path tracing (which has actually been the original motivation behind photon mapping [Jensen 1995]).

Several related approaches for clustering both spatial [Jakob et al. 2011] and directional [Lafortune and Willems 1995; Jensen et al. 2001; Lehtinen et al. 2008; Hachisuka et al. 2008; Lehtinen et al. 2009; Lehtinen et al. 2010; Vorba et al. 2014] photon distributions based on expectation maximisation are already available.

The most relevant is the work of Vorba et al. [2014] which employs a mixture of spherical Gaussians to cluster indirect illumination on surfaces, and uses this information for importance sampling in an additional rendering pass. In theory this approach could be transferred to volumetric rendering directly, since the Gaussian mixtures are not limited to hemispherical distributions (despite being used that way). It is however likely that designing an efficient representation suitable for volumes will require additional considerations about spatially distributing the caches. Another issue to overcome would be factoring the phase function into the cached distributions, since currently the method relies on multiple importance sampling to combine the cached information with surface BRDFs (causing issues if both of the distributions have appreciable anisotropic behaviour).

7.1.3 Principal-Ordinates Propagation (POP)

Chapter 5 proposes a novel discrete ordinates method capable of computing light transport in heterogeneous participating media which exhibit arbitrarily anisotropic scattering. This approach does not require any precomputations, which makes it well suitable for simulating dynamic and evolving media without extra considerations. Both the scattering and the radiance distributions are represented by the Henyey-Greenstein function; the latter (radiant energy) is propagated by a novel scheme using lattices oriented along estimated principal light directions. The employed representation also adapts to and pre-filters incident lighting.

In general the steps of the proposed method are physically-plausible. The employed heuristics introduce a certain bias but allow making design decisions that result in real-time performance on contemporary graphics hardware. Also the decomposition into a finite number of directions for distant light can only be successful if the variation of the initial light distribution is not too high; this however holds for all the HDR environment maps we tested in our experiments. In addition the prefiltered initialization can be used to avoid discretization artefacts in favour of a smooth approximation.

Temporal coherence The propagation stage of our method has turned out to be very stable and robust in respect to changes of the observer position and the medium itself. However, some temporal coherency issues still occur if the illumination discretisation changes too abruptly, either due to changes of the propagated importance function or simply changes in the lighting itself. These issues are typical to instant radiosity approaches [Keller 1997] or any other approach that relies on a discrete representation of the source illumination. An easy workaround is to apply some form of temporal filtering to the importance function (which can be as simple as averaging the importance maps across several consecutive frames). This works to an extent, but a more systematic solution would be to use an illumination discretisation method that explicitly enforces temporal coherence, for instance akin to the clustering approach of Prutkin et al. [2012] proposed in surface rendering.

Approximate principal ordinates Although we have not observed significant problems in this regard, theoretically in volumes with high density gradients the light distribution might not be faithfully reproduced by the HG basis aligned with the initial light direction. This is because strong heterogeneities of the medium may cause local deviations of the principal propagation direction, which globally aligned rectilinear grids cannot follow. An attractive way to deal with this problem would be using adaptive propagation grids. Such grids could still be treated as locally Euclidean, but

their topology would be 'warped' along the locally varying principal directions of propagation, as long as this variation remains smooth. We believe that this alignment should follow the gradient of the medium density, which amounts to solving the Eikonal equation commonly used in rendering media with varying refractive index (see for instance Ihrke et al. [2007]).

Surface boundaries Handling of opaque surfaces is a problem related to the above and one of the intrinsic limitations of DOM which our approach shares. While the light attenuated by the medium and reflected from surfaces back into the medium is handled by the local (spherical) propagation grids, higher-order interactions between surfaces and media are currently not accounted for. Although the first-order bounce is usually the most salient (since it reproduces most of the colour bleeding from surfaces to media), one way to reproduce multiple bounces would be to trace the local VPLs beyond the first surface interaction. This would be similar to volumetric instant radiosity [Engelhardt et al. 2010; Engelhardt et al. 2012], except that the energy from VPLs would not be gathered directly, but transported further in the local grids to account for additional higher scattering orders.

Alternatively it is possible to treat solid surfaces as strong volumetric scatterers and occluders, and represent them using voxelisation (either in a separate volume or merged with the medium density field). This is an approach used, e. g., by Kaplanyan and Dachsbacher [2010] and Billeter et al. [2012]. The main limiting factor is naturally the propagation grids' resolution (in our case typically in the range of 8^3–20^3), but it is conceivable that low-frequency (i. e., diffuse) illumination could be represented this way with the help of proper pre-filtering.

Convergence rate Another property inherent to all FE methods is their dependence on the discretisation accuracy. The resolution of every propagation grid is limited and the general limitations of discrete sampling apply – for finer details higher resolutions are required. Our approach deals with this issue by using multiple superimposed but relatively small propagation grids, in which a low number of iterations is sufficient to propagate most of the radiant energy. The upsampling and prefiltering steps also help to defer these issues and for typical volume data sets moderate grid resolutions have shown to be capable of handling a wide range of illumination conditions and medium properties.

However, on the basic level the relative complexity of our method is still $O(n^3)$ in space and at least $O(mn^3)$ in computation, where n is the relative scale of the smallest detail one wishes to reproduce. The additional term m expresses the increase in the number of iterations necessary for the solution to converge; typically $m = O(n^{1+p})$ with $p \in [0,1]$ (depending on the medium properties). It is the increase of the number of required iterations that makes DOM methods unusable for higher resolutions of the propagation lattice.

The classic solution to these problems is to use a hierarchical discretisation scheme, which is widely utilised in volume radiosity [Bhate 1993; Sillion 1994; Sillion 1995]. Stam [1995] suggested this approach to accelerate his local propagation method based on the diffusion approximation. In the context of POP, this would lead to a slightly increased spatial complexity of $O(n^3 \log^3 n)$, but would dramatically reduce the number of necessary iterations to $m = O(\log^{1+p} n)$. This however represents a challenging direction for future work – in contrast to isotropic transport (as described by diffusion, for instance), designing a hierarchical propagation scheme in the context of an anisotropic method is much more difficult. This is especially the case with POP, since the employed HG basis is non-linear.

A related characteristic of FE methods is that their rate of convergence depends not only on the propagation domain resolution but also on the optical thickness of the simulated medium; especially

for dense high-albedo media the number of iterations required for producing a converged solution might be prohibitively high (even in small domains). Employing ideas from similarity theory [Wyman et al. 1989] and utilising mathematical properties of the HG distribution (Section 2.3.3), this issue is effectively dealt with by using the proposed two-step propagation scheme (Section 5.4.4). Here only a small part of energy is propagated in full-resolution anisotropic mode. To our benefit, highly scattering media also decrease the anisotropy of the propagated light faster, allowing us to switch to the cheaper isotropic propagation mode earlier. The lighting frequencies resulting from the isotropic transport are by definition low and therefore lower-resolution propagation domains are sufficient here as well. It is likely that a hierarchical propagation scheme would increase the efficiency of this step even more.

7.1.4 Spectral Ray Differentials (SRD)

Chapter 6 introduces a method for reconstructing dispersion phenomena. Aside from providing a better mathematical description of these effects, the main purpose of SRD is reducing noise when rendering chromatic phenomena by offline ray and particle tracing methods as well as their interactive counterparts. In general these results extend the family of ray differentials [Igehy 1999], which have proven widely useful and are implemented in many contemporary renderers.

A common side-effect of image reconstruction techniques based on a first-order approximation, including the proposed approach, is that they introduce bias into the final result due to the finite footprint of the respective employed filter. This effectively introduces a certain amount of blur into the image, which however in most cases is visually more appealing than uncorrelated noise. On top of that the progressive formulation of SRD makes this bias consistently diminishing, so that in the limit the solution converges to the true, unbiased one. Additionally this enables a very rapid, near interactive way of working with dispersion – even in the context of non-interactive rendering methods. This is because the variable reconstruction scale (i. e., the filter footprint) makes it possible to first render dispersion fast and coarsely (with larger bias), and then by fluently decreasing this value achieving a more accurate, consistently converging solution.

In addition to the applications in standard Monte-Carlo rendering pipelines (Section 6.5), other specific areas of rendering might benefit from our work. Examples of these include simulation of light transport in optical systems, most prominently human eye [Kakimoto et al. 2004; Ritschel et al. 2009] and artificial lens systems [Hullin et al. 2011; Hullin et al. 2012]. The application of SRD is even more beneficial in rendering meta-materials or exotic materials with high IOR ranges, where the dispersion is much stronger. Therefore, while standard solutions will produce even more noise than usually (due to the higher divergence of dispersed light), spectral differentials will maintain their good convergence properties since their footprints grow proportionally with the magnitude of dispersion. Rendering such materials results in especially interesting images with a certain artistic value.

Convergence Our analysis shows that the convergence of SRD-based solutions is much better than their basic (point-based) counterparts. This naturally comes from the fact that each linear sample footprint generally covers many pixels in the resulting image. Since the size of the footprint is defined in the world space and not in the image space, an interesting side effect is that the convergence with SRD becomes even better if the resolution of the rendered image is higher (because the number of pixels covered by a particular sample footprint increases proportionally with resolution).

Robustness As in the previous work on differentials it is to be assumed that the differentiated

function is sufficiently smooth. Spectrally-varying transmission, absorption or emission do change the (radiance) function smoothness, especially if the respective spectra are peaky (i. e., they contain higher frequencies than the original illumination). Understanding how this affects spectral differentials definitely requires further investigation. It is certainly feasible that higher-order differentiation within the spectrum might solve the associated issues.

A related question is how to utilise spectral differentials in more general conditions. We have demonstrated their use in the comparatively simple context of ray and particle tracing methods where the reconstruction is done by gathering or splatting. Methods such as photon mapping with kernel density estimation, bi-directional path tracing, or even their combination – vertex connection and merging [Georgiev et al. 2012] would require additional considerations when extending their respective estimates into the spectral domain. Further evaluation would also be required to find robust ways of applying SRD on surfaces with complex geometry, e. g., strongly concave ones, where the proposed method based on geometrically aware multi-lateral weighting might not be sufficient.

Gathering versus scattering The presented gathering-based implementations currently assume that at a single image location the dispersive caustic is dominated by photons that are refracted mostly by the same sequence of interfaces (or different interfaces with similar relative orientations). This is not the case for example when two distinct caustics cross each other, and for overlapping transparent surfaces. In such areas, our approach gracefully degrades to the unfiltered original because the size of the accumulated footprint tends to diminish (especially if the incoming rays' differentials are close to perpendicular). The scattering solution does not suffer from this limitation and is to be preferred, although its implementation is more application-specific. Consequently, software limitations or other requirements can make gathering a necessary alternative, so it is useful to have it included in the analysis.

Parametrisation Same as all progressive photon mapping approaches our algorithm is controlled by a scaling parameter. Although the intuition behind it is easily grasped, the actual value optimal in a given scene configuration is not easy to determine without prior experimentation. A very interesting direction for future work is therefore an automatic determination of an optimal value for this parameter. This might be based on a suitable statistical error metric, similar to the work of Hachisuka et al. [2010]. Conceivably, different values should be used in different parts of the scene (or even per-pixel) instead of a single global value, depending on the local energy density from the incoming illumination.

Beyond dispersion A natural course of action would be to extend SRD beyond just dispersive refraction. Other existing sources of fascinating chromatic effects are of interest to computer graphics, most notably diffraction and interference. These phenomena are however more complex, since they require wave-optical considerations, and as such deriving corresponding differentials will likely be more involved. In spite of that it might still be feasible, given that at least diffraction can (to a good extent) be simulated within ray-based rendering frameworks [Stam 1999; Cuypers et al. 2012]. Even dispersion itself requires additional work, since transparent media can cause light paths to curve if their refractive varies smoothly, which is something that SRD ignore at the moment. Such extension should be possible by differentiating the Eikonal equation that describes this behaviour.

As already mentioned, spectral differentials belong to the group of ray differentials which comprises image-space differentials (i. e., the classic differentials of Igehy [1999]), light-space differentials (dubbed photon differentials by Schjøth et al. [2007]), path differentials (describing glossy reflection and refraction [Suykens and Willems 2001]), and other methods that employ differential-like

techniques (e. g., by Fabianowski and Dingliana [2009]). It is logical and highly desirable to consider the use of differentials to reconstruct other phenomena, especially distribution effects which tend to contribute the most to the variance in MC solutions. One of the ways to reconstruct such effects is axis-aligned and sheared filtering, which most notably utilises frequency analysis [Durand et al. 2005] to explain effects like soft shadows [Mehta et al. 2012], diffuse inter-reflection [Mehta et al. 2013], motion blur and depth of field [Belcour et al. 2013; Mehta et al. 2014], their combinations [Yan et al. 2014], and to a certain extent even scattering [Belcour et al. 2014]. While these methods achieve tremendous results in noise reduction, their mathematical complexity requires many approximations to be made during the derivation but also reconstruction itself, especially when reconstructing multiple different phenomena together. The major advantage of ray differentials is that they are tracked individually for each ray or path, so that the reconstruction can be performed with considerations about each individual sample, in contrast to the above approaches which perform the filtering based on local statistics about incoming light, scene configuration, etc. Ultimately, we envision the derivation of a *total ray differential*, which would consider the full dimensionality of the radiance function and track the differentials in respect to all its variables. Reconstructing the illumination would then entail combining all this information in a joint multi-dimensional filtering step.

Finally, given the existence of differentiation schemes that describe distributed effects (primarily glossy surface reflections [Chen and Arvo 2000; Suykens and Willems 2001]), it seems a realistic proposition to derive *scattering differentials* since (anisotropic) scattering shares many similarities with glossy reflection. In addition to reconstruction, scattering differentials could be an important ingredient in designing higher-level rendering primitives in the context of volumetric rendering. Efforts in this direction have already been made, mainly represented by methods utilising photon beams [Jarosz et al. 2011a], diffusion of such beams [Habel et al. 2013], and virtual beam lights [Novák et al. 2012a]. This spiritually follows the classic work in surface rendering that has striven to enrich rays as a rendering primitive (such as beams [Heckbert and Hanrahan 1984], cones [Amanatides 1984], pencils [Shinya et al. 1987] and voxel cones [Crassin et al. 2011]). It is an interesting subject for future work whether such scattering differentials make a conceptual sense, and if so how can they be derived and applied to improve the quality of volumetric rendering solutions.

7.1.5 General Discussion

The methods presented in this thesis have similar applications but are largely complementary in the types of conditions they target. The common important trait they share is a complete independence on any precomputation that many other interactive methods rely on. This enables them to be applied in any conditions where the rendered scene is *dynamic* and changes *unpredictably*. It should also be understood that all the documented implementations are working prototypes, implying that numerous low-level optimisations could still be performed and therefore the performance measurement figures should be taken as rather conservative estimates. In fact, none of the methods exhibits any significant bottleneck, so a consistently good GPU occupancy is achieved. Thanks to that it is to be assumed that the performance of all the proposed methods will scale well with the constantly increasing processing power of graphics hardware.

- **Screen-space scattering** is a very fast method that achieves computation times of only a few milliseconds per frame (and a significant part of that comes from the use of a costly double bi-cubic interpolation). As such it qualifies for use in high-performance real-time applications that may require the rendering of homogeneous media. These are primarily computer games

and training simulators, but possibly also image editing tools since SSS does not require a 3D representation of the scene to operate.

- **Amortised photon mapping** runs in about 10–20 milliseconds per frame, for a single cloud. This still qualifies it for real-time applications with higher computational resources, but not for games which need to assume an average consumer hardware. However, APM degrades gracefully with lower resolutions of the caching lattice, and, at the cost of a linear memory increase the photon tracing cost can be amortised across longer time periods. Combined with the proposed hierarchical caching scheme it is conceivable that the total per-frame cost could drop below 10 ms, which is affordable even in gaming.
- **Principal-ordinates propagation** reaches speeds of about 30–60 milliseconds per frame, at least for the tested scenes. This will not be sufficient for high-performance applications (at least within the next few years) but can be used for applications which require high-quality rendition of volumetric media but also interactive feedback. Prime examples of these are scientific and medical visualisations. It should be noted that for the somewhat increased computation price (measured in the context of interactive applications) POP delivers unprecedented qualitative results. To our knowledge no other existing method with comparable performance can reproduce the effects of anisotropic multiple scattering for such a broad range of media, and even non-interactive rendering algorithms struggle with some of the lighting phenomena that POP can handle.
  Noticeably, POP targets a superset of media than what APM can handle. In terms of quality APM is surpassed by POP even for clouds, simply because the latter uses multiple lobes for representing the radiance directionality. On the other hand APM is several times faster (even with the relatively conservative settings used for its evaluation) because it exploits the assumptions of temporal coherence in clouds that POP cannot rely on.
- Finally, **spectral ray differentials** target all applications that are based on tracing rays or ray-like primitives. These can be very accurate and non-interactive stochastic MC methods but also interactive algorithms based, e. g., on photon splatting. The overhead of tracking SRD is usually negligible, since it takes only a handful of arithmetic instructions per each dispersive interaction to update them. The cost of their reconstruction varies for each applied method, but again is never a very significant factor. Most importantly, SRD typically lead to an order-of-magnitude less noise compared to the baseline methods, and the progressive diminishing of their approximation bias enables a seamless transition from an interactive manipulation mode to a non-interactive regime that produces an accurate converged solution.

Scattering and blurring At this point we can see that on the macroscopic scale, scattering shares many similarities with blurring and filtering in general. For instance, taking a closer look at the relations between the screen-space scattering and principal-ordinates propagation methods, it can be seen that they deal with similar problems. The pyramidal approach applied to fast filtering in SSS is designed to overcome the innate complexity of computing a response of an arbitrarily large filtering kernel. There are of course some limitations of this approximate approach – the partial kernels contained in each of the filtering pyramid levels have an increasing but finite size, while the Gaussian filter that is being approximated is infinite for even very small standard deviations. The difference can be apparent for light sources with very high brightness but small distance to the camera. Aside from that, however, the filtering pyramid successfully fulfils its purpose: to connect all image pixels and distribute energy between them in a logarithmic number of steps. As explained in Section 7.1.3 the propagation scheme of POP suffers from virtually the same issue – how to efficiently distribute energy between the lattice voxels without requiring a super-linear number of iterations to do so. Although the information contained in the propagation lattices is richer than what images represent,

it is likely that some inspirations can be taken from pyramidal filtering when designing a hierarchical propagation solution for the anisotropic light transport.

Furthermore, by utilising this knowledge one could design a hybrid method for volumetric light transport that, akin to POP, propagates light in principal lattices, but instead of an iterative approach performs only a single forward sweep through the medium. Unsurprisingly, methods based on similar considerations already exist. Light propagation can be done in a single blurring sweep but with neglecting an entire backward portion of the path space [Kniss et al. 2002; Riley et al. 2004]. An accurate propagation in this sense on the other hand requires multiple sweeps [Fattal 2009]. Bi-directional scattering-surface reflectance distribution functions (BSSRDFs) require no sweeps at all, but cannot handle heterogeneous media, suffer from difficult boundary conditions such as concave object boundaries, and despite continuous improvements [Frisvad et al. 2014] are unable to represent fully anisotropic scattering. By performing a sweep through the medium, the shortcomings of the BSSRDF-based approaches could be removed if the blurring kernel used in SSS was extended to represent directional radiance distributions as well. Removing the limitation to only a forward portion of the transport would require a global knowledge about the medium, which could however be provided as simply as by building a MIP-map of the medium density field.

Perception of media The aspect of volumetric rendering which we have not considered in sufficient detail is the perception of effects created by participating media. While measuring and validating the methods against accurate reference solutions is definitely important, in the end the resulting images will be observed by humans (in most cases anyway). Early work by Sundstedt et al. [2007] has identified that—as can be expected—the most salient regions within participating media are those with low attenuation (in other words, good visibility) and edges of objects and lighting features (most notably light shafts and caustics). Objective frequency analysis supports this as well [Belcour et al. 2014]. The situation is however more complex than that. As shown by Gkioulekas et al. [2013] and confirmed by our observations, even the simplest 1D space of HG phase functions is highly non-linear, which only gets stronger as one considers more general phase functions and all other basic medium parameters. Understanding the perception of media under such conditions is not trivial. Recent work by Nguyen et al. [2015b; 2015a] has examined the topology of other non-linear spaces, specifically colours and shapes, showing methods of extracting lower dimensional embeddings of these spaces. Identifying such embeddings provides insights into the structure of their parent spaces and enables users to more easily navigate within them. We believe that a similar analysis could be very beneficial in our situation as well.

Media editing In a direct relation to the perception of participating media are their manipulation and editing. In addition to the direct importance of editing for rendering content creation, it impacts areas such as constructive 3D fabrication [Papas et al. 2013], since most objects made this way have some translucent qualities, desirable or not. While acquiring shapes of most solid media is relatively established, obtaining density fields of fluid phenomena is significantly more difficult. This is especially true for large-scale media such as clouds, which are practically impossible to acquire in a laboratory and difficult to simulate the dynamics of [Heus 2008]. For this reason techniques for generating cloud shapes from single images have been proposed [Dobashi et al. 2010; Dobashi et al. 2012], but still a lot of space for improvement remains.

Direct manipulation of existing data is not straightforward either, due to their three-dimensional nature and expertise that is required from the user to understand the impact of changing parameters defined in terms of light transport physics. While direct edits to the medium parameters can be visualised interactively [Hašan and Ramamoorthi 2013], inverse editing guided by simple user

painting is so far only feasible under the strong constraint of single-scattering, and still does not achieve interactive responses [Klehm et al. 2013; Klehm et al. 2014]. This is caused by the overwhelming dimensionality of this problem. Additionally, the existing methods natively allow only for changing the medium emissivity and albedo, and the more interesting parameters such as density and anisotropy remain unsolved. It is therefore apparent that future work in this area should ultimately strive to provide interactive inverse editing of as broad range of parameters as possible. It is arguably safe to assume that solving this task will not be feasible in the near future, as it will require developing our understanding of media, more advanced mathematical tools, and last but not least, computing hardware.

Stereo rendering In recent years, an increasing popularity has been gained by stereoscopic display technologies, such as conventional or head-mounted displays and projection systems. While stereoscopic rendering can be trivially implemented as an interleaved generation of two images with an appropriate horizontal offset, in practice most applications generate the stereo pair from a single image, where this is aided by a perceptually motivated disparity field derived from the rendered objects' depth [Didyk et al. 2011]. Similar approaches are also necessary in decoupled rendering architectures, where most computation is done server-side and sent to a client for displaying. The potential latency of such systems needs to be compensated for without having full knowledge about the rendered scene [Lochmann et al. 2014]. The key step of these approaches is *warping*, which essentially hallucinates the information needed to synthesize novel views based on the already rendered frame. When considering only (opaque) surfaces, one can rely on the existence of a surface single point that the virtual camera captures for each pixel. This is however not the case with media, where each pixel corresponds to an entire linear integral through the medium. Synthesizing novel views of volumetric media is therefore a challenging but interesting direction for further investigation.

7.2 Closing Messages

This thesis is built on the assumption that synthesizing realistic images should be based on physical laws. It argues that, for designing efficient rendering methods, these laws need to be understood to the extent required to build the intuition about their behaviour on a macroscopic scale. One therefore needs to go beyond mere optimisation of existing rendering frameworks – it is necessary to develop more abstracted higher-level models, which are simple and intuitive, but still only make sensible assumptions within the boundaries dictated by physics.

The focus of the presented work has been on participating media as one of the most important constituents of our visual perception of the world. A lot of research effort has already been invested into developing more narrow-oriented techniques that focus on specific phenomena. Thanks to these contributions the graphics community now has a good understanding of many types of volumetric materials and media. The logical future course of action is using this understanding to design more general approaches that would unify the way how such media are handled in various graphics workflows.

In a broader sense, we believe that the effect of complex lighting on dynamic participating media is an exciting visual phenomenon that deserves more dedicated research, even outside the classic topics, for instance to better understand human perception of volumetric illumination or the artistic practice which can be applied to depict it.

Bibliography (Own Work)

ELEK, O., BAUSZAT, P., RITSCHEL, T., MAGNOR, M. AND SEIDEL, H.-P. (2014a): Progressive spectral ray differentials. In *Proceedings of International Workshop on Vision, Modeling and Visualization*, 151–58 7, 9

ELEK, O., BAUSZAT, P., RITSCHEL, T., MAGNOR, M. AND SEIDEL, H.-P. (2014b): Spectral ray differentials. *Computer Graphics Forum (Proceedings of EGSR)*, 33, 113–22 7, 9

ELEK, O. AND KMOCH, P. (2010): Real-time spectral scattering in large-scale natural participating media. In *Proceedings of Spring Conference on Computer Graphics*, 77–84 57

ELEK, O., RITSCHEL, T., DACHSBACHER, C. AND SEIDEL, H.-P. (2014c): Interactive light scattering with principal-ordinate propagation. In *Proceedings of Graphics Interface*, 87–94 7, 8

ELEK, O., RITSCHEL, T., DACHSBACHER, C. AND SEIDEL, H.-P. (2014d): Principal-ordinates propagation for real-time rendering of participating media. *Computers & Graphics*, 45, 28–39 7, 8

ELEK, O., RITSCHEL, T. AND SEIDEL, H.-P. (2013): Real-time screen-space scattering in homogeneous environments. *IEEE Computer Graphics and Applications*, 33, 53–65, Special issue: Scattering 7

ELEK, O., RITSCHEL, T., WILKIE, A. AND SEIDEL, H.-P. (2012a): Interactive cloud rendering using temporally-coherent photon mapping. In *Proceedings of Graphics Interace*, 141–48 7

ELEK, O., RITSCHEL, T., WILKIE, A. AND SEIDEL, H.-P. (2012b): Interactive cloud rendering using temporally coherent photon mapping. *Computers & Graphics*, 36, 1109–18 7

Bibliography

ADAMS, A., GELFAND, N., DOLSON, J. AND LEVOY, M. (2009): Gaussian KD-trees for fast high-dimensional filtering. *ACM Trans. Graph.* 28, 21:1–21:12 38

AIRIEAU, B., MENEVEAUX, D., BRIDAULT, F. AND BLASI, P. (2011): Photon streaming for interactive global illumination in dynamic scenes. *Visual Computer*, 27, 229–240 32, 22

AKENINE-MÖLLER, T., HAINES, E. AND HOFFMANN, N. (2008): Real-time rendering. 3rd edition. 6

AMANATIDES, J. (1984): Ray tracing with cones. *SIGGRAPH Comput. Graph.* 18, 129–35 122

ARNOLD (2015): Arnold renderer., By Solid Angle 28, 18

ARVO, J. (1986): Backward ray tracing. In *In SIGGRAPH Course Notes* 26, 35, 16, 25

BAI, L., WU, Z.-S., TANG, S.-Q., LI, M., XIE, P.-H. AND WANG, S.-M. (2011): Study on phase function in Monte Carlo transmission characteristics of poly-disperse aerosol. *J. Optical Engineering*, 50, 016002 21, 11

BARAN, I., CHEN, J., RAGAN-KELLEY, J., DURAND, F. AND LEHTINEN, J. (2010): A hierarchical volumetric shadow algorithm for single scattering. In *ACM Trans. Graph. (Proc. SIGGRAPH)* Volume 29, 33, 23

BAUSZAT, P., EISEMANN, M. AND MAGNOR, M. (2011): Guided image filtering for interactive high-quality global illumination. *Comp. Graph. Forum*, 30, 1361–68 38

BELCOUR, L., BALA, K. AND SOLER, C. (2014): A local frequency analysis of light scattering and absorption. *ACM Trans. Graph.* 33, 5 36, 122, 124, 26

BELCOUR, L., SOLER, C., SUBR, K., HOLZSCHUCH, N. AND DURAND, F. (2013): 5D covariance tracing for efficient defocus and motion blur. *ACM Trans. Graph.* 32, 31 36, 122, 26

BERNABEI, D., HAKKE PATIL, A., BANTERLE, F., GANOVELLI, F., DI BENEDETTO, M., PATTANAIK, S. AND SCOPIGNO, R. (2011): A parallel architecture for interactive rendering of scattering and refraction effects. *Computer Graphics and Applications, IEEE*, 32, 34–43 33, 23

BERTALMIO, M., FORT, P. AND SANCHEZ-CRESPO, D. (2004): Real-time, accurate depth of field using anisotropic diffusion and programmable graphics cards. In *Proc. 3DPVT* 47

BHATE, N. (1993): Application of rapid hierarchical radiosity to participating media. In *Proc. of Advanced Techniques in Animation, Rendering, and Visualization*, 43–53 29, 119, 19

BHATE, N. AND TOKUTA, A. (1992): Photorealistic volume rendering of media with directional scattering. In *Eurographics Workshop on Rendering*, 227–46 29, 19

BILLETER, M., SINTORN, E. AND ASSARSSON, U. (2012): Real-time multiple scattering using light propagation volumes. In *Proc. I3D* 33, 70, 119, 23

BINZONI, T., LEUNG, T. S., GANDJBAKHCHE, A. H., RÜFENACHT, D. AND DELPY, D. T. (2006): The use of the Henyey–Greenstein phase function in Monte Carlo simulations in biomedical optics. *Physics in Medicine and Biology*, 51, N313 21, 11

BLASI, P., LE SAEC, B. AND SCHLICK, C. (1993): A rendering algorithm for discrete volume density objects. *Computer Graphics Forum*, 12, 201–10 21, 11

BLINN, J. F. (1982): Light reflection functions for simulation of clouds and dusty surfaces. *SIGGRAPH Comput. Graph.* 16, 21–29 25, 15

BORLUM, J., CHRISTENSEN, B. B., KJELDSEN, T. K., MIKKELSEN, P. T., NOE, K. O., RIMESTAD, J. AND MOSEGAARD, J. (2011): SSLPV: Subsurface Light Propagation Volumes. In *Proc. HPG* 33, 23

BORN, M. F. AND WOLF, E. (1999): Principles of optics. 7th edition. 2, 11, 34, 1, 24

BOULT, T. E. AND WOLBERG, G. (1992): Correcting chromatic aberrations using image warping. In *Proc. CVPR*, 684–87 36, 26

BOUTHORS, A. (2008): Real-time realistic rendering of clouds. Ph. D thesis, Université Joseph-Fourier 33

BOUTHORS, A., NEYRET, F. AND LEFEBVRE, S. (2006): Real-time realistic illumination and shading of stratiform clouds. In *Proc. Workshop on Natural Phenomena* 21, 33, 11, 23

BOUTHORS, A., NEYRET, F., MAX, N., BRUNETON, E. AND CRASSIN, C. (2008): Interactive multiple anisotropic scattering in clouds. In *Proc. I3D* 21, 33, 11, 23

BRUNETON, E. AND NEYRET, F. (2008): Precomputed atmospheric scattering. In *Proc. EGSR* 57

BURT, P. J. (1981): Fast filter transforms for image processing. *Computer Graphics and Image Processing*, 16, 20–51 38, 39, 42, 46, 47, 51

CABRAL, B. AND LEEDOM, L. C. (1993): Imaging vector fields using line integral convolution. In *Proc. SIGGRAPH*, 263–70 106

CEREZO, E., PÉREZ-CAZORLA, F., PUEYO, X., SERON, F. J. AND SILLION, F. (2005): A survey on participating media rendering techniques. *Visual Comput.* 21, 303–28 25, 15

CHANDRASEKHAR, S. (1960): Radiative transfer. 4, 5, 24, 29, 33, 78, 14, 19, 23

CHEN, M. AND ARVO, J. (2000): Perturbation methods for interactive specular reflections. *IEEE Trans. Vis. Comput. Graph.* 6, 253–64 122

CIE (1932): Commission internationale de l'Eclairage proceedings. Cambridge University Press 101

COHEN, M. F., WALLACE, J. AND HANRAHAN, P. (1993): Radiosity and realistic image synthesis. 29, 19

COLEMAN, W. (1968): Mathematical verification of a certain Monte Carlo sampling technique and applications of the technique to radiation transport problems. In *Nuclear Science and Engineering*, 76–81 23, 13

COOK, R. L., PORTER, T. AND CARPENTER, L. (1984): Distributed ray tracing. *SIGGRAPH Comput. Graph.* 18, 137–45 39, 29

CORNETTE, W. M. AND SHANKS, J. G. (1992): Physically reasonable analytic expression for the single-scattering phase function. *Applied Optics*, 31 (16), 3152–60 21, 11

CRASSIN, C., NEYRET, F., SAINZ, M., GREEN, S. AND EISEMANN, E. (2011): Interactive indirect illumination using voxel cone tracing. *Computer Graphics Forum*, 30 (7), 1921–30 122

CUYPERS, T., HABER, T., BEKAERT, P., OH, S. B. AND RASKAR, R. (2012): Reflectance model for diffraction. *ACM Trans. Graph.* 31, 122 121

DACHSBACHER, C. AND STAMMINGER, M. (2005): Reflective shadow maps. In *Proc. of I3D*, 203–13 85

DAMMERTZ, H., KELLER, A. AND LENSCH, H. (2010): Progressive point-light-based global illumination. *Comp. Graph. Forum*, 29, 2504–15 28, 18

DECORO, C., WEYRICH, T. AND RUSINKIEWICZ, S. (2010): Density-based outlier rejection in Monte Carlo rendering. *Computer Graphics Forum (Proc. Pacific Graphics)*, 29, 7 38

DEERING, M., WINNER, S., SCHEDIWY, B., DUFFY, C. AND HUNT, N. (1988): The triangle processor and normal vector shader: a VLSI system for high performance graphics. *SIGGRAPH Comput. Graph.* 22, 21–30 39, 29

DELBRACIO, M., MUSÉ, P., BUADES, A., CHAUVIER, J., PHELPS, N. AND MOREL, J.-M. (2014): Boosting monte carlo rendering by ray histogram fusion. *ACM Trans. Graph.* 33, 8 38

D'EON, E., LUEBKE, D. AND ENDERTON, E. (2007): Efficient rendering of human skin. In *Rendering Techniques* 30, 20

D'EON, E. AND IRVING, G. (2011): A quantized-diffusion model for rendering translucent materials. *ACM Trans. Graph.* 30, 56 30, 20

DIDYK, P., RITSCHEL, T., EISEMANN, E., MYSZKOWSKI, K. AND SEIDEL, H.-P. (2011): A perceptual model for disparity. *ACM Trans. Graph. (Proc. of SIGGRAPH)* 30 125

DMITRIEV, K., BRABEC, S., MYSZKOWSKI, K. AND SEIDEL, H.-P. (2002): Interactive global illumination using selective photon tracing. In *Proc. EGRW*, 25–36 32, 22

DOBASHI, Y., IWASAKI, W., ONO, A., YAMAMOTO, T., YUE, Y. AND NISHITA, T. (2012): An inverse problem approach for automatically adjusting the parameters for rendering clouds using photographs. *ACM Trans. Graph.* 31, 145 124

DOBASHI, Y., SHINZO, Y. AND YAMAMOTO, T. (2010): Modeling of clouds from a single photograph. *Computer Graphics Forum*, 29, 2083–90 124

DONNER, C. AND JENSEN, H. W. (2005): Light diffusion in multi-layered translucent materials. *ACM Trans. Graph.* 24, 1032–39 30, 20

DONNER, C., LAWRENCE, J., RAMAMOORTHI, R., HACHISUKA, T., JENSEN, H. W. AND NAYAR, S. (2009): An empirical BSSRDF model. *ACM Trans. Graph.* 28, 30 30, 20

DONNER, C., WEYRICH, T., D'EON, E., RAMAMOORTHI, R. AND RUSINKIEWICZ, S. (2008): A layered, heterogeneous reflectance model for acquiring and rendering human skin. *ACM Trans. Graph.* 27, 140 30, 20

DURAND, F., HOLZSCHUCH, N., SOLER, C., CHAN, E. AND SILLION, F. (2005): A frequency analysis of light transport. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 24 (3), 1115–26 36, 122, 26

DUTRÉ, P., BEKAERT, P. AND BALA, K. (2006): Advanced global illumination. 2nd edition. 11, 25, 1, 15

EGAN, K., HECHT, F., DURAND, F. AND RAMAMOORTHI, R. (2011): Frequency analysis and sheared filtering for shadow light fields of complex occluders. *ACM Trans. Graph.* 30, 9 36, 26

EISEMANN, E. AND DURAND, F. (2004): Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 23, 673–78 101

ENGELHARDT, T. AND DACHSBACHER, C. (2010): Epipolar sampling for shadows and crepuscular rays in participating media with single scattering. In *Proc. I3D* 33, 116, 23

ENGELHARDT, T., NOVÁK, J. AND DACHSBACHER, C. (2010): Instant multiple scattering for interactive rendering of heterogeneous participating media. Karlsruhe Institute of Technology – Technical report 27, 85, 119, 17

ENGELHARDT, T., NOVAK, J., SCHMIDT, T.-W. AND DACHSBACHER, C. (2012): Approximate bias compensation for rendering scenes with heterogeneous participating media. *Comp. Graph. Forum*, 31, 2145–54 27, 85, 119, 17

FABIANOWSKI, B. AND DINGLIANA, J. (2009): Interactive global photon mapping. *Comp. Graph. Forum (Proc. EGSR)*, 28 (4), 1151–59 36, 122, 27

FATTAL, R. (2009): Participating media illumination using light propagation maps. *ACM Trans. Graph.* 28, 7 29, 124, 19

FEYNMAN, R. P. AND HIBBS, A. R. (1965): Quantum mechanics and path integrals. 30, 20

FISHER, R. (1953): Dispersion on a sphere. *Proc. of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 217, 295–305 21, 11

FRISVAD, J. R., HACHISUKA, T. AND KJELDSEN, T. K. (2014): Directional dipole model for subsurface scattering. *ACM Trans. Graph.* 34, 5 30, 124, 20

GARDNER, G. Y. (1985): Visual simulation of clouds. In *Proc. SIGGRAPH*, 297–304 33, 23

GASTAL, E. S. L. AND OLIVEIRA, M. M. (2011): Domain transform for edge-aware image and video processing. *ACM Trans. Graph.* 30, 69 38

GASTAL, E. S. L. AND OLIVEIRA, M. M. (2012): Adaptive manifolds for real-time high-dimensional filtering. *ACM Trans. Graph.* 31, 33 38

GEIST, R., RASCHE, K., WESTALL, J. AND SCHALKOFF, R. (2004): Lattice-Boltzmann lighting. In *Proc. EGSR*, 355–62 5, 29, 19

GEORGIEV, I., KŘIVÁNEK, J., DAVIDOVIČ, T. AND SLUSALLEK, P. (2012): Light transport simulation with vertex connection and merging. *ACM Trans. Graph.* 31, 192 121

GKIOULEKAS, I., XIAO, B., ZHAO, S., ADELSON, E., ZICKLER, T. AND BALA, K. (2013): Understanding the role of phase function in translucent appearance. *ACM Trans. Graph.* 32 21, 124, 11

GONZALEZ, R. AND WOODS, R. (2001): Digital image processing. 2nd edition. 37

GORAL, C. M., TORRANCE, K. E., GREENBERG, D. P. AND BATTAILE, B. (1984): Modeling the interaction of light between diffuse surfaces. *SIGGRAPH Comput. Graph.* 18, 213–22 29, 19

GREEN, G. (1828): An essay on the application of mathematical analysis to the theories of electricity and magnetism. 31, 21

GREGER, G., SHIRLEY, P., HUBBARD, P. AND GREENBERG, D. (1998): The irradiance volume. *Computer Graphics and Applications, IEEE*, 18 (2), 32–43 59

HABEL, R., CHRISTENSEN, P. H. AND JAROSZ, W. (2013): Photon beam diffusion: a hybrid monte carlo method for subsurface scattering. *Computer Graphics Forum* 32 30, 122, 20

HABER, J., MAGNOR, M. AND SEIDEL, H.-P. (2005): Physically-based simulation of twilight phenomena. *ACM Trans. Graph.* 24, 1353–73 29, 19

HACHISUKA, T., JAROSZ, W., GEORGIEV, I., KAPLANYAN, A. AND NOWROUZEZAHRAI, D. (2013): State of the art in photon density estimation. In *SIGGRAPH Asia Courses* 27, 17

HACHISUKA, T., JAROSZ, W. AND JENSEN, H. W. (2010): A progressive error estimation framework for photon density estimation. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 29, 144 121

HACHISUKA, T. AND JENSEN, H. W. (2009): Stochastic progressive photon mapping. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 28, 141 28, 17

HACHISUKA, T., OGAKI, S. AND JENSEN, H. W. (2008): Progressive photon mapping. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 27, 130 28, 17

HANRAHAN, P. AND KRUEGER, W. (1993): Reflection from layered surfaces due to subsurface scattering. In *Proc. SIGGRAPH* 26, 16

HAŠAN, M. AND RAMAMOORTHI, R. (2013): Interactive albedo editing in path-traced volumetric materials. *ACM Transactions on Graphics*, 32, 11 124

HE, K., SUN, J. AND TANG, X. (2010): Guided image filtering. In *Proc. ECCV* 38

HECKBERT, P. S. AND HANRAHAN, P. (1984): Beam tracing polygonal objects. *SIGGRAPH Comput. Graph.* 18, 119–27 122

HENYEY, L. G. AND GREENSTEIN, J. L. (1941): Diffuse radiation in the Galaxy. *Astrophysical Journal*, 93, 70–83 19, 9

HERZOG, R., HAVRAN, V., KINUWAKI, S., MYSZKOWSKI, K. AND SEIDEL, H.-P. (2007): Global illumination using photon ray splatting. *Comp. Graph. Forum (Proc. Eurographics)*, 26 (3), 503–13 32, 22

HEUS, T. (2008): On the edge of a cloud. Ph. D thesis, Technische Universiteit Delft 124

HOTTEL, H. C. AND SAROFIM, A. F. (1967): Radiative transfer. 29, 19

HOŠEK, L. AND WILKIE, A. (2012): An analytic model for full spectral sky-dome radiance. *ACM Transactions on Graphics* 31 32, 22

HUANG, J., BOUBEKEUR, T., RITSCHEL, T., HOLLÄNDER, M. AND EISEMANN, E. (2011): Separable approximation of ambient occlusion. In *Proc. of Eurographics Short Papers* 39, 29

HULLIN, M., EISEMANN, E., SEIDEL, H.-P. AND LEE, S. (2011): Physically-based real-time lens flare rendering. *ACM Trans. Graph.* 30, 108 35, 120, 25

HULLIN, M. B., HANIKA, J. AND HEIDRICH, W. (2012): Polynomial optics: A construction kit for efficient ray-tracing of lens systems. *Comp. Graph. Forum (Proc. EGSR)*, 31, 1375–83 120

HULST, H. C. VAN DE (1981): Light scattering by small particles. 12, 13, 16, 1, 3, 5

IGEHY, H. (1999): Tracing ray differentials. In *Proc. SIGGRAPH* 36, 37, 94, 95, 98, 120, 121, 26, 27

IHRKE, I., ZIEGLER, G., TEVS, A., THEOBALT, C., MAGNOR, M. AND SEIDEL, H.-P. (2007): Eikonal rendering: efficient light transport in refractive objects. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 26, 59 33, 59, 119, 23

JAKOB, W. (2010): Mitsuba renderer. 18, 8

JAKOB, W., REGG, C. AND JAROSZ, W. (2011): Progressive expectation-maximization for hierarchical volumetric photon mapping. *Computer Graphics Forum*, 30 (4), 1287–97 118

JAROSZ, W. (2014): Private communication. 100

JAROSZ, W., DONNER, C., ZWICKER, M. AND JENSEN, H. W. (2008a): Radiance caching for participating media. *ACM Trans. Graph.* 27, 7 27, 58, 59, 17

JAROSZ, W., NOWROUZEZAHRAI, D., SADEGHI, I. AND JENSEN, H. W. (2011a): A comprehensive theory of volumetric radiance estimation using photon points and beams. *ACM Trans. Graph.* 30, 5 27, 54, 57, 122, 17

JAROSZ, W., NOWROUZEZAHRAI, D., THOMAS, R., SLOAN, P.-P. AND ZWICKER, M. (2011b): Progressive photon beams. *ACM Trans. Graph.* 30, 181 28, 94, 99, 100, 17

JAROSZ, W., ZWICKER, M. AND JENSEN, H. W. (2008b): The beam radiance estimate for volumetric photon mapping. *Computer Graphics Forum (Proc. Eurographics)*, 27, 557–66 27, 17

JENKINS, F. AND WHITE, H. (2001): Fundamentals of optics. 97

JENSEN, H. W. (1995): Importance driven path tracing using the photon map. In *Rendering Techniques* 117

JENSEN, H. W. (1996): Global illumination using photon maps. In *Proc. of EGWR*, 21–30 26, 28, 16, 17

JENSEN, H. W. (2001): Realistic image synthesis using photon mapping. 35, 25

JENSEN, H. W. AND BUHLER, J. (2002): A rapid hierarchical rendering technique for translucent materials. *ACM Trans. Graph.* 21, 576–81 30, 20

JENSEN, H. W. AND CHRISTENSEN, P. H. (1998): Efficient simulation of light transport in scenes with participating media using photon maps. In *Proc. SIGGRAPH*, 311–20 26, 58, 16

JENSEN, H. W., MARSCHNER, S. R., LEVOY, M. AND HANRAHAN, P. (2001): A practical model for subsurface light transport. In *Proc. SIGGRAPH* 27, 30, 17, 20

JIMENEZ, J., SUNDSTEDT, V. AND GUTIERREZ, D. (2009): Screen-space perceptual rendering of human skin. *ACM Trans. Appl. Percept.* 6, 23 39, 29

JIMÉNEZ, J.-R., MYSZKOWSKI, K. AND PUEYO, X. (2005): Interactive global illumination in dynamic participating media using selective photon tracing. In *Proc. SCCG*, 211–18 32, 22

JOHNSON, M. K. AND FARID, H. (2006): Exposing digital forgeries through chromatic aberration. In *Proc. ACM Multimedia and Security*, 48–55 36, 26

JÖNSSON, D., SUNDÉN, E., YNNERMAN, A. AND ROPINSKI, T. (2014): A survey of volumetric illumination techniques for interactive volume rendering. *Computer Graphics Forum*, 33, 27–51 34, 24

KAJIYA, J. T. AND KAY, T. L. (1989): Rendering fur with three dimensional textures. *SIGGRAPH Comput. Graph.* 23, 271–80 21, 11

KAJIYA, J. T. (1986): The rendering equation. *SIGGRAPH Comput. Graph.* 20, 143–50 4, 25, 26, 35, 15, 16

KAJIYA, J. T. AND HERZEN, B. P. V. (1984): Ray tracing volume densities. In *Proc. SIGGRAPH*, 165–74 29, 19

KAKIMOTO, M., MATSUOKA, K., NISHITA, T., NAEMURA, T. AND HARASHIMA, H. (2004): Glare generation based on wave optics. In *Proc. Pacific Graphics*, 133–40 35, 120, 25

KALANTARI, N. K. AND SEN, P. (2013): Removing the noise in Monte Carlo rendering with general image denoising algorithms. *Computer Graphics Forum (Proc. Eurographics)* 32 38

KANG, S. B. (2007): Automatic removal of chromatic aberration from a single image. In *Proc. CVPR* 36, 26

KAPLANYAN, A. AND DACHSBACHER, C. (2010): Cascaded light propagation volumes for real-time indirect illumination. In *Proc. I3D* 32, 70, 82, 119, 22

KAWASE, M. (2005): Practical implementation of high dynamic range rendering. In *Proc. of Game Developers Conference* 39, 29

KELLER, A. (1997): Instant radiosity. In *Proc. SIGGRAPH*, 49–56 85, 118

KILGARD, M. J. (2001): Chromatic aberration. CEDEC 2001 Talk (NVIDIA Corporation) 35, 25

KLEHM, O., IHRKE, I., SEIDEL, H.-P. AND EISEMANN, E. (2013): Volume stylizer: tomography-based volume painting. In *Proc. I3D* 125

KLEHM, O., IHRKE, I., SEIDEL, H.-P. AND EISEMANN, E. (2014): Property and lighting manipulations for static volume stylization using a painting metaphor. *IEEE Transactions on Visualization and Computer Graphics*, 20, 983–95 125

KNAUS, C. AND ZWICKER, M. (2011): Progressive photon mapping: a probabilistic approach. *ACM Trans. Graph.* 30, 25 28, 94, 100, 17

KNISS, J., PREMOZE, S., HANSEN, C. AND EBERT, D. (2002): Interactive translucent volume rendering and procedural modeling. In *Proc. Visualization*, 109–16 33, 34, 63, 124, 23, 24

KOA, M. D. AND JOHAN, H. (2014): ESLPV: enhanced subsurface light propagation volumes. *Visual Computer*, 30, 821–31 33, 23

KOCH, K., MCLEAN, J., SEGEV, R., FREED, M. A., BERRY, M. J., BALASUBRAMANIAN, V. AND STERLING, P. (2006): How much the eye tells the brain. *Current Biology*, 16, 1428–34 1

KONTKANEN, J., RÄSÄNEN, J. AND KELLER, A. (2004): Irradiance filtering for monte carlo ray tracing. In *Monte Carlo and Quasi-Monte Carlo Methods* 36, 26

KOPF, J., COHEN, M. F., LISCHINSKI, D. AND UYTTENDAELE, M. (2007): Joint bilateral upsampling. *ACM Trans. Graph*, 26 (3), 96 38, 55, 60

KROES, T. (2015): Exposure Render. 28, 18

KRÜGER, J., BÜRGER, K. AND WESTERMANN, R. (2006): Interactive screen-space accurate photon tracing on GPUs. In *Proc. EGSR* 32, 33, 22, 23

KRÜGER, J. AND WESTERMANN, R. (2003): Acceleration techniques for GPU-based volume rendering. In *Proc. IEEE Visualization* 57, 62

KŘIVÁNEK, J., GEORGIEV, I., HACHISUKA, T., VÉVODA, P., ŠIK, M., NOWROUZEZAHRAI, D. AND JAROSZ, W. (2014): Unifying points, beams, and paths in volumetric light transport simulation. *ACM Trans. Graph.* 33, 103 27, 17

LAFORTUNE, E. P. AND WILLEMS, Y. D. (1996): Rendering participating media with bidirectional path tracing. In *Proc. EGWR*, 91–100 26, 16

LAI, G. AND CHRISTENSEN, N. J. (2007): A compression method for spectral photon map rendering. In *Proc. WSCG* 35, 25

LAINE, S., SARANSAARI, H., KONTKANEN, J., LEHTINEN, J. AND AILA, T. (2007): Incremental instant radiosity for real-time indirect illumination. In *Proc. EGSR*, 277–86 28, 18

LAROCHE, C. A. AND PRESCOTT, M. A. (1994): Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients., US Patent 5,373,322 36, 26

LATHROP, K. D. (1968): Ray effects in discrete ordinates equations. *Nuclear Science and Engineering*, 32, 357–69 29, 19

LAURIJSSEN, J., WANG, R., DUTRÉ, P. AND BROWN, B. (2010): Fast estimation and rendering of indirect highlights. *Computer Graphics Forum*, 29, 1305–13 118

LEE, S., KIM, G. J. AND CHOI, S. (2009): Real-Time Depth-of-Field Rendering Using Anisotropically Filtered Mipmap Interpolation. *IEEE Transactions on Visualization and Computer Graphics*, 15, 453–64 39, 42, 46, 47, 50, 51, 29

LEHMANN, E. L. AND CASELLA, G.; CASELLA, G., FIENBERG, S. AND OLKIN, I., EDITORS (2001): Theory of point estimation. 2nd edition. 4

LEVIN, J. R.; MANDL, H. AND LEVIN, J. R., EDITORS (1989): A transfer of appropriate processing perspective of pictures in prose. 1

LI, T.-M., WU, Y.-T. AND CHUANG, Y.-Y. (2012): Sure-based optimization for adaptive sampling and reconstruction. *ACM Trans. Graph.* 31, 194 36

LIKTOR, G. AND DACHSBACHER, C. (2011): Real-time volume caustics with adaptive beam tracing. In *Proc. I3D* 33, 23

LIU, Q. AND WENG, F. (2006): Combined Henyey-Greenstein and Rayleigh phase function. *Applied Optics*, 45, 7475–79 21, 11

LOCHMANN, G., REINERT, B., RITSCHEL, T., MÜLLER, S. AND SEIDEL, H.-P. (2014): Real-time reflective and refractive novel-view synthesis. In *Proc. of International Workshop on Vision, Modeling and Visualization (VMV)*, 9–16 125

LOKOVIC, T. AND VEACH, E. (2000): Deep shadow maps. In *Proc. SIGGRAPH*, 385–92 33, 23

LOPEZ-MORENO, J., CABANES, A. AND GUTIERREZ, D. (2008): Image-based participating media. In *Proc. CEIG* 39, 28

MAX, N. (1994): Efficient light propagation for multiple anisotropic volume scattering. In *Proc. 5th Eurographics Workshop on Rendering* 29, 19

MAX, N. (1995): Optical models for direct volume rendering. *IEEE Trans. Vis. and Computer Graphics*, 1, 99–108 13, 75, 3

MAX, N., SCHUSSMAN, G., MIYAZAKI, R., IWASAKI, K. AND NISHITA, T. (2004): Diffusion and multiple anisotropic scattering for global illumination in clouds. *J. WSCG*, 1–3, 277–84 21, 10

MCGUIRE, M. AND LUEBKE, D. (2009): Hardware-accelerated global illumination by image space photon mapping. In *Proc. HPG* 32, 59, 114, 22

MEHRABIAN, A.; HAYDEN, R. AND EVANS, P., EDITORS (1981): Silent messages: implicit communication of emotions and attitudes. 2nd edition. 1

MEHTA, S. U., WANG, B. AND RAMAMOORTHI, R. (2012): Axis-aligned filtering for interactive sampled soft shadows. *ACM Trans. Graph.* 31, 163 36, 122, 26

MEHTA, S. U., WANG, B., RAMAMOORTHI, R. AND DURAND, F. (2013): Axis-aligned filtering for interactive physically-based diffuse indirect lighting. *ACM Trans. Graph.* 32, 96 36, 122, 26

MEHTA, S. U., YAO, J. X., RAMAMOORTHI, R. AND DURAND, F. (2014): Factored axis-aligned filtering for rendering multiple distribution effects. *ACM Trans. Graph.* 33, 57 36, 122, 26

MIE, G. (1908): Beiträge zur optik trüber medien, speziell kolloidaler metallösungen. *Annalen der Physik*, 330 (3), 377–445 17, 7

MISHRA, S. AND PRASAD, M. (1998): Radiative heat transfer in participating media – a review. *Sadhana*, 23 (2), 213–32 25, 15

MITTRING, M. (2007): Finding next gen: CryEngine 2. In *SIGGRAPH courses* 39, 29

MOON, J. T., WALTER, B. AND MARSCHNER, S. (2008): Efficient multiple scattering in hair using spherical harmonics. *ACM Trans. Graph.* 27, 31 27, 59, 17

MUSBACH, A., MEYER, G. W., REITICH, F. AND OH, S. H. (2014): Full wave modeling of light propagation and reflection. *Comp. Graph. Forum* 32 (6) 70

NALBACH, O., RITSCHEL, T. AND SEIDEL, H.-P. (2014a): Deep screen space. In *Proc. of I3D* 116

NALBACH, O., RITSCHEL, T. AND SEIDEL, H.-P. (2014b): Deep screen space for indirect lighting of volumes. In *Proc. of International Workshop on Vision, Modeling and Visualization (VMV)* 116

NARASIMHAN, S. G., GUPTA, M., DONNER, C., RAMAMOORTHI, R., NAYAR, S. K. AND JENSEN, H. W. (2006): Acquiring scattering properties of participating media by dilution. *ACM Trans. Graph.* 25, 1003–12 19, 29, 32, 83, 89, 8, 22

NARASIMHAN, S. G. AND NAYAR, S. K. (2003): Shedding light on the weather. In *Proc. CVPR* 32, 42, 116, 22, 28

NARASIMHAN, S. G., RAMAMOORTHI, R. AND NAYAR, S. K. (2004): Analytic rendering of multiple scattering in participating media. Columbia University – Technical report 32, 42, 116, 22

NASSAU, K. (2001): The physics and chemistry of color: the fifteen causes of color. 2nd edition. 1

NGUYEN, C. H., NALBACH, O., RITSCHEL, T. AND SEIDEL, H.-P. (2015a): Guiding image manipulations using shape-appearance subspaces from co-alignment of image collections. *Comp. Graph. Forum (Proc. Eurographics)* 34 124

NGUYEN, C. H., RITSCHEL, T. AND SEIDEL, H.-P. (2015b): Data-driven color manifolds. *ACM Trans. Graph.* 34 124

NISHITA, T., DOBASHI, Y. AND NAKAMAE, E. (1996): Display of clouds taking into account multiple anisotropic scattering and sky light. In *Proc. SIGGRAPH*, 379–86 33, 23

NOVÁK, J., NOWROUZEZAHRAI, D., DACHSBACHER, C. AND JAROSZ, W. (2012a): Progressive virtual beam lights. *Comp. Graph. Forum*, 31, 1407–13 28, 122, 17

NOVÁK, J., NOWROUZEZAHRAI, D., DACHSBACHER, C. AND JAROSZ, W. (2012b): Virtual ray lights for rendering scenes with participating media. *ACM Trans. Graph. (Proc. of SIGGRAPH)* 31 27, 28, 17

NOVÁK, J., SELLE, A. AND JAROSZ, W. (2014): Residual ratio tracking for estimating attenuation in participating media. *ACM Trans. Graph.* 33, 179 23, 13

OCTANE (2015): Octane Render., By Otoy Inc. 28, 18

OREN, M. AND NAYAR, S. (1994): Generalization of Lambert's reflectance model. In *Proc. SIGGRAPH* 2

PAPAS, M., REGG, C., JAROSZ, W., BICKEL, B., JACKSON, P., MATUSIK, W., MARSCHNER, S. AND GROSS, M. (2013): Fabricating translucent materials using continuous pigment mixtures. *ACM Trans. Graph. (Proc. of SIGGRAPH)* 32 124

PATTANAIK, S. N. AND MUDUR, S. P. (1993): Computation of global illumination in a participating medium by Monte Carlo simulation. *J. Visualization and Computer Animation*, 4, 133–52 26, 16

PAULY, M., KOLLIG, T. AND KELLER, A. (2000): Metropolis light transport for participating media. In *Proc. EGWR*, 11–22 26, 16

PEGORARO, V., SCHOTT, M. AND SLUSALLEK, P. (2011): A mathematical framework for efficient closed-form single scattering. In *Proc. Graphics Interface* 32, 22

PETSCHNIGG, G., SZELISKI, R., AGRAWALA, M., COHEN, M., HOPPE, H. AND TOYAMA, K. (2004): Digital photography with flash and no-flash image pairs. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 23, 664–72 101

PHARR, M. AND HUMPHREYS, G. (2010): Physically based rendering: from theory to implementation. 2nd edition. 20, 25, 36, 10, 15, 26

PRAUN, E. AND HOPPE, H. (2003): Spherical parametrization and remeshing. *ACM Trans. Graph.* 22, 340–49 84

PREETHAM, A., SHIRLEY, P. AND SMITS, B. (1999): A practical analytic model for daylight. In *Proc. SIGGRAPH*, 91–100 32, 22

PREMOŽE, S., ASHIKHMIN, M., RAMAMOORTHI, R. AND NAYAR, S. K. (2004): Practical rendering of multiple scattering effects in participating media. In *Proc. EGSR* 30, 31, 42, 78, 116, 20, 21, 28

PREMOŽE, S., ASHIKHMIN, M. AND SHIRLEY, P. (2003): Path integration for light transport in volumes. In *Proc. EGRW* 30, 31, 42, 78, 20, 21, 28

PRUTKIN, R., KAPLANYAN, A. AND DACHSBACHER, C. (2012): Reflective shadow map clustering for real-time global illumination. In *Proc. of Eurographics 2012 Short Papers*, 9–12 88, 118

PURCELL, T., DONNER, C., CAMMARANO, M., JENSEN, H. AND HANRAHAN, P. (2003): Photon mapping on programmable graphics hardware. In *Proc. Graphics Hardware*, 41–50 32, 22

RAAB, M., SEIBERT, D. AND KELLER, A. (2006): Unbiased global illumination with participating media. In *Proc. Monte Carlo and Quasi-Monte Carlo Methods*, 591–606 23, 26, 12, 16

REN, Z., ZHOU, K., LI, T., HUA, W. AND GUO, B. (2010): Interactive hair rendering under environment lighting. *ACM Trans. Graph.* 29, 55 21, 11

REN, Z., ZHOU, K., LIN, S. AND GUO, B. (2008): Gradient-based interpolation and sampling for real-time rendering of inhomogeneous, single-scattering media. Microsoft Research (MSR-TR-2008-51). – Technical report 33, 23

RILEY, K., EBERT, D. S., KRAUS, M., TESSENDORF, J. AND HANSEN, C. D. (2004): Efficient rendering of atmospheric phenomena. In *Proc. EGSR* 33, 124, 23

RITCHIE, M., MODERN, G. AND MITCHELL, K. (2010): Split second motion blur. In *SIGGRAPH Talks* 39, 29

RITSCHEL, T., GROSCH, T. AND SEIDEL, H.-P. (2009): Approximating dynamic global illumination in image space. In *Proc. I3D* 35, 39, 120, 25, 29

ROKITA, P. (1993): Fast generation of depth of field effects in computer graphics. *Computers & Graphics*, 17, 593–95 39, 29

ROUSSELLE, F., KNAUS, C. AND ZWICKER, M. (2011): Adaptive sampling and reconstruction using greedy error minimization. *ACM Trans. Graph.* 30, 159 36

ROUSSELLE, F., KNAUS, C. AND ZWICKER, M. (2012): Adaptive rendering with non-local means filtering. *ACM Trans. Graph.* 31, 195 38

RUSHMEIER, H. E. AND TORRANCE, K. E. (1987): The zonal method for calculating light intensities in the presence of a participating medium. *Computer Graphics (Proc. SIGGRAPH)*, 21, 293–302 29, 18

SALVI, M., VIDIMČE, K., LAURITZEN, A. AND LEFOHN, A. (2010): Adaptive volumetric shadow maps. In *Proceedings of the 21st Eurographics conference on Rendering* 33, 23

SCHJØTH, L., FRISVAD, J. R., ERLEBEN, K. AND SPORRING, J. (2007): Photon differentials. In *Proc. GRAPHITE* 36, 94, 121, 26

SCHWENK, K. (2013): Filtering techniques for low-noise previews of interactive stochastic ray tracing. Ph. D thesis, Technical University Darmstadt 38

SEN, P. AND DARABI, S. (2012): On filtering the noise from the random parameters in Monte Carlo rendering. *ACM Trans. Graph.* 31, 18 38

SHINYA, M., TAKAHASHI, T. AND NAITO, S. (1987): Principles and applications of pencil tracing. *SIGGRAPH Comput. Graph.* 21, 45–54 122

SIGG, C. AND HADWIGER, M.; PHARR, M. AND FERNANDO, R., EDITORS (2005): CHAP. 20. INGPU Gems 2. 49

SILLION, F. X. (1994): Clustering and volume scattering for hierarchical radiosity calculations. In *Proc. of Eurographics Workshop on Rendering* 29, 119, 19

SILLION, F. X. (1995): A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Transactions on Visualization and Computer Graphics*, 1, 240–54 29, 119, 19

SLOAN, P., KAUTZ, J. AND SNYDER, J. (2002): Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *ACM Trans. Graph.* 33, 23

STAM, J. (1995): Multiple scattering as a diffusion process. In *Proc. EGWR*, 41–50 27, 29, 119, 17, 19

STAM, J. (1999): Diffraction shaders. In *Proc. of SIGGRAPH*, 101–10 121

STRUTT, J. W. (1871): On the light from the sky, its polarization and colour. *Philosophical Magazine*, 41, 107–20 17, 6

SUN, B., RAMAMOORTHI, R., NARASIMHAN, S. G. AND NAYAR, S. K. (2005): A practical analytic single scattering model for real time rendering. *ACM Trans. Graph.* 24, 1040–49 32, 22

SUNDSTEDT, V., GUTIERREZ, D., ANSON, O., BANTERLE, F. AND CHALMERS, A. (2007): Perceptual rendering of participating media. *ACM Trans. Appl. Percept.* 4, 15 124

SUYKENS, F. AND WILLEMS, Y. D. (2001): Path differentials and applications. In *Proc. EGWR*, 257–68 36, 94, 121, 122, 26

SZIRMAY-KALOS, L., SBERT, M. AND UMENHOFFER, T. (2005): Real-time multiple scattering in participating media with illumination networks. In *Proc. EGSR* 33, 23

SZIRMAY-KALOS, L., TÓTH, B. AND MAGDICS, M. (2011): Free path sampling in high resolution inhomogeneous participating media. *Computer Graphics Forum*, 30, 85–97 23, 13

TESSENDORF, J. (1987): Radiative transfer as a sum over paths. *Phys. Rev. A*, 35, 872–78 30, 31, 20, 21

TESSENDORF, J. (2011): Angular smoothing and spatial diffusion from the Feynman path integral representation of radiative transfer. *J. Quantitative Spectroscopy & Radiative Transfer*, 112, 751–60 30, 31, 20, 21

THOMAS, S. W. (1986): Dispersive refraction in ray tracing. *The Visual Computer*, 2, 3–8 35, 25

TOMASI, C. AND MANDUCHI, R. (1998): Bilateral filtering for gray and color images. In *Proc. of ICCV* 38

TOUBLANC, D. (1996): Henyey-Greenstein and Mie phase functions in Monte Carlo radiative transfer computations. *Appl. Opt.* 35, 3270–74 21, 11

VEACH, E. (1997): Robust Monte Carlo methods for light transport simulation. Ph. D thesis, Stanford University 6, 30, 36, 20, 26

VEACH, E. AND GUIBAS, L. J. (1994): Bidirectional estimators for light transport. In *Proc. of Eurographics Rendering Workshop* 26, 16

VEACH, E. AND GUIBAS, L. J. (1995): Optimally combining sampling techniques for Monte Carlo rendering. In *Proc. SIGGRAPH* 26, 16

VEACH, E. AND GUIBAS, L. J. (1997): Metropolis light transport. In *Proc. SIGGRAPH* 26, 16

VORBA, J., KARLÍK, O., ŠIK, M., RITSCHEL, T. AND KŘIVÁNEK, J. (2014): On-line learning of parametric mixture models for light transport simulation. *ACM Trans. Graph. (Proc. of SIGGRAPH)* 33 118

WANG, N. (2003): Realistic and fast cloud rendering. In *J. Graphics, GPU & Game Tools* 33, 23

WANG, Y., WANG, J., HOLZSCHUCH, N., SUBR, K., YONG, J.-H. AND GUO, B. (2010): Real-time rendering of heterogeneous translucent objects with arbitrary shapes. *Comp. Graph. Forum*, 29, 497–506 33, 23

WENZEL, C. (2006): Real-time atmospheric effects in games. In *SIGGRAPH Course*, 113–128 33, 23

WHITTED, T. (1980): An improved illumination model for shaded display. *Communications of ACM*, 23, 343–49 2

WILKIE, A., NAWAZ, S., DROSKE, M., WEIDLICH, A. AND HANIKA, J. (2014): Hero wavelength spectral sampling. *Computer Graphics Forum (Proc. EGSR)* 33 35, 25

WOODCOCK, E., MURPHY, T., HEMMINGS, P. AND LONGWORTH, T. (1965): Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry. In *Proc. Application of Computing Methods to Reactor Problems*, 557–579 23, 12

WYMAN, C. AND RAMSEY, S. (2008): Interactive volumetric shadows in participating media with single-scattering. In *Proc. IEEE Interactive Ray Tracing*, 87–92 33, 35, 23, 25

WYMAN, C. AND DAVIS, S. (2006): Interactive image-space techniques for approximating caustics. In *Proc. I3D* 114

WYMAN, D. R., PATTERSON, M. S. AND WILSON, B. C. (1989): Similarity relations for the interaction parameters in radiation transport. *Applied Optics*, 28 (24), 5243–49 27, 54, 58, 84, 120, 17

YAN, L.-Q., UDAY MEHTA, S., RAMAMOORTHI, R. AND DURAND, F. (2014): Fast 4D sheared filtering for interactive rendering of distribution effects. EECS Department, University of California, Berkeley (UCB/EECS-2014-174). – Technical report 36, 122, 26

YUE, Y., IWASAKI, K., CHEN, B.-Y., DOBASHI, Y. AND NISHITA, T. (2010): Unbiased, adaptive stochastic sampling for rendering inhomogeneous participating media. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 29, 177:1–177:8 23, 13

ZHAO, S., JAKOB, W., MARSCHNER, S. AND BALA, K. (2011): Building volumetric appearance models of fabric using micro CT imaging. *ACM Trans. Graph.* 30, 44 21, 11

ZHAO, S., RAMAMOORTHI, R. AND BALA, K. (2014): High-order similarity relations in radiative transfer. *ACM Trans. Graph.* 33, 104 21, 27, 84, 11, 17

ZHOU, K., HOU, Q., WANG, R. AND GUO, B. (2008a): Real-time kD-tree construction on graphics hardware. In *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* Volume 27,, 126 32, 22

ZHOU, K., REN, Z., LIN, S., BAO, H., GUO, B. AND SHUM, H.-Y. (2008b): Real-time smoke rendering using compensated ray marching. In *ACM Trans. Graph.*, 36 29, 33, 19, 23

ZINKE, A. AND WEBER, A. (2007): Light scattering from filaments. *IEEE Transactions on Visualization and Computer Graphics*, 13, 342–56 21, 11

ZINKE, A., YUKSEL, C., WEBER, A. AND KEYSER, J. (2008): Dual scattering approximation for fast multiple scattering in hair. *ACM Trans. Graph.* 27, 32 21, 11

ZOLEK, N. S., WOJTKIEWICZ, S. AND LIEBERT, A. (2008): Correction of anisotropy coefficient in original Henyey Greenstein phase function for Monte Carlo simulations of light transport in tissue. *J. Biocybernetics And Biomedical Engineering*, 28, 59–73 21, 11