

Light Transport in Participating Media



UNIVERSIDAD DE
ZARAGOZA



CENTRO POLITÉCNICO
SUPERIOR



DEPARTAMENTO DE
INFORMÁTICA E INGENIERÍA
DE SISTEMAS



GRUPO DE INFORMÁTICA GRÁFICA
AVANZADA (ADVANCED COMPUTER
GRAPHICS GROUP)

Adolfo Muñoz Orbañanos

Supervised by:

Diego Gutierrez Perez

Francisco Jose Seron Arbeloa

Departamento de Informática e Ingeniería de Sistemas

Universidad de Zaragoza

I would like to dedicate this thesis to all the people who has suffered most from it:

- To my supervisors, who have suffered several dozens of crazy ideas, who have received (and read) tons of e-mails regarding this research (some of them sent later than midnight), and who have carefully read this document several times.
- To these people who will come to the final presentation of this PhD with the only aim of supporting me, knowing that it might be the most boring hour of their lives. I promise I will do my best to keep it as amusing and interesting as possible for you too.
- All my friends who have dared to ask "What is your PhD about?" and have even waited for the full answer.
- My parents and brother, who dinner after dinner have suffered my urgent needs of suddenly writing equations on a paper napkin, my absent-minded moments and my endless talks. Never underestimate the inspirational power an uninformed suggestion, if the person that gives it is really motivated to help you.
- To Silvia, who is most probably the one who has suffered the most of this PhD. She has been forgiving when all these "I just need half an hour more" have actually become three hours more. She has been patient with all my babbling and talking related to research, deadlines, physics of light, rainbows, translucency... And most important, she has been there for me even when I was not there for her, either physically (when my body was on the lab for hours) or psychologically (when we were together but my mind was still on the research).

Acknowledgements

All this research would have never ever been possible without the cooperation of the following people:

- The supervisors: Diego Gutierrez and Francisco Jose Seron.
- The co-authors of past published papers: Oscar Anson, Jorge Lopez-Moreno, Belen Masia, Alfonso Tolosa, Jorge Jimenez, Francisco Jose Seron and Diego Gutierrez.
- The co-authors of (hopefully) future published papers: Iman Sadeghi, Philip Laven, Wojciech Jarosz, Henrik Wann Jensen, Jose Ignacio Echevarria, Jorge Lopez-Moreno, Mashhuda Glencross, Francisco Jose Seron and Diego Gutierrez.
- The people of the Advanced Computer Graphics Group (GIGA) of the University of Zaragoza.
- All the people who wrote those tons of papers that inspired me for this research.

This research was partly done under the sponsorship of:

- The Spanish Ministry of Education and Research through the project TIN2004-07672-C03-03TIN2004.
- The Spanish Ministry of Science and Technology through the project TIN2007-63025.
- Diputación General de Aragón (CONAID).
- Caja de Ahorros de la Inmaculada (bank) through the project IT21/08.

Contents

1	Introduction	1
1.1	Light scattering	1
1.2	Participating media	2
1.3	Goals	3
1.4	Measurable results of this PhD	3
1.4.1	Publications	3
1.4.2	Projects	4
1.4.3	Research stays	5
1.5	Distribution of this document	5
	References	7
2	Transport of Electromagnetic Radiation	9
2.1	Maxwell's Equations	9
2.1.1	Wave-like properties of electromagnetic radiation	10
2.1.1.1	Polarization	11
2.1.1.2	Interference	12
2.1.1.3	Other wave phenomena	14
2.1.2	Particle-like properties of electromagnetic radiation	14
2.2	Index of refraction and wave propagation	14
2.2.1	Eikonal equation and Fermat's principle	15
2.2.2	Non-linear media	15
2.2.3	Dielectric vs. dielectric interface	16
2.3	Absorption	18
2.3.1	Absorbing media	19
2.3.2	Dielectric vs. metal interface	19

CONTENTS

2.3.3	Inelastic scattering	20
2.4	Light scattering	20
2.4.1	Participating media	21
2.4.2	Surfaces: the boundary between two media	24
2.4.3	Wavelength redistribution	27
2.5	Summary	29
	References	32
3	Physically - based modeling	33
3.1	Basics and previous work	33
3.1.1	Index of refraction	34
3.1.2	BRDFs	34
3.1.3	Participating media	36
3.1.3.1	Rainbows	37
3.1.4	BSSRDFs	38
3.2	Inverse rendering using genetic algorithms	40
3.2.1	Brief introduction about genetic algorithms	40
3.2.2	Appearance acquisition	41
3.2.2.1	Variables and objective function	41
3.2.2.2	Implementation of the algorithm	42
3.2.2.3	Parameter space	43
3.3	Estimating BSSRDFs from single HDR photographs	45
3.3.1	Algorithm	45
3.4	Generalizing to Single Images	49
3.5	The ocean: building a model from measured data and standards	52
3.5.1	The Bio-Optical Model	53
3.5.2	Modeling Absorption	54
3.5.3	Modeling Elastic Scattering	55
3.5.4	Modeling Inelastic Scattering	56
3.5.4.1	Fluorescence	56
3.5.4.2	Raman scattering	58
3.6	Summary	58
	References	64

4	Simulation	65
4.1	Basics and previous work	65
4.1.1	Monte Carlo integration	65
4.1.2	Ray tracing	66
4.1.3	Photon mapping	66
4.1.4	Participating media	67
4.1.5	Inelastic scattering	69
4.1.6	Non-linear media	69
4.1.7	Wave phenomena	70
4.2	Simulation of inelastic scattering on participating media	70
4.3	Optimized rendering of non-linear media	74
4.3.1	General considerations for optimization techniques	74
4.3.1.1	Factors of computation time	75
4.3.2	Optimization techniques for curved ray tracing	76
4.3.2.1	Parallelization	76
4.3.2.2	Symmetries	76
4.3.2.3	Undersampling	77
4.3.2.4	Path caching	77
4.3.3	Smart step selection	78
4.3.4	Optimization techniques for curved particle tracing	78
4.3.4.1	Parallelization	78
4.3.4.2	Symmetry	79
4.3.4.3	Undersampling	79
4.3.4.4	Path caching	80
4.3.4.5	Smart step selection	80
4.3.5	Numerical methods	81
4.4	Polarization and interference: The rainbow	81
4.4.1	Optics of water drops	81
4.4.1.1	Spherical water drops	81
4.4.1.2	Non-spherical water drops	83
4.4.2	Light transport	84
4.4.2.1	Ray-particle interactions	86
4.4.2.2	Optical path	87

CONTENTS

4.4.2.3	Focal lines	88
4.4.3	Ray casting	89
4.4.3.1	Casting rays	90
4.4.3.2	Collecting sphere	93
4.4.3.3	Interpolation and interference	94
4.5	Summary	96
	References	100
5	Results	101
5.1	Material acquisition from inverse rendering	101
5.2	BSSRDF approximation from single HDR photographs	103
5.3	Underwater imagery	107
5.4	Efficiency of non-linear media simulation	109
5.4.1	Study of numerical methods	109
5.4.2	Efficiency due to optimization techniques	111
5.4.2.1	Test scenes	111
5.4.2.2	Rendering times study	113
5.5	Rainbow rendering using simulated phase functions	115
5.5.1	Validation	115
5.5.2	Result	117
	References	122
6	Conclusions	123
6.1	Future work	124

Chapter 1

Introduction

1.1 Light scattering

Maxwell's equations ([Max65](#)) completely define the behavior of electromagnetic radiation, according to the electric and magnetic fields. Visible light is electromagnetic radiation, and therefore its behavior also follows the same laws. For a perfect physically-based simulation of the illumination of a scene, the ideal algorithm would solve these equations for the whole input data of the scene. However this would be unpractical even for the simplest scene due to:

- Very high and unreasonable computation times
- The need to model the scene in terms of electric and magnetic fields.

In Computer Graphics, even in the field of physically-based rendering, simplified and more intuitive models are used. These simplifications enable a more practical simulation, with understandable and intuitive modeling based on light sources, geometries, material parameters and participating media, plus they enable algorithms that simulate the illumination on those scenes in an adequate trade-off between accuracy and rendering time.

Still, the potential results of a simplified system will always be a subset of all the potential optic phenomena that are owed to Maxwell equations. By simplifying the simulation, even though rendering time decreases and scene modeling becomes more intuitive, limitations on the possible simulated scenes arise.

However, these limitations can be overcome. More complex scene models that are able to represent scenes that include new phenomena without losing intuitiveness can be designed. New algorithms that are able to simulate a broader spectra of phenomena can be developed. Actually, there is plenty of previous work that tries to focus into these phenomena that are related to the nature of light, considering complex physically based models and creating new algorithms to simulate these new scenes. But, where is the limit? How close can we get to the complete simulation of light scattering without getting to unreasonable rendering times? How complex can a physically-based model become without losing intuitiveness and practicality? How close can we get to Maxwell's equations?

These are the questions that this PhD is trying to answer by trying to get to these limits, going further into the physical simulation of light scattering in scenes while keeping the modeling of scenes as intuitive and simple as possible.

1. INTRODUCTION

1.2 Participating media

Participating media have always been an active research field in computer graphics. Its extra dimensionality (as opposed to surfaces) turns the modeling and simulation of participating media a very exciting, full of potential field. At every differential point along the optical path traversing a participating medium, electromagnetic radiation is altered in some way. The key equation of radiative transfer through participating media is called the Radiative Transfer Equation ([Cha60](#)):

$$\frac{dL(\vec{\omega}_o)}{dl} = L_e(\vec{\omega}_o) - (\sigma_a + \sigma_s)L(\vec{\omega}_o) + \sigma_s \int_{\Omega} p(\vec{\omega}_i, \vec{\omega}_o) L_i(\vec{\omega}_i) d\vec{\omega}_i \quad (1.1)$$

which basically indicates that radiance L gets affected by the participating media at every differential point along the optical path l . Infinitesimally, there can be light emission (addend $L_e(\vec{\omega}_o)$), extinction due to absorption or out-scattering (addend $-(\sigma_a + \sigma_s)L(\vec{\omega}_o)$) or radiance increase coming from in-scattering from other light paths (addend $\sigma_s \int_{\Omega} p(\vec{\omega}_i, \vec{\omega}_o) L_i(\vec{\omega}_i) d\vec{\omega}_i$). A more detailed explanation on this can be found on section [2.4.1](#). This equation is defined for each differential wavelength (spectral dependence) and can present different coefficients at every differential point (non-homogeneous media).

However, this previous equation is incomplete. First of all, it is defined per differential wavelength but does not consider the possibility of wavelength transfer, which happens with some phenomena such as fluorescence. Also, it gives a definition of what happens along the optical path, but, which optical path is being traversed? The obvious answer would be a straight line along $\vec{\omega}_o$, but the path could be curved by a non-homogeneous index of refraction along the medium (which would cause some phenomena such as mirages). A participating medium can be participating either because it affects the radiance of light that traverses it or because it affects the path of light itself.

So, even though the RTE is able to model a participating media, there are some phenomena, such as inelastic scattering (fluorescence) or non-linear media (in which the path of light gets curved) that cannot be considered by only that equation. Although both phenomena have already been considered by some computer graphics work, this PhD expects to provide methods that are more general while being as efficient as possible both for non-linear media and for inelastic scattering.

On the other hand, the RTE depends on several coefficients (σ_a and σ_s) and the phase function $p(\vec{\omega}_i, \vec{\omega}_o)$. Those coefficients and function define the medium itself, and they need to be specified for any medium. But... which coefficients model an specific medium? Where do we get them from? What phenomena are being modeled by a specific definition of the phase function? This PhD deals with this problem for some specific media, trying to cover as many different modeling approaches as possible:

- **Ocean water.** Ocean water is a very interesting participating medium, because of its multiple constituents and suspended particles. Furthermore, it presents a wide range of inelastic scattering phenomena. The models for the corresponding coefficients and phase functions are gathered from literature about the topic, from standard measured data and from several studies, effectively composing a full bio-optical model.
- **Rainbows.** The phase function that models water droplets suspended in air is what gives rainbows the colored arcs that we are used to. Instead of measuring these data, this PhD simulates the electromagnetic phenomena that happen inside a water drop in order to get the corresponding phase function.
- **Translucent materials from pictures.** A translucent object's material is actually a participating medium. While measuring the translucency properties of these materials requires of specialized equipment or is a very underconstrained problem, part of the research of this PhD deals with the approximation of a translucent material from a single picture.

1.3 Goals

All the specific goals of this PhD are related to the exploration of the potential of the physics of light scattering for simulating new phenomena in participating media, and trying to keep the corresponding physically-based models as simple and intuitive as possible and rendering times as short as possible. These goals are:

- Simplify the modeling of materials by exploring the potential of estimating their physical properties from single photographs, focusing on complex materials such as translucent ones.
- Include the concept of inelastic scattering phenomena (like fluorescence) into participating media simulation. Try to keep the modeling of the corresponding participating media as simple as possible.
- Simulate non-linear media (media in which the index of refraction is not constant but varies at each differential point) in the most efficient way without losing optical accuracy.
- Study and simulate the phase function of rainbows, as an example of the combination of several complex optic phenomena such as polarization and wave interference.

1.4 Measurable results of this PhD

1.4.1 Publications

Part of the work on this PhD has already been published:

- The genetic algorithms based technique that is able to capture material properties from inverse rendering (presented in section 3.2 with results in section 5.1) has already been published in two international conferences ([MMT⁺09](#); [MMTG09](#)).
- The bio-optical model of ocean water (see section 3.5) along with the algorithm that is able to render inelastic scattering phenomena within participating media (presented in section 4.2) with the corresponding results (section 5.3) have been published in Eurographics ([GSMA08](#)). This conference's proceedings are included in the journal *Computer Graphics Forum*, which in 2008 had an impact index of 1.860 (2.059 is the average of the last 5 years), which is the 21st out of 86 of the subject category *Computer Science, Software Engineering* of the JCR list ([JCR](#)).
- The efficiency techniques for non-linear media simulations (presented in section 4.3 with results in section 5.4) have appeared in two different publications in different development stages ([MGS06](#); [MGS07](#)). The second of this publications is the journal named *Visual Computer*, had in 2007 an impact factor of 0.690, being the 54th out of 84 of the JCR list ([JCR](#)).

Also, two extra publications are planned:

1. INTRODUCTION

- The BSSRDF approximation algorithm that is able to estimate BSSRDFs from single HDR photographs without prior knowledge (presented in section 3.3 with results in section 5.2) is scheduled to be submitted for SIGGRAPH 2010 (with proceedings appearing on the journal ACM Transactions on Graphics. This work has been done in cooperation with Jose Ignacio Echevarria, Francisco J. Seron, Jorge Lopez and Diego Gutierrez from Universidad de Zaragoza, and with Mashhuda Glencross, from the University of Manchester
- The algorithm that is able to simulate phase functions from ray tracing with polarization and interference information (presented in section 4.4 with results in section 5.5) is scheduled to be submitted to the ACM Transactions on Graphics during 2010. This work has been done in cooperation with Philip Laven, from the European Broadcasting Union (Geneva), Wojciech Jarosz, from Disney Research (Zürich) and Iman Sadeghi and Henrik Wann Jensen from the University of California San Diego.

The ACM Transactions on Graphics journal had an impact factor in 2008 of 3.383 (being the 3rd out of 86) of the JCR list (**JCR**), with a 5 years average impact factor of 4.997 (the 1st out of 86).

1.4.2 Projects

The process of this PhD has included the collaboration on the following projects:

- **Iluminaciones Virtuales en Instalaciones a Tamaño Real. (OTRI 2003/0258)** from April to December 2003. Development of a light simulation system and a CAD tool for luminary design and distribution. Main researcher: Francisco J. Seron. Funded by the company Indalux Iluminación Técnica, S. L.
- **MUMA: Development of Multi-Disciplinary Management Strategies for Conservation and Use of Heritage Sites in Asia and Europe. (ASI/B7-301/98/679/051)** from September 2004 to December 2006 (funded by the Asia-Link program of the European Commission). During this period a collaborative web platform was developed and several lectures and seminars were given in different universities of Germany, China and India. Main researcher: Diego Gutierrez.
- **Sistema Informático para el Ajuste de Colores de Esmaltes Vítreos. (OTRI 2006/0056)** from September 2005 to December 2006 (as main researcher). Development of a predictive system for vitreous enamels on metal. Main researcher: Adolfo Muñoz. Funded by the company Vitrex, S. L.
- **Interfaces gráficas y algoritmos para aplicaciones basadas en la simulación de luz artificial y natural (TIN2004-07672-C03-03)** from May 2006 to November 2007. Funded by the Spanish Ministry of Education and Research. Main researcher: Francisco J. Seron
- **Fotografía Computacional (UZ2007-TEC-06)** from January to December 2008. Project about Computational Photography. Funded by the Universidad de Zaragoza. Main researcher: Francisco J. Seron
- **TANGIBLE: Humanos Virtuales Realistas e Interacción Natural y Tangible. (TIN2007-63025)** from October 2007 until the present day. Funded by the Spanish Ministry of Science and Technology. Main researcher: Francisco J. Seron

1.4.3 Research stays

During the period of this PhD two research stays have been done:

- Two weeks at University of California San Diego with Henrik Wann Jensen. This period originated the research related to the rainbow (see sections 4.4 and 5.5).
- Two months at University of Bristol with Erik Reinhard. This period helped with the research about shape from shading and depth estimation techniques, which were applied to the BSSRDF project (see section 3.3).

1.5 Distribution of this document

This PhD is organized as follows:

- Chapter 2 introduces the concepts of the physics of light scattering, as base for the remaining chapters.
- Chapter 3 focuses on the modeling of scenes with extended optical behavior. It focuses first on modeling ocean water as an example of a participating media with inelastic scattering, and next on the modeling of complex materials (including translucent ones) by taking the corresponding properties from single pictures.
- Chapter 4 introduces the simulation of scenes that are complex from the physical point of view. It includes a physically-based simulation of the interactions on a water drop in order to get the phase function of a rainbow (considering polarization and interference), defines the algorithm that is able to model participating media with inelastic scattering and explores efficient techniques for simulating non-linear media.
- Chapter 5 presents the results of this work.
- Chapter 6 finalizes this dissertation by summarizing the conclusions.

1. INTRODUCTION

References

- [Cha60] S. Chandrasekhar, *Radiative transfer*, Dover Publications, Inc., 1960. 2
- [GSMA08] Diego Gutierrez, Francisco J. Serón, Adolfo Muñoz, and Oscar Anson, *Visualizing underwater ocean optics*, Computer Graphics Forum **27** (2008), no. 2, 547–556.
- [JCR] <http://www.accesowok.fecyt.es/jcr/>. 3, 4
- [Max65] James Cler Maxwell, *A dynamical theory of the electromagnetic field*, Philosophical Transactions of the Royal Society of London (1865), no. 155, 459–512. 1
- [MGS06] Adolfo Muñoz, Diego Gutierrez, and Francisco J. Serón, *Efficient physically-based simulation of non-linear media*, GRAPHITE (Y. T. Lee, Siti Mariyam Hj. Shamsuddin, Diego Gutierrez, and Norhaida Mohd Suaib, eds.), ACM, 2006, pp. 97–105. 3
- [MGS07] Adolfo Munoz, Diego Gutierrez, and Francisco J. Seron, *Optimization techniques for curved path computing*, Vis. Comput. **23** (2007), no. 7, 493–502. 3
- [MMT⁺09] Belen Masia, Adolfo Munoz, Alfonso Tolosa, Oscar Anson, Jorge Lopez-Moreno, Jorge Jimenez, and Diego Gutierrez, *Genetic algorithms for estimation of reflectance parameters*, Poster Proceedings of Spring Conference on Computer Graphics, 2009, pp. 39–44. 3
- [MMTG09] Adolfo Munoz, Belen Masia, Alfonso Tolosa, and Diego Gutierrez, *Single-image appearance acquisition using genetic algorithms*, Proceedings of Computer Graphics, Visualization, Computer Vision and Image Processing, 2009, pp. 24–32. 3

REFERENCES

Chapter 2

Transport of Electromagnetic Radiation

2.1 Maxwell's Equations

The principles of all electromagnetic phenomena are defined (in a very concise and elegant way) by Maxwell's equations (Max65). The work by James Clerk Maxwell provides, among other research, a set of twenty equations that set the fundamentals of modern dynamic electromagnetism. For the sake of simplicity, the following equations are included. They are on differential form and assume absence of magnetic or polarizable media:

$$\nabla \cdot \vec{E} = \frac{\rho}{\epsilon_0} = 4\pi k\rho \quad (2.1)$$

$$\nabla \cdot \vec{B} = 0 \quad (2.2)$$

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad (2.3)$$

$$\nabla \times \vec{B} = -\frac{4\pi k}{c^2} \vec{J} + \frac{1}{c^2} \frac{\partial \vec{E}}{\partial t} = \mu_0 \vec{J} + \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t} \quad (2.4)$$

where \vec{E} is the electric field vector, \vec{B} is the magnetic field vector, ρ is the total charge density, ϵ_0 is the electrical permittivity of free space, $k = (4\pi\epsilon_0)^{-1}$ is Coulomb's constant, t represents time, \vec{J} is the total current density vector, μ_0 is the magnetic permeability of free space and c is the speed of light on free space. The operators are: $\nabla \cdot$ is the divergence operator, $\nabla \times$ is the curl operator and ∂ represents the partial derivative.

For more details or other forms under different assumptions Maxwell's work (Max65) is a must read. Furthermore, given the fact that light is an electromagnetic wave and therefore its behavior is also governed by the same equations, Maxwell's work has become one of the fundamentals of modern

2. TRANSPORT OF ELECTROMAGNETIC RADIATION

optics (BW02). Actually, the whole physics of light can be derived from it, as it can be seen in next subsections. All the deductions are not included here for the sake of brevity but they can be found on several optics books (BW02; LLT95).

One of the most relevant facts about electromagnetic radiation (and more specifically, light) is the *wave-particle duality*. Depending on the concepts and deductions done from Maxwell's equations, electromagnetic radiation exhibits both particle-like and wave-like properties.

2.1.1 Wave-like properties of electromagnetic radiation

The microscopical behavior of electromagnetic radiation (such as visible light) can be deduced from Maxwell's equations. The equation corresponding to an electromagnetic wave can be inferred by considering either the electric field (\vec{E}) or the magnetic field (\vec{B}):

$$\begin{aligned} \left(\nabla - \frac{1}{c^2} \partial^2 \partial t^2 \right) \vec{E} &= 0 \\ \left(\nabla - \frac{1}{c^2} \partial^2 \partial t^2 \right) \vec{B} &= 0 \end{aligned} \quad (2.5)$$

where c is the speed of light and t represents time. As the deduction is equivalent both for \vec{E} and \vec{B} , the rest of this section is focused on the electric field. The corresponding derivations for the magnetic field are equivalent.

The general solution for equation 2.5 is a lineal superposition of waves in the form:

$$\vec{E}(\vec{r}, t) = g(\omega t - \vec{w} \cdot \vec{r}) \quad (2.6)$$

where ω is the angular frequency and \vec{w} is the wave vector, which must satisfy $|\vec{w}| = \frac{\omega}{c}$. g can be any function, although usually monochromatic sine waves are used. Being λ the wavelength of the wave, we can also define the equality $|\vec{w}| = \frac{2\pi}{\lambda}$.

Considering function g in equation 2.6 as a monochromatic sinusoid for a plane electromagnetic wave in which the propagation direction is along the z axis we get that an electromagnetic wave is a linear superposition of the following:

$$\vec{E}(\vec{r}, t) = \begin{pmatrix} A_x \cos\left(\frac{2\pi}{\lambda} z - \omega t + \delta_x\right) \\ A_y \cos\left(\frac{2\pi}{\lambda} z - \omega t + \delta_y\right) \\ 0 \end{pmatrix} \quad (2.7)$$

where A_x and A_y are the amplitude components of the wave at the x and y axis, respectively and δ is the corresponding phase. Notice that the full wave is the combination of the previous equation for each different wavelength λ and phase δ . The corresponding irradiance E is¹:

¹In optic books, this magnitude is often called *intensity* and noted as I . However, on radiometry (and on physically-based rendering), intensity is used to name the radiant power per unit solid angle (see section 2.4). The notation E for irradiance is widely used and should not be confused with \vec{E} , standard notation for the electric field

$$E = \frac{c^2 \epsilon_0}{2v} (|A_x|^2 + |A_y|^2) \quad (2.8)$$

where v is the speed of light in the medium being traversed.

It can be concluded that electromagnetic radiation consists on waves that are perpendicular to their corresponding propagation direction (in the case of plane waves), as in figure 2.1. The oscillation of the wave occurs along the plane perpendicular to the propagation path, according to the electric and the magnetic field. This nature of light leads to numerous phenomena, like polarization of light, the interference between waves or diffraction.

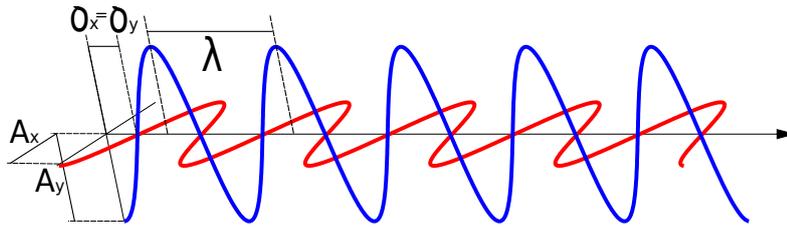


Figure 2.1: Electromagnetic wave

2.1.1.1 Polarization

Depending on the nature of the light sources, due to interactions with materials with some specific properties (see section 2.2.3) or due to light traversing some specific filters, a correlation on the orientation of the waves might appear. When this correlation does not exist at all, it is said that the light is *unpolarized*. When the correlation is absolute the light is *fully polarized*. In general circumstances, light is partially polarized and can be expressed as a linear combination of unpolarized fully polarized light.

In the case of a planar wave, a fully polarized ray of an electromagnetic wave for a specific wavelength λ can be expressed as follows:

$$\vec{E}(\vec{r}, t) = \begin{pmatrix} A_x \cos\left(\frac{2\pi}{\lambda}z - \omega t + \delta_x\right) \\ A_y \cos\left(\frac{2\pi}{\lambda}z - \omega t + \delta_y\right) \\ 0 \end{pmatrix} \quad (2.9)$$

Notice that, while on previous section the full wave was defined as a linear superposition of monochromatic waves, a fully polarized ray is not anymore a complete linear superposition, but a wave that is just a simple combination of the two components on the axis perpendicular to the propagation direction. Depending on the relationship of the two phases of the two components there are several kinds of polarizations that can be identified, as it can be seen on figure 2.2. The corresponding name of each of the identified types comes from the projection of the wave on the plane perpendicular to the propagation direction. The general case of a fully polarized electromagnetic wave is an ellipse, which is often called *polarization ellipse*.

2. TRANSPORT OF ELECTROMAGNETIC RADIATION

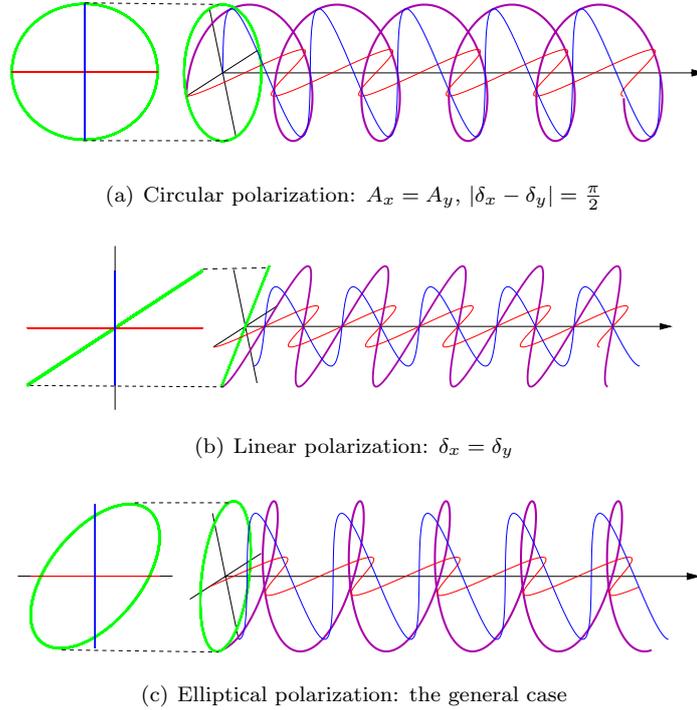


Figure 2.2: Different kinds of polarization, according to the relative amplitudes and phases of each of the components.

2.1.1.2 Interference

Another (and most probably the most widely known) phenomena that is related to waves is called *interference*. Interference happens when two waves are superimposed in space. If decomposed into differential monochromatic waves, the addition of two waves with the same wavelength λ results into another monochromatic wave with λ wavelength and with new amplitude and phase.

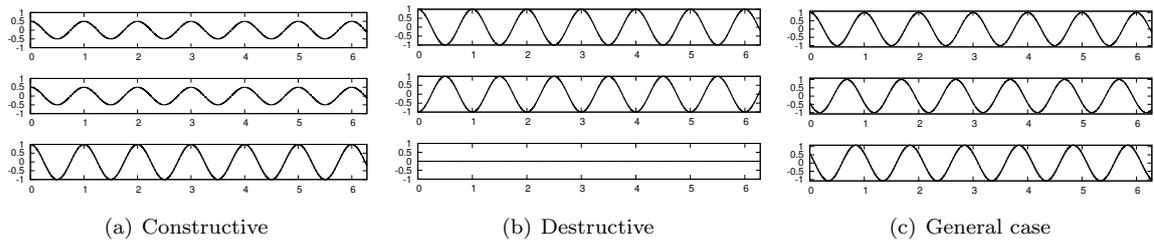


Figure 2.3: Interference between two waves, different cases. The superposition of the top two waves gives as result the bottom wave.

Depending on the difference between the phase of the two interfering waves there can be different kinds of interference (see figure 2.3). If the phases of the two waves are equal (phase difference 0) there is *constructive* interference, what means that their corresponding amplitudes are added and the resulting phase does not change. If the phase difference is π , then there is *destructive* interference,

what means that one amplitude is subtracted from the other one (potentially extinguishing the wave). The general case is an intermediate point between constructive and destructive interference.

Interference is actually a simple addition of waves. It can be proved that the addition of two sine waves is another sine wave:

$$A_1 \cos(2\pi\lambda^{-1} - \omega t + \delta_1) + A_2 \cos(2\pi\lambda^{-1} - \omega t + \delta_2) = A_3 \cos(2\pi\lambda^{-1} - \omega t + \delta_3) \quad (2.10)$$

where A_1 and A_2 are the amplitudes, δ_1 and δ_2 are the phases and A_3 and δ_3 are the resulting amplitude and phase. In order to compute A_3 and δ_3 , the easiest way is to realize that a sine wave of the form $A \cos(C + \delta)$ can be expressed as a phase vectors, also named *phasors* (Gia89) (if C is time invariant). The representing phasor has magnitude A and angle δ (see figure 2.4), and as a consequence interference can be solved by phasor addition (two dimensional vectors). These two dimensional vectors are often algebraically considered as complex numbers, as they share the same properties. The algebraic operation of phasors is used to calculate all operations between sine waves.

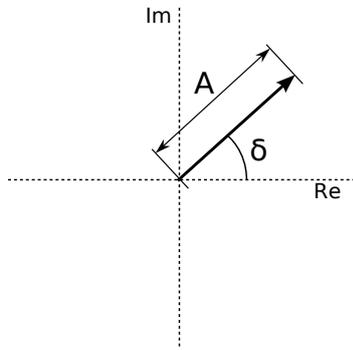


Figure 2.4: A phasor can be represented by a complex number (cartesian coordinates)

In the case of planar electromagnetic waves, a fully polarized light can be represented by the combination of two waves perpendicular to the propagation direction (see section 2.1.1.1). Therefore interference of two fully polarized electromagnetic waves is equivalent to the addition of two pairs of phasors (see figure 2.5). In order to calculate interference between non fully polarized electromagnetic waves, they can be decomposed as linear superposition of fully polarized waves.

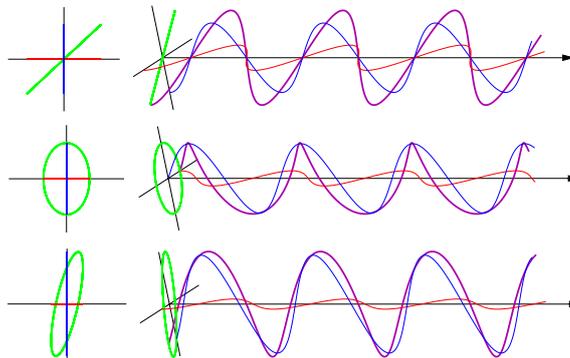


Figure 2.5: Interference between fully-polarized electromagnetic waves

2. TRANSPORT OF ELECTROMAGNETIC RADIATION

Interference has not been very relevant for the computer graphics community, apart from some specific works (GMN94), as often it is not perceivable. However, it is possible to perceive interference for instance on rainbows (more specifically, on the corresponding supernumerary arcs of the rainbow). As contribution of this PhD, a simulator of phase functions for rainbow rendering that takes interference (and polarization) into account has been developed (see section 4.4).

2.1.1.3 Other wave phenomena

There are still other phenomena related to the wave-like nature that can be considered. For instance, *surface waves* (mostly seen on fluids) are waves that propagate along the interface between two media. Also, *diffraction* happens when a wave encounters an obstacle that is of a size comparable to the wavelength. The wave scatters into multiple directions as a diffraction interaction happens. This can occur on thin slits or even on the differential edge of objects.

The behavior of electromagnetic waves under these phenomena is out of the scope of this PhD but can be found in optic books (Cow08).

2.1.2 Particle-like properties of electromagnetic radiation

Light exhibits particle-like properties as well as wave-like ones. Quantum optics deals with these particle-like properties, and define the fundamental elementary quantum of energy of light as the *photon*.

A photon is the fundamental unit for light and its energy is defined to be $hc\lambda^{-1}$ where h is *Planck's constant* and c is the speed of light. A light ray can be modeled as a stream of photons that interact with other quantum particles that define the matter. These interactions can turn into energy losses for the photon, or even energy gains. Quantum mechanics define momentum and spin momentum properties for photons, that can be related to polarization states (see section 2.1.1.1).

Some light transport concepts can be defined easier as particles than as waves (for instance, absorption on section 2.3 and inelastic scattering on section 2.3.3). A deeper insight into quantum optics concepts can be found on books about the topic (SZ97).

2.2 Index of refraction and wave propagation

The index of refraction of a medium is defined as the ratio of the velocity of a wave phenomenon on that medium according to its velocity on a reference medium. In the case of light, this reference medium is usually free space.

$$n = \frac{c}{v} = \sqrt{\frac{\mu\epsilon}{\mu_0\epsilon_0}} \quad (2.11)$$

μ is the magnetic permeability of the medium and ϵ is the electric permittivity of the medium. The rest of the symbols (c , μ_0 and ϵ_0) are the corresponding equivalents on free space, and have been defined in section 2.1.

2.2.1 Eikonal equation and Fermat's principle

From the definition of the index of refraction (Equation 2.11) and Maxwell equations, the eikonal equation can be deduced (BW02):

$$|\nabla \mathcal{F}(\vec{r})|^2 = n^2(\vec{r}) \quad (2.12)$$

where: \vec{r} represents a differential point in space and \mathcal{F} is defined as the optical path (a scalar value), also called eikonal function, related to the path of light. Both n and \mathcal{F} can vary at every differential point in space.

A discontinuity on n marks at conceptual level the boundary between two media, and under this boundary equation 2.12 presents (at least) two valid solutions, that are related to two different path of light: reflection and refraction. By studying the differential behavior of this equation, the full path of light between any two points in space can be deduced.

Fermat's principle is a variation (and a particular case) of the formulation of the eikonal equation. It says that a light ray, in going between two points, must traverse an optical path length which is stationary with respect to variations of the path. By "stationary" it means that it can be either a minimum (the usual), a maximum or a saddle point. If we define the optical path length l from a point \vec{a} to a point \vec{b} as

$$l = \mathcal{F}(\vec{b}) - \mathcal{F}(\vec{a}) = \int_{\vec{a}}^{\vec{b}} n(\vec{s}) d\vec{s} \quad (2.13)$$

where \vec{s} defines all the differential points traversed by light. Fermat's principle can be written as $\delta l = 0$. Notice the relationship between Fermat's principle and the eikonal equation in relation with the eikonal function \mathcal{F} and the optical path length l .

2.2.2 Non-linear media

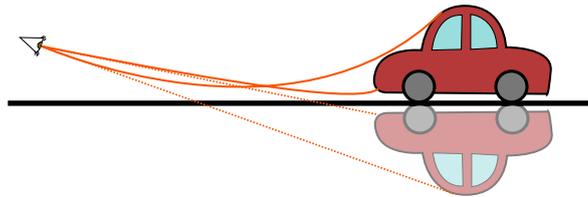


Figure 2.6: A non-linear media (with non-constant index of refraction) makes light follow a curved path, generating interesting phenomena such as mirages.

According to Fermat's principle, if the index of refraction in a medium is constant, then the path of light is a straight line. However, most of the media are not homogeneous, but they present small variations on the index of refraction along their volume. This provokes that the trajectory of light is curved. Most of the times these variations are visually imperceivable by the human eye, although in some cases the path is altered in such a way that the resulting phenomena is perceived.

That is the case of mirages: when a driver sees some water on the road that magically disappears, or on a dessert some water pools seem to appear and disappear on the sand. The optical phenomenon

2. TRANSPORT OF ELECTROMAGNETIC RADIATION

that is causing this perfection is related to the fact that the air that is close to the sand or the asphalt on the road is at very high temperatures, while upper layers of air are colder. The temperature of air is directly related to the index of refraction of the light, so as a consequence there is a smoothly varying index of refraction. The mirage of the water pool is just a set of light rays which are curved enough to get into the sky, giving the illusion of water on the floor (see figure 2.6).

Part of the work of this PhD is related to the efficient and accurate simulation of non-linear media. There is already some previous research on simulating non-linear media (BTL90; Mus90; LHM05; Grö95) although those works are restricted to specific distributions on the index of refraction, while this PhD is built upon an algorithm that is general and independent from specific distributions (SGGC05; GSMA06). Section 4.3 contains the techniques used for an efficient simulation method, while section 5.4 shows the results of the simulations and the corresponding studies about speedups and efficiency of the algorithm.

2.2.3 Dielectric vs. dielectric interface

A dielectric medium is a medium that is non conducting, an insulator. Ideally, this means that its electric field is completely isolated, and as a consequence electromagnetic radiation is not absorbed as it traverses the medium. Although all media are not absolutely dielectric, when absorption is negligible they can be approximated as one. The boundary between two dielectric media is characterized by the lack of absorption on any of the two media. The equivalent deductions for conducting (absorbing) media can be found on section 2.3.2

When a ray of light reaches an interface between two dielectrics with different indices of refraction a discontinuity on n appears along the surface of the interface. The eikonal equation then presents two solutions, related to the reflection and refraction of light (notice that Fermat's principle can also be used for this deduction). These two solutions are the law of reflection (equation 2.14) and Snell's law of refraction (equation 2.15):

$$\theta_r = \theta_i \tag{2.14}$$

$$n_t \sin(\theta_t) = n_i \sin(\theta_i) \tag{2.15}$$

where θ_i is the angle of incidence (measured to the normal to the interface), θ_r is the angle of the reflected light, θ_t is the angle of the refracted ray and n_i and n_t are the indices of refraction of the two media. Figure 2.7 depicts these light paths.

Equation 2.15 might not have a solution for certain angles, depending on the values of the indices of refraction. In this case, all the light is reflected. This phenomenon is called *total internal reflection*.

Both on reflection and refraction, the two components of light are scaled according to Fresnel coefficients. As light is an electromagnetic wave, the component of light which is parallel to the plane of the interaction (the plane that contains incident ray and normal) is scaled by one coefficient, while the component that is perpendicular to the plane of interaction is scaled by another one.

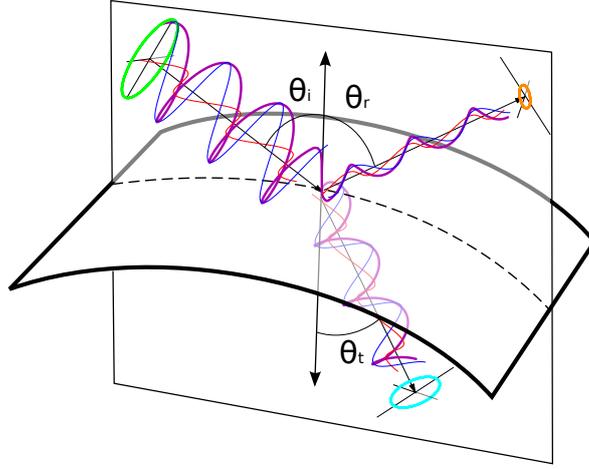


Figure 2.7: Interaction on a dielectric vs. dielectric interface. Snell's law and the law of perfect reflection define the paths of the refracted and reflected rays, while Fresnel coefficients define how each of the components of light are affected by reflection or refraction.

$$\begin{aligned}
 r_{\perp} &= \frac{\sin(\theta_t - \theta_i)}{\sin(\theta_t + \theta_i)} \\
 r_{\parallel} &= \frac{\tan(\theta_t - \theta_i)}{\tan(\theta_t + \theta_i)} \\
 t_{\perp} &= 1 - r_{\perp} \\
 t_{\parallel} &= 1 - r_{\parallel}
 \end{aligned} \tag{2.16}$$

where r_{\perp} is the factor that scales the amplitude of the perpendicular component of the incoming light for reflection, r_{\parallel} is the factor that scales the amplitude of the parallel component of the incoming light for reflection and t_{\perp} and t_{\parallel} are the corresponding factors for refraction (see figure 2.7). In the case of total internal reflection $r_{\perp} = r_{\parallel} = 1$ and $t_{\perp} = t_{\parallel} = 0$. Furthermore, when total internal reflection happens, the phases of the different components of the reflected ray are modified. The phase shifts δ_{\parallel} and δ_{\perp} for the reflected fields parallel and perpendicular to the plane of incidence under total reflection are (GMN94):

$$\begin{aligned}
 \tan(\delta_{\parallel}/2) &= -\frac{\sqrt{\sin^2 \theta_i - n^2}}{n^2 \cos \theta_i} \\
 \tan(\delta_{\perp}/2) &= -\frac{\sqrt{\sin^2 \theta_i - n^2}}{\cos \theta_i}
 \end{aligned} \tag{2.17}$$

Equation 2.16 models a medium boundary at which light is either reflected or transmitted, without any kind of energy loss. This is a good approximation when absorption is negligible compared to other phenomena. Next section considers that this absorption is not negligible and therefore must be considered.

2. TRANSPORT OF ELECTROMAGNETIC RADIATION

2.3 Absorption

When light traverses a medium which presents some conductivity then the power of the electromagnetic radiation is systematically taken up by the medium. As photons interact with the electrons of the atoms, their energy disappears. This power is *absorbed* by the corresponding material, usually being transformed into another form of energy, such as heat.

Actually, most of the solid materials are conductive up to some certain point, and therefore exhibit some absorption. The color perceived for an opaque object comes from the integration of the energy at differential wavelengths that has not been absorbed by the object.

The conductivity of a specific medium σ can be defined as the ratio between the electric field \vec{E} and the current density vector \vec{J} , so $\vec{J} = \sigma \vec{E}$. Dielectric materials presented on previous section have $\sigma = 0$ so there is no current on the material ($\vec{J} = \vec{0}$). This enables simplifications for the deductions from Maxwell's equations (see section 2.1), that are no longer possible with non-dielectric materials ($\sigma > 0$).

However it can be proven (BW02) that including the conductivity into Maxwell's equations can be done by defining the *complex electrical permittivity* $\hat{\epsilon}$ as follows:

$$\hat{\epsilon} = \epsilon + i \frac{4\pi\sigma}{\omega} \quad (2.18)$$

and therefore the complex index of refraction \hat{n} as:

$$\hat{n} = \mu \hat{\epsilon} = n(1 + i\kappa) \quad (2.19)$$

where κ is the *attenuation index*, that can be calculated by substituting on equation 2.18:

$$\kappa = \frac{2\pi\mu\sigma}{\omega n^2} \quad (2.20)$$

This attenuation index is related to the amount of power absorbed by a medium as light traverses it. It depends on wavelength, so the absorption at some wavelengths might be different than at other wavelengths.

2.3.1 Absorbing media

As light traverses an absorbing medium, its irradiance E gets diminished at every differential point of the optical path l as follows:

$$\frac{dE}{dl} = -\sigma_a E \quad (2.21)$$

where σ_a is the the *absorption coefficient*¹. This coefficient can be deduced from the definition of the attenuation index:

$$\sigma_a = \frac{4\pi}{\lambda} \kappa \quad (2.22)$$

In the case of a homogeneous constant σ_a the attenuation of an irradiance E_0 after the light has traveled an optical path d can be easily deduced from equation 2.21:

$$E = E_0 e^{-d\sigma_a} \quad (2.23)$$

what means that the intensity is decaying by a factor of e^{-1} when the light has traveled a distance $\sigma_a^{-1} = \lambda(4\pi\kappa)^{-1}$. If κ is several orders of magnitude than the wavelength λ , light gets almost completely absorbed in less than a millimeter. This average traveled distance is often called *optical distance* or *optical thickness*.

2.3.2 Dielectric vs. metal interface

The equations that define the interface between a dielectric and a metal (conductive material) can easily be deduced from Snell's law and Fresnel equations (see section 2.2.3), by substituting the index of refraction n for the complex index of refraction \hat{n} . By doing so, it can be noticed that the phase of each of the components of the reflected wave is altered in most cases. Also, for any specific wavelength, complex indices of refraction lead to the logical result that part of the incident intensity is not reflected nor transmitted (it is absorbed).

The light that is transmitted inside the metal then gets absorbed as stated in the previous section. Depending on the absorption coefficient σ_a (that depends on κ and in the end on the conductivity of the medium) light travels shorter or further until it gets almost completely absorbed. A value of $\sigma_a = 0$ shows that the material is a dielectric (as conductivity is 0 too) and therefore the corresponding object is *transparent*. The bigger the values of σ_a , the shorter distance is traveled by light before getting almost extinct. At some point this distance becomes negligible and imperceivable, and therefore the corresponding object is *opaque*. Intermediate values correspond to *translucent* materials.

¹The notation for the absorption coefficient σ_a should not be confused with the notation for the conductivity σ . This confusing notation is owed to the fact that the notation for the conductivity has been taken from optic books while the notation for the absorption coefficient comes from rendering books. As the absorption coefficient is widely used for rendering participating media, the most widely used notation has been preferred. It appears with this notation in the rest of the text.

2. TRANSPORT OF ELECTROMAGNETIC RADIATION

2.3.3 Inelastic scattering

Absorption has been defined as a stream of photons that interact with the electrons of a conductive material. As electrons statistically interact with photons and absorb them from the stream, the resulting irradiance gets diminished. However, there can be different energy transfers between a photon and a fundamental particle that interacts with it. Sometimes it is possible for the photon to interact with a molecule or fundamental particle in such a way that the absorption of energy is partial. According to the definition of photon energy in section 2.1.2, a change of energy implies a change on wavelength. This effect is called *inelastic scattering*, and is the cause of several phenomena such as fluorescence or Raman scattering (Mob94).

Photon interactions can be classified in three broad categories: absorption (as in section 2.3), elastic scattering (as in section 2.2.3) or inelastic scattering, depending on whether the scattered photon loses, maintains or changes its energy in the process. The inelastic scattering events can be further sub-classified according to the nature of the energy transfer: *Stokes* scattering, when a molecule of the medium absorbs the photon and re-emits it with a lower energy (having transferred energy to the medium), and *anti-Stokes* scattering, when the re-emitted photon absorbs energy from the fundamental particles in the medium. Anti-Stokes scattering is less common than Stokes, although there are some phenomena such as Raman scattering that can be classified as anti-Stokes.

The inelastic scattering process implies an energy transfer from wavelength λ_i to λ_o , with λ_i being the excitation wavelength and λ_o the re-emitted wavelength. Stokes implies a shift towards longer wavelengths, whereas in anti-Stokes scattering the scattered photon has a shorter wavelength. The probability of each of these transfers to happen depends completely on the nature of the particle itself, so it is no universal expression for defining inelastic scattering, apart from defining the probability of each differential wavelength λ_i to be inelastically scattered to each differential wavelength λ_o .

A relevant contribution of this PhD is related to the modeling of inelastic scattering on participating media (see section 3.5) and an algorithm that simulates this phenomenon (see section 4.2).

2.4 Light scattering

The behavior of a differential electromagnetic wave has been defined on previous sections. The integration on the continuum of the differential electromagnetic waves would give as result the lighting of a scene. However, in practice, using electromagnetic theory for any task would become really daunting since the very beginning. Would it make sense to measure the amplitude of every differential wave on a sensor? Would it be practical to simulate the continuum of light with differential photons for rendering a scene? The answer is obviously no, or at least not always.

Computer Graphics and radiometry deal with quantities that globally take into account the differential behavior of light, enabling their practical measurement and simulation. The most relevant ones can be found on table 2.1. Given one of those quantities, we can define also its spectral form by considering the quantity per nanometer. For instance, spectral irradiance would be measured in watts per square meter per nanometer (including the wavelength).

The branch of Computer Graphics that deals with physically-based rendering is heavily based on these radiometric quantities (PH04; DBB02). The irradiance E for electromagnetic waves has been defined in equation 2.8, and the rest of radiometric quantities can be computed by differentiating irradiance with respect to the solid angle (for radiance), integrating along the surface area (for flux) or both (intensity). Therefore, the irradiance E is the quantity that enables us to relate the electromagnetic differential optics with the radiometric units that are often used for computer graphics.

Quantity	Description	Symbol	Unit
Flux	Energy per unit time	Φ	Watt (W)
Intensity	Power per unit solid angle	I	Watt per steradian - $\left(\frac{W}{sr}\right)$
Irradiance	Incident power per unit surface	E	Watt per square meter - $\left(\frac{W}{m^2}\right)$
Exitance	Power emitted per unit surface	M	Watt per square meter - $\left(\frac{W}{m^2}\right)$
Radiance	Power per projected area per unit solid angle	L	Watt per steradian per square meter - $\left(\frac{W}{sr \cdot m^2}\right)$

Table 2.1: Radiometric quantities

2.4.1 Participating media

Participating media are those media which alter in some way the electromagnetic radiation that traverses them. Section 2.2.2 has shown how a media can alter the path of light by having a non-constant index of refraction along its volume, so non-linear media are actually a subset of participating media. Furthermore, section 2.3.1 presents absorbing media, in which the power of the electromagnetic radiation gets diminished (transformed into heat as light traverses it).

Also, other interactions of participating media include the possibility of containing small floating particles (as fog or smoke). These small particles potentially have different optical properties and therefore act as different media. As a consequence, these interfaces between the hosting medium and the particles act according to what has been commented in sections 2.2.3 and 2.3.2. The particles are potentially small, even microscopical, with refractions, reflections and absorption interactions happening at a very differential level (see figure 2.8). Given its high dimensionality and its interesting visual effects, participating media have been very relevant for the Computer Graphics community (CPP+05; GNJJ08; GJDN09; GJJD09)

However, all these differential scattering interactions can be defined using two statistical properties:

- The *scattering coefficient* σ_s that defines the amount of radiance that gets scattered per traveled unit distance.
- The *phase function* $p(\vec{\omega}_i, \vec{\omega}_o)$ that defines the statistical angular distribution of outgoing radiation at a differential scattering event.

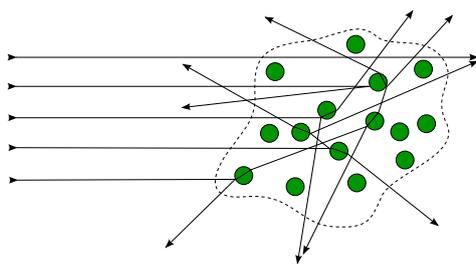


Figure 2.8: Differential scattered rays can be statistically modeled by a scattering coefficient and a phase function

Depending on the point of view, scattering consists of:

- From one side, light that is traversing a specific path gets scattered and therefore there exist a differential attenuation on that path of light. This is often called *out-scattering*.

2. TRANSPORT OF ELECTROMAGNETIC RADIATION

- From the other side, light that is traversing a specific path receives the radiance scattered from other paths of light into its corresponding direction. This is often called *in-scattering*.

The expression that defines the behavior of out-scattering is:

$$\frac{dL}{dl} = -\sigma_s L \quad (2.24)$$

Notice that σ_s might not be homogeneous, although if it is homogeneous, solving Equation 2.24 is trivial. Figure 2.9 explains the concept of out-scattering.

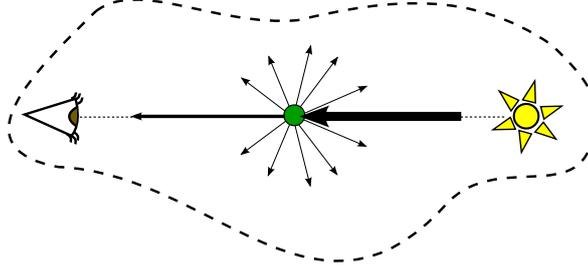


Figure 2.9: Out-scattering: As the ray traverses the participating medium, an amount of radiance given by the scattering coefficient is scattered outside the path of light.

For in-scattering it is needed to know the angular distribution of the scattered radiance. This information is given by the phase function, that can be defined as follows

$$p(\vec{\omega}_i, \vec{\omega}_o) = \frac{dL_o(\vec{\omega}_o)}{dL_i(\vec{\omega}_i)} \quad (2.25)$$

where ω_i represents the direction of the incident radiance L_i and ω_o represents the direction of the outgoing radiance L_o . Notice that this phase function might vary at every single differential point in the volume, therefore becoming non-homogeneous and including a dependency on \vec{r} (the point in the volume). Thus, the outgoing radiance from in-scattering can be expressed as follows:

$$\frac{dL_o(\vec{\omega}_o)}{dl} = \sigma_s \int_{\Omega} p(\vec{\omega}_i, \vec{\omega}_o) L_i(\vec{\omega}_i) d\vec{\omega}_i \quad (2.26)$$

where \int_{Ω} refers to the integral of all the differential incident directions along a sphere. Equation 2.26 can be explained by considering that at every differential point in the optical path, part of all incident radiance (coming from all possible directions) is transferred to the path at that differential point. This can be seen on figure 2.10. Also, dependency on the point on the volume \vec{r} could be included here.

The scattering coefficient and the phase function can depend on the wavelength (be different for each differential wavelength) or even on the differential point in the volume of the medium (non-homogeneous media). However, in order to be physically correct the phase function must always satisfy the following:

- It must be reciprocal for all $\vec{\omega}_i$ and $\vec{\omega}_o$: $p(\vec{\omega}_i, \vec{\omega}_o) = p(\vec{\omega}_o, \vec{\omega}_i)$.

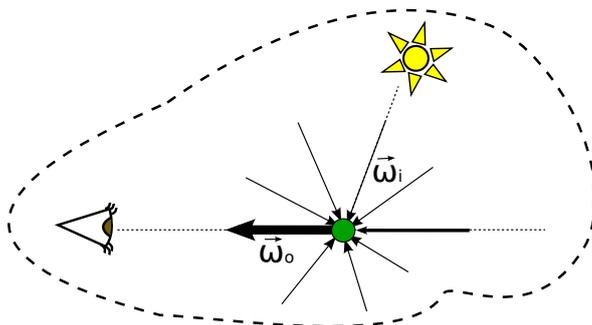


Figure 2.10: In-scattering: As the ray traverses the participating medium, an amount of radiance given by the scattering coefficient is scattered into an angular distribution given by the phase function.

- Total outgoing radiance must be equal to the total incoming radiance:

$$\int_{\Omega} p(\vec{\omega}_i, \vec{\omega}_o) d\vec{\omega}_i = 1 \quad (2.27)$$

If the incident irradiance comes directly from the light sources, the phenomenon is often called *single scattering*. However, the incident radiance might come from a previous interaction in the medium, what is called *multiple scattering*.

Some media can also emit light (maybe due to some chemical reaction, as in fire). This phenomena, often called *medium emission*, adds power to the corresponding optical path that traverses the medium¹. It can be modeled as a radiance L_e and considered as:

$$\frac{dL(\vec{\omega})}{dl} = L_e(\vec{\omega}) \quad (2.28)$$

Of course this medium emission can depend on wavelength and be non-homogeneous in the same way than absorption and scattering coefficients.

By adding equations 2.21, 2.24, 2.26 and 2.28 it is possible to get the Radiance Transfer Equation (RTE) (Cha60):

$$\frac{dL(\vec{\omega}_o)}{dl} = L_e(\vec{\omega}_o) - (\sigma_a + \sigma_s)L(\vec{\omega}_o) + \sigma_s \int_{\Omega} p(\vec{\omega}_i, \vec{\omega}_o) L_i(\vec{\omega}_i) d\vec{\omega}_i \quad (2.29)$$

that defines the variation on the radiance of a ray that traverses a participating medium. Its optical path is still defined by the eikonal equation (2.12). The terms that reduce the amplitudes along the path, are often added up into a coefficient called *extinction coefficient* $\sigma_k = \sigma_a + \sigma_s$. Equation 2.29 depends on each differential wavelength λ and each of the coefficients can depend on the position on the volume $\vec{r}(l)$, but the corresponding dependencies have been omitted for clarity.

¹This concept is included here for completeness. A more in-depth study of medium emission, although being so interesting that would require at least its own separate section, is out of the scope of this PhD. Following the same philosophy, the optical definitions of all kinds of light sources have been deliberately omitted in this text.

2. TRANSPORT OF ELECTROMAGNETIC RADIATION

Participating media and scattering in general are a very wide field of research (CPP+05; GNJJ08; GJDN09; GJJD09). Chapter 3 introduces some concepts about measuring and modeling participating media from the corresponding coefficients and phase functions, while chapter 4 briefly shows how can they be simulated. As contribution of this PhD, a case study of modeling a participating media from measured data has been done (section 3.5) and an algorithm to properly simulate this model has been designed and developed (section 4.2). Furthermore, another contribution is the modeling of a phase function by simulating the differential interactions between the electromagnetic waves and a model of a particle (section 4.4).

2.4.2 Surfaces: the boundary between two media

As it has been shown on previous sections, the interface or boundary between two different media interacts with light in several ways (absorption, reflection, refraction...). This boundary can be modeled as a surface: the surface of an object made of metal inside the atmosphere is actually the boundary between air and metal. Actually, in computer graphics, it is common to define the boundary between two different media with a surface expressed mathematically or by a mesh of triangles.

The interaction between light and flat surfaces can be easily be simulated and modeled by Fresnel equations or their complex counterpart (sections 2.2.3 and 2.3.2). However, when the surface presents some edges or rugosities, these equations must be applied differentially along the surface, considering potential auto-occlusions (facets hiding other facets), self-shadows (facets that interrupt the path of light) and different orientations (see figure 2.11). Furthermore, this rugosities could be microscopical (what is called *micro-geometry*).

Instead of considering each differential interaction, all the interactions can be gathered into a function that statistically define the differential geometry of the material.

A Bidirectional Reflectance Distribution Function (BRDF) considers the distribution of all the differential reflection interactions (see figure 2.11). It can be defined as follows:

$$f_r(\vec{\omega}_i, \vec{\omega}_o) = \frac{dL_r(\vec{\omega}_o)}{dE(\vec{\omega}_i)} \quad (2.30)$$

where f_r represents the BRDF, ω_i represents the direction of the incident irradiance E and ω_o represents the direction of the reflected radiance L_r (see figure 2.12). This function gives a factor for the reflected radiance given an input and an output direction. Notice that this definition of the BRDF can be extended so it also depends on the interaction point of the surface (spatially varying BRDF), on the wavelength (spectral BRDF) or even on time (time dependent BRDF).

Equation 2.30 can be rewritten so it is possible to compute the reflected radiance:

$$L_r(\vec{\omega}_o) = \int_{\Omega} f_r(\vec{\omega}_i, \vec{\omega}_o) E(\vec{\omega}_i) d\vec{\omega}_i = \int_{\Omega} f_r(\vec{\omega}_i, \vec{\omega}_o) L_i(\vec{\omega}_i) (\vec{w}_i \cdot \vec{n}) d\vec{\omega}_i \quad (2.31)$$

where \int_{Ω} refers to the integral over all the hemisphere of incident directions, and L_i is the incident radiance. Notice also in this integral the equality $E(\vec{\omega}_i) = L_i(\vec{\omega}_i) (\vec{w}_i \cdot \vec{n})$. Equation 2.31 is often called the *render equation* (Kaj86).

The render equation provides an expression that calculates all the possible outgoing radiance in any direction in the hemisphere, considering all the incoming radiance in the whole hemisphere. This incident irradiance can come from light sources (direct illumination) or can come as outgoing radiance from another interaction that is defined by the same equation (indirect illumination).

In order to be physically correct, the BRDF must satisfy both the following:

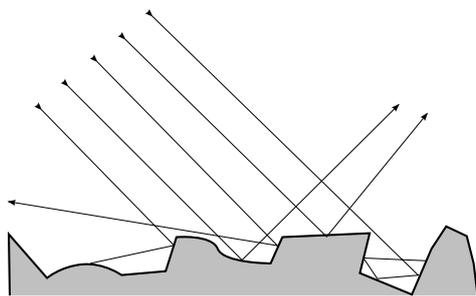


Figure 2.11: Differential reflected rays can be statistically modeled by a BRDF

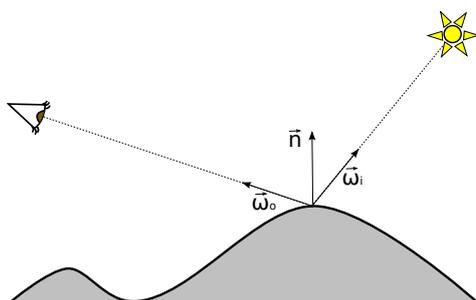


Figure 2.12: BRDF

- It must be reciprocal for all directions $\vec{\omega}_i$ and $\vec{\omega}_o$: $f_r(\vec{\omega}_i, \vec{\omega}_o) = f_r(\vec{\omega}_o, \vec{\omega}_i)$.
- It must conserve energy for all incoming directions $\vec{\omega}_i$: $\int_{\Omega} f_r(\vec{\omega}_i, \vec{\omega}_o) (\vec{\omega}_i \cdot \vec{n}) d\vec{\omega}_i \leq 1$

While the integral of the phase function had to equal 1, BRDFs take into account both absorption on the surface and self shadowing due to rugosities on the surface, and therefore its integral can equal less than one.

In the same way than the BRDF is defined, the Bidirectional Transmission Distribution Function (BTDF) considers the distribution of all transmitted (refracted) rays along a surface (figure 2.13). This function can be noted as $f_t(\vec{\omega}_i, \vec{\omega}_o)$, and its definition and integration is analogous to the BRDF (equations 2.30 and 2.31) with the sole difference that in this case $\vec{\omega}_i$ and $\vec{\omega}_o$ point towards different sides on the surface (see figure 2.14). Notice that, in the case of absorbing media with an attenuation index greater than a specific amount, there is no need for this function, as light would get extinct in a very short distance.

However, a BRDF (even a spatially varying BRDF) does not consider all the differential interactions that happen on a surface. A ray could potentially be transmitted inside the surface and exit the surface at a different point, maybe due to the micro-geometry, or maybe due to particles below the surface (see figure 2.15). These interactions that happen under the surface are widely known as *subsurface scattering* and can be modeled by a Bidirectional Subsurface Reflection Distribution Function (BSSRDF). Materials that present these kind of interactions are, for instance, translucent materials such as marble or skin, in which the attenuation index is below a certain level so light, although attenuated, is able to travel enough distance inside the object.

2. TRANSPORT OF ELECTROMAGNETIC RADIATION

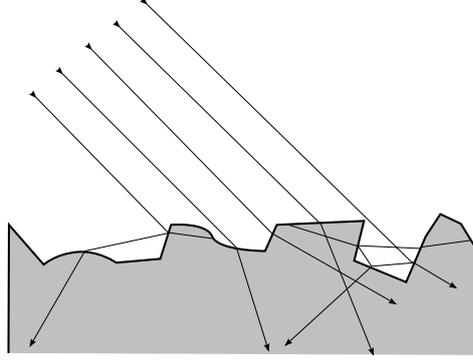


Figure 2.13: Differential transmitted (refracted) rays can be statistically modeled by a BTDF

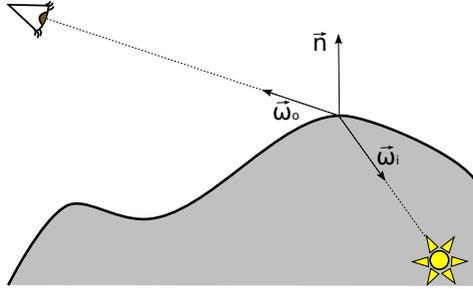


Figure 2.14: BTDF

The BSSRDF can be defined as follows:

$$f_{ss}(\vec{r}_i, \vec{\omega}_i, \vec{r}_o, \vec{\omega}_o) = \frac{dL_{ss}(\vec{r}_o, \vec{\omega}_o)}{dE(\vec{r}_i, \vec{\omega}_i)} \quad (2.32)$$

where \vec{r}_i and \vec{r}_o are the incident and outgoing points, respectively. This means that light can potentially enter the surface at one point and exit it at a different point. Notice that a BRDF is a particular case of a BSSRDF in which $\vec{r}_i = \vec{r}_o$. See figure 2.16 for details.

Computing subsurface scattering is more complex than a single reflection, but can be done as follows:

$$L_{ss}(\vec{r}_o, \vec{\omega}_o) = \int_A \int_{\Omega} f_{ss}(\vec{r}_i, \vec{\omega}_i, \vec{r}_o, \vec{\omega}_o) E(\vec{r}_i, \vec{\omega}_i) d\vec{\omega}_i d\vec{r}_i \quad (2.33)$$

in which \int_A integrates along the area of all the differential points along the surface of the object. The dependency on the differential point on the surface, that was ignored on equation 2.31, now becomes relevant.

Subsurface scattering happens at boundaries between two media in which (at least) one of them is a participating media (as in section 2.4.1), and as a consequence it is possible to model the process of subsurface scattering as a participating media: with absorption and scattering coefficients, plus a phase function.

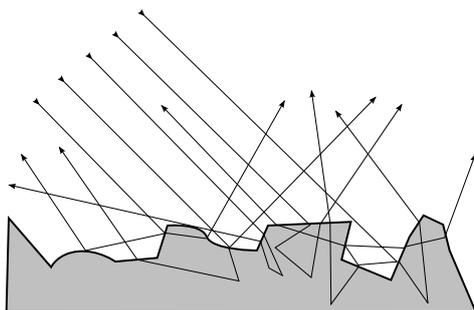


Figure 2.15: A more general case of the BRDF. Differential rays that are reflected or interact under the surface and exit at a different location can be modeled by a BSSRDF

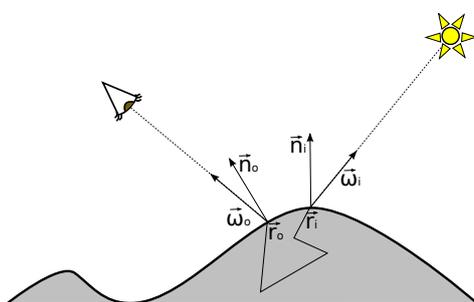


Figure 2.16: BSSRDF explanation

Modeling a BRDF, BTDF or BSSRDF is still a field of research. There are several models that are based on different more or less intuitive parameters, that have been measured using different techniques or that have been simulated. This is further developed on chapter 3. Part of the work of this PhD is related to the estimation of these functions from rendered images (called inverse rendering, see section 3.2), and further specializes into estimating plausible BSSRDF (the most complex case) from single HDR photographs (see section 3.3). Chapter 4 briefly introduces how to simulate global illumination considering these functions.

2.4.3 Wavelength redistribution

The inelastic scattering process (presented in section 2.3.3) implies an energy transfer from wavelength λ_i to λ_o with a specific probability, and therefore can be modeled as a function $\gamma(\lambda_i, \lambda_o)$ that defines the probability of an excitant photon of wavelength λ_i being inelastically scattered and shifting its wavelength to λ_o , and is called *energy transfer function* (see figure 2.17). This function should satisfy $\int_W \gamma(\lambda_i, \lambda_o) d\lambda_i \leq 1$ for all outgoing wavelengths (where \int_W refers to the integral along the whole wavelength spectrum). When $\lambda_i = \lambda_o$, this function represents the specific case of elastic scattering.

In order to include this wavelength redistribution into participating media, a new inelastic term must be added to the Radiative Transfer Equation (equation 2.29), giving a new equation called the Full Radiative Transfer Equation:

2. TRANSPORT OF ELECTROMAGNETIC RADIATION

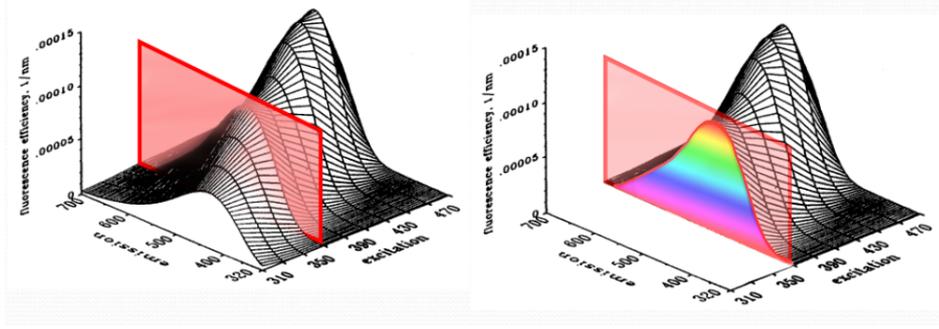


Figure 2.17: Energy transfer function

$$\begin{aligned} \frac{dL(\vec{\omega}_o, \lambda_o)}{dl} = & L_e(\vec{\omega}_o, \lambda_o) - \left(\sigma_a(\lambda_o) + \sigma_s(\lambda_o) + \int_W \gamma(\lambda_i, \lambda_o) d\lambda_i \right) L(\vec{\omega}_o, \lambda_o) + \\ & \sigma_s(\lambda_o) \int_{\Omega} p(\vec{\omega}_i, \vec{\omega}_o, \lambda_o) L_i(\vec{\omega}_i, \lambda_o) d\vec{\omega}_i + \\ & \int_W \gamma(\lambda_i, \lambda_o) \int_{\Omega} p_2(\vec{\omega}_i, \vec{\omega}_o, \lambda_i) L_i(\vec{\omega}_i, \lambda_i) d\vec{\omega}_i d\lambda_i \end{aligned} \quad (2.34)$$

in which p_2 is a phase function, different from p , related to the angular distribution of inelastic scattering, that is most of the times isotropic. Notice that this time the dependencies on the wavelength have been made explicit (they were implicit on equation 2.29). It is possible to use a more general inelastic phase function on the last term of previous equation as $p_i(\vec{\omega}_i, \vec{\omega}_o, \lambda_i, \lambda_o)$ (which would be equivalent in the inelastic term to $\gamma(\lambda_i, \lambda_o) p_2(\vec{\omega}_i, \vec{\omega}_o, \lambda_i)$), which would enable also the possibility of different angular distributions dependent both on the incident and outgoing wavelengths. However, participating media do not usually show such dependencies and measuring so many degrees of freedom could be quite unpractical, so the full inelastic phase function is often omitted.

A related dimensionless function that describes inelastic scattering is the *quantum yield* $\Gamma(\lambda_i)$, defined as the total number of photons emitted at all wavelengths divided by the number of photons absorbed at excitation wavelength λ_i . It is related to the spectral quantum efficiency function by:

$$\Gamma(\lambda_i) = \int_W \gamma(\lambda_i, \lambda_o) d\lambda_i \quad (2.35)$$

Notice that this quantum yield can be seen as an inelastic scattering coefficient σ_i . As a consequence a new function must be defined for considering the dependency between wavelengths:

$$f_{\lambda}(\lambda_i, \lambda_o) = \frac{\gamma(\lambda_i, \lambda_o)}{\sigma_i(\lambda_o)} \quad (2.36)$$

where this new function $f_{\lambda}(\lambda_i, \lambda_o)$ is called *wavelength redistribution function* and satisfies $\int_W f_{\lambda}(\lambda_i, \lambda_o) d\lambda_i = 1$. These terms can be included into the FRTE, giving the following equation:

$$\begin{aligned}
\frac{dL(\vec{\omega}_o, \lambda_o)}{dl} = & L_e(\vec{\omega}_o, \lambda_o) - (\sigma_a(\lambda_o) + \sigma_s(\lambda_o) + \sigma_i(\lambda_o))L(\vec{\omega}_o, \lambda_o) + \\
& \sigma_s(\lambda_o) \int_{\Omega} p(\vec{\omega}_i, \vec{\omega}_o, \lambda_o) L_i(\vec{\omega}_i, \lambda_o) d\vec{\omega}_i + \\
& \sigma_i(\lambda_o) \int_W \int_{\Omega} f_{\lambda}(\lambda_i, \lambda_o) p_2(\vec{\omega}_i, \vec{\omega}_o, \lambda_i) L_i(\vec{\omega}_i, \lambda_i) d\vec{\omega}_i d\lambda_i
\end{aligned} \tag{2.37}$$

which is another representation for the FRTE.

For surfaces, inelastic scattering is also easily considered just by integrating along the whole spectrum and considering the potential energy transfers. As an example, inelastic reflection (without subsurface scattering) can be computed by the following equation (based on the render equation 2.31):

$$L_r(\vec{\omega}_o, \lambda_o) = \int_W \gamma(\lambda_o, \lambda_i) \int_{\Omega} f_r(\vec{\omega}_i, \vec{\omega}_o, \lambda_o) L_i(\vec{\omega}_i, \lambda_i) (\vec{\omega}_i \cdot \vec{n}) d\vec{\omega}_i d\lambda_i \tag{2.38}$$

where now the dependencies on wavelengths have been made explicit. In the same way than the phase function, a more general BRDF can be defined that extends both $\gamma(\lambda_o, \lambda_i)$ and $f_r(\vec{\omega}_i, \vec{\omega}_o, \lambda_o)$, called inelastic BRDF and noted $f_{ri}(\vec{\omega}_i, \vec{\omega}_o, \lambda_o, \lambda_i)$.

Having so many degrees of freedom, inelastic scattering is not easy to model. However some examples can be found on chapter 3. More specifically, section 3.5 show one of the contributions of this PhD, which is a model of ocean water that includes inelastic scattering. There are some previous works that can simulate inelastic scattering on surfaces (Gla94; WTP01) or even on participating media (CS04) with some limitations. As contribution of this research a new algorithm that is able to simulate global illumination with inelastic scattering in participating media free of the restrictions of previous approaches is presented in section 4.2.

2.5 Summary

This chapter introduces the basics of Optics that are relevant for physically-based Computer Graphics research. Starting from Maxwell's equations, several optics phenomena and the equations that define them have been presented as base for further chapters. The different deductions and variations over the introduced concepts lead to the research undertaken for this PhD in next chapters.

The work of this PhD is based into accurate modeling and simulations techniques for advanced light scattering phenomena. Starting from non-linear media, explained in section 2.2.2, several techniques for efficiently simulating them are presented in section 4.3, with results in the form of images and efficiency studies shown in section 5.4.

Accurate modeling of participating media scattering (introduced in section 2.4.1) is a daunting task, that has been also undertaken in section 4.4: an algorithm that is able to simulate phase functions for particles suspended in air enables us to render rainbows (see section 5.5) and that can be applied for any particle geometry. Furthermore, this simulation takes into account the wave nature of light (section 2.1.1), considering polarization (introduced in section 2.1.1.1) and interference (introduced

2. TRANSPORT OF ELECTROMAGNETIC RADIATION

in section 2.1.1.2), and taken into account the optics of dielectric vs. dielectric interfaces (see section 2.2.3).

Also, on this PhD, some phenomenon owed to the particle nature of light is taken into account: inelastic scattering (explained section 2.3.3), that is also present in some participating media, is studied and modeled in section 3.5 (using as example of such ocean water), while an algorithm able to render it is explained in section 4.2. The results of these simulations are shown in section 5.3.

In order to complete the potential of physically-based modeling, some algorithms have been developed in order to gather physically-based reflectance models (both BRDF and BSSRDF, introduced in section 2.4.2) from renders (see section 3.2). In this line, an algorithm has also been developed that estimates physically-based BSSRDFs from single photographs (see section 3.3).

References

- [BTL90] Marc Berger, Terry Trout, and Nancy Levit, *Ray tracing mirages*, IEEE Comput. Graph. Appl. **10** (1990), no. 3, 36–41. [16](#)
- [BW02] M. Born and E. Wolf, *Principles of optics: Electromagnetic theory of propagation, interference and diffraction of light*, Cambridge University Press, 2002. [10](#), [15](#), [18](#)
- [Cha60] S. Chandrasekhar, *Radiative transfer*, Dover Publications, Inc., 1960. [23](#)
- [Cow08] John M. Cowley, *Diffraction physics*, Elsevier Science Ltd, 2008. [14](#)
- [CPP⁺05] Eva Cerezo, Frederic Pérez, Xavier Pueyo, Francisco J. Seron, and François X. Sillion, *A survey on participating media rendering techniques*, vol. 21, 2005. [21](#), [24](#)
- [CS04] E. Cerezo and F. J. Seron, *Rendering natural waters taking fluorescence into account*, Computer Animation and Virtual Worlds **15** (2004), 1–14. [29](#)
- [DBB02] Philip Dutre, Philippe Bekaert, and Kavita Bala, *Advanced global illumination*, AK Peters, Ltd., July 2002. [20](#)
- [Gia89] Douglas C. Giancoli, *Physics for scientists and engineers*, Prentice Hall, 1989. [13](#)
- [GJDN09] Diego Gutierrez, Wojciech Jarosz, Craig Donner, and Srinivasa G. Narasimhan, *Scattering*, SIGGRAPH '09: ACM SIGGRAPH 2009 Courses (New York, NY, USA), ACM, 2009, pp. 1–397. [21](#), [24](#)
- [GJJD09] Diego Gutierrez, Henrik Wann Jensen, Wojciech Jarosz, and Craig Donner, *Scattering*, SIGGRAPH ASIA '09: ACM SIGGRAPH ASIA 2009 Courses (New York, NY, USA), ACM, 2009, pp. 1–620. [21](#), [24](#)
- [Gla94] A. S. Glassner, *A model for fluorescence and phosphorescence*, Fifth Eurographics Workshop on Rendering, Springer-Verlag, 1994, pp. 60–70. [29](#)
- [GMN94] Jay S. Gondek, Gary W. Meyer, and Jonathan G. Newman, *Wavelength dependent reflectance functions*, SIGGRAPH '94 (New York, NY, USA), ACM, 1994, pp. 213–220. [14](#), [17](#)
- [GNJJ08] Diego Gutierrez, Srinivasa G. Narasimhan, Henrik Wann Jensen, and Wojciech Jarosz, *Scattering*, SIGGRAPH Asia '08: ACM SIGGRAPH ASIA 2008 courses (New York, NY, USA), ACM, 2008, pp. 1–12. [21](#), [24](#)
- [Grö95] Meister Eduard Gröller, *Nonlinear raytracing: visualizing strange worlds*, Visual Computer **11** (1995), no. 5, 263–274. [16](#)
- [GSMA06] Diego Gutierrez, Francisco J. Serón, Adolfo Muñoz, and Oscar Anson, *Simulation of atmospheric phenomena.*, Computers & Graphics **30** (2006), no. 6, 994–1010. [16](#)

REFERENCES

- [Kaj86] James T. Kajiya, *The rendering equation*, SIGGRAPH Comput. Graph. **20** (1986), no. 4, 143–150. [24](#)
- [LHM05] Andrei Lințu, Jörg Haber, and Marcus Magnor, *Realistic solar disc rendering*, WSCG'2005 Full Papers Conference Proceedings (Vaclav Skala, ed.), 2005, pp. 79–86. [16](#)
- [LLT95] S.G. Lipson, H. Lipson, and D.S. Tanhauser, *Optical physics*, third edition ed., Cambridge, 1995. [10](#)
- [Max65] James Cler Maxwell, *A dynamical theory of the electromagnetic field*, Philosophical Transactions of the Royal Society of London (1865), no. 155, 459–512. [9](#)
- [Mob94] Curtis D. Mobley, *Light and water: Radiative transfer in natural waters*, Academic Press, Inc., San Diego, 1994. [20](#)
- [Mus90] F. Kenton Musgrave, *A note on ray tracing mirages*, IEEE Comput. Graph. Appl. **10** (1990), no. 6, 10–12. [16](#)
- [PH04] Matt Pharr and Greg Humphreys, *Physically based rendering: From theory to implementation*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004. [20](#)
- [SGGC05] F. J. Seron, D. Gutierrez, G. Gutierrez, and E. Cerezo, *Technical section: Implementation of a method of curved ray tracing for inhomogeneous atmospheres*, Comput. Graph. **29** (2005), no. 1, 95–108. [16](#)
- [SZ97] Marian O. Scully and M. Suhail Zubairy, *Quantum optics*, Cambridge University Press, 1997. [14](#)
- [WTP01] Alexander Wilkie, Robert F. Tobler, and Werner Purgathofer, *Combined rendering of polarization and fluorescence effects*, Proceedings of the 12th Eurographics Workshop on Rendering Techniques (London, UK), Springer-Verlag, 2001, pp. 197–204. [29](#)

Chapter 3

Physically - based modeling

3.1 Basics and previous work

All the equations that govern the behavior of electromagnetic radiation (see chapter 2) present variables, coefficients and functions that depend on the medium or material that light is interacting with. The definition of these, however, is not trivial. There are several ways of deducing or defining the parameters that govern the behavior of light when it interacts with the corresponding medium or material.

Intuitive models: Functions that present a high dimensionality (such as BRDFs and BSSRDFs) can be simplified by defining a function model based on intuitive parameters, such as rugosity, thickness or specularity, that a user or modeler can understand. One of the drawbacks is that there exists the possibility that the resulting model might not be physically plausible.

Measuring / acquiring: The parameters and functions can be estimated from reality with the appropriate measure equipment. Depending on the algorithm the complexity of the equipment setup can vary, from arrays of cameras with complex lighting setups to simple photographs. We refer the reader to the excellent work by Weyrich and colleagues ([WLL⁺08](#)) for a more comprehensive overview of appearance acquisition techniques.

Simulating: Another option also used in computer graphics for defining coefficients and material functions is to simulate the full phenomenon, by considering the microscopical interactions, the nature of electromagnetic waves or even Maxwell's equations.

Next subsections present a small overview about the previous work for physically-based modeling of indices of refraction, BRDFs, participating media and BSSRDFs.

3. PHYSICALLY - BASED MODELING

3.1.1 Index of refraction

Measuring indices of refraction for dielectric (and some non-dielectric) materials can easily be done by using a spectrometer (Jam07). Specific values for different materials have already been measured and can be found either around the world wide web (NPL) or from the publications of standardization associations (ftPoWS). However there has been little research dealing with the modeling of distributions of indices of refractions along volumes for non-linear media (see section 2.2.2), with some exceptions.

One of these exceptions is an accurate model of the atmosphere (GSMA06). This model is based on the USA 1976 Standard Atmosphere (USA76), and includes an user-driven Atmospheric Profile Manager, that enables to include inversion layers (zones in which temperature does not follow the standards) for de-standardization. A remarkable feature of this model is that the resulting index of refraction is a function of just height (one dimension), a fact that enables us to do several optimizations of the algorithm, as we will see later (see section 4.3).

3.1.2 BRDFs

BRDFs (see section 2.4.2) have been usually defined by intuitive models created by several different authors, attempting to simplify all the differential interactions into some mathematical expression that is easily configured. Probably the most simple of them is the *Lambertian BRDF*, which defines light to be scattered equally in all directions on the hemisphere, hence considering diffuse lighting only:

$$f_r(\vec{\omega}_i, \vec{\omega}_o) = \frac{k_d}{\pi} \quad (3.1)$$

where k_d is a coefficient that defines the ratio between incident and exitant flux (including micro-geometry self-shadowing and material absorption), and must be in the range [0..1] for energy conservation. As k_d depends on wavelength (because absorption depends on wavelength too, as seen on section 2.3), this coefficient is also giving information about the color of the material. Figure 3.1 shows the lambertian BRDF and an image rendered with it.

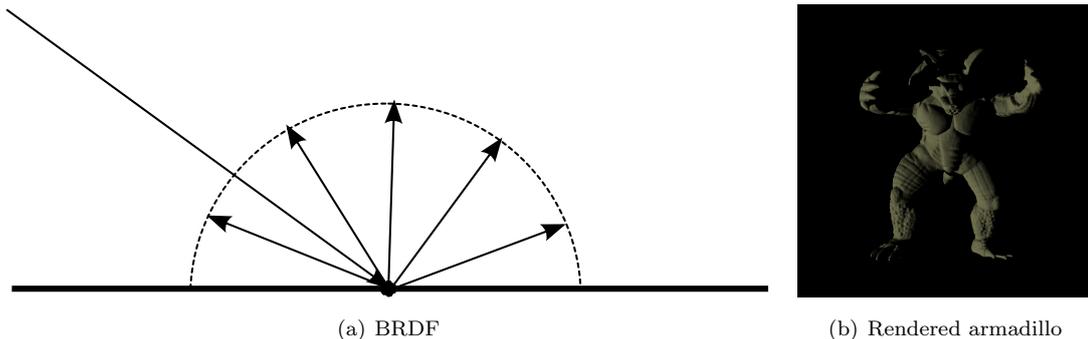


Figure 3.1: Lambertian BRDF. Notice the diffuse behavior of light.

However, the simplicity of the lambertian model is not able to model many phenomena, most of them very common. Not all the materials present such extreme micro-geometry rugosities, which

causes specular highlights to appear on the surface. Phong BRDF (Pho73) includes also a specular highlights. The physically-based Phong BRDF is defined as follows:

$$f_r(\vec{\omega}_i, \vec{\omega}_o) = \frac{k_d}{\pi} + k_s \frac{n+2}{2\pi} (\cos \alpha)^n \quad (3.2)$$

where k_s is the specular coefficient and must satisfy $k_d + k_s \leq 1$, α is the angle between $\vec{\omega}_o$ and the perfect specular reflection of $\vec{\omega}_i$ (according to the law of reflection, defined in section 2.2.3) and n is an exponent that defines the roughness of the surface. Sometimes, a more intuitive parameter b (angle covered by the specular highlight) is used instead. The relationship between n and b is:

$$n = -\frac{\ln 2}{\ln(\cos(b))} \quad (3.3)$$

Shadows get completely black if no global illumination is computed using the Phong BRDF. Sometimes, in order to emulate global illumination, a constant ambient lighting added is included into the Phong model. Figure 3.2 shows a BRDF and an object rendered using the BRDF.

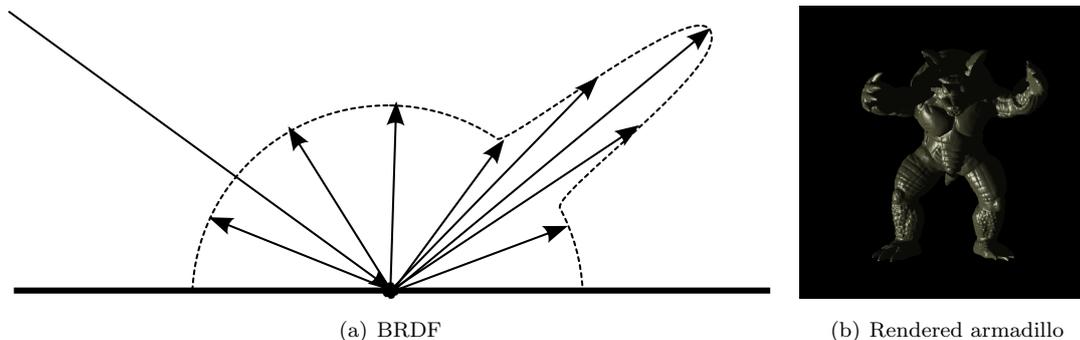


Figure 3.2: Phong BRDF. Notice the specular highlight and the ambient lighting.

There are many more BRDF models, some that pursue efficiency (Bli77), and some that model complex phenomena (War92b; CT81). Also, BTDF models can easily be deduce from BRDF models, just by considering the inverted hemisphere, and the refracted direction according to Snell’s law (see section 2.2.3) instead of the reflected one.

Measuring BRDFs is also possible. An obvious choice to measure general reflection properties is using a gonireflectometer (LFTW06). However, a complete characterization of a spectral, anisotropic BRDF may require up to 10^5 samples, so several optimization strategies have been introduced. By using a camera instead of a single photo-receptor, lots of samples can be obtained simultaneously (War92a). However, calibration issues need to be considered, which make measurements less precise.

In our work, we are interested in exploring the feasibility of appearance acquisition of complex materials by using genetic algorithms (see section 3.2).

Also, instead of measuring the BRDF, it is possible to define the micro-geometry and simulate the resulting BRDF using some kind of algorithm that samples that micro-geometry and considers different microscopic phenomena. Although this approach is not very common, given the effort of accurately modeling the corresponding micro-geometry, there are some works that follow it (WAT92; APS00), some of them even being able to simulate complex phenomena such as wave interference (GMN94) (see section 2.1.1.2).

3. PHYSICALLY - BASED MODELING

3.1.3 Participating media

There are several intuitive models that represent the phase functions of participating media. The simplest of them is analogous is the *isotropic* phase function, expressed as follows:

$$p(\vec{\omega}_i, \vec{\omega}_o) = \frac{1}{4\pi} \quad (3.4)$$

which satisfy the requirements of a physically-based phase function (see section 2.4.1). This phase function, actually, shows no directional dependency: light is uniformly distributed (see figure 3.3).

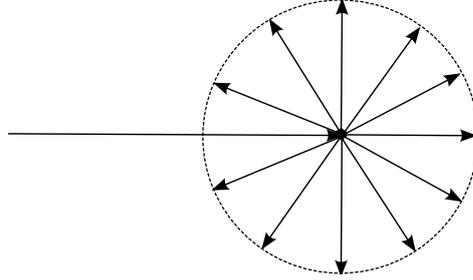


Figure 3.3: Isotropic phase function.

However, most of the times it is interesting to define the angular distribution of the radiance owed to a participating medium. Is light scattered backwards? Is it scattered mainly in forward directions? This behavior can be modeled by the Henyey-Greenstein phase function (HG41):

$$p(\vec{\omega}_i, \vec{\omega}_o) = \frac{1}{2(1 - 2g|\vec{\omega}_i \cdot \vec{\omega}_o| + g^2)^{\frac{3}{2}}} \quad (3.5)$$

where $g \in (-1, 1)$ is the *integral cosine* of the phase function. Values bigger than 0 mean forward scattering, while negative values mean backward scattering (see figure 3.4). The particular case of $g = 0$ is equivalent to the isotropic phase function.

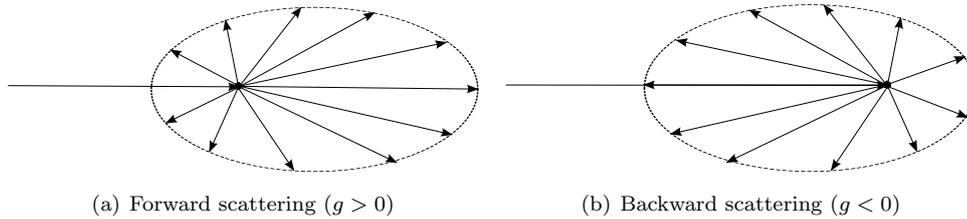


Figure 3.4: Henyey-Greenstein phase function

There are several other expressions for defining phase function models in different papers related to different knowledge fields such as astrophysics or oceanography (Mob94). Part of the research of this PhD (included in section 3.5) defines a bio-optical model of ocean water based on oceanographic studies and standard values, including absorption and scattering coefficients, plus the corresponding phase function definitions. Furthermore, it is the first full model of participating media that includes

(and simulates) wavelength redistribution functions that define inelastic scattering phenomena (see section 2.3.3).

Another possibility for defining the parameters of a participating media is by measuring them. Given its high dimensionality, its a difficult task. One of the most relevant breakthroughs on measuring coefficients for participating media works by diluting specific concentrations of different media on water and estimates the properties of the diluted material from a photograph of it (NGD⁺06). Another brilliant work is able to gather time-varying properties using a complex setup including a laser beam (HED05). Also, it is possible to gather non-homogeneous coefficients by using structured lighting (GNG⁺08).

The properties of participating media can also be simulated. Lorenz-Mie theory (Lor90; Mie08) derives a phase function definition (with the corresponding absorption and scattering coefficients for spherical particles on non-absorbing media, directly from Maxwell's equations (see section 2.1). These results have been used for simulating the properties of a medium composed by several different particle components (FCJ07).

Part of the research of this PhD (available on section 4.4) simulates a phase function given a particle of arbitrary shape, considering wave phenomena such as polarization and interference. This enables the simulation of rainbows, one of the most interesting phenomena owed to participating media.

3.1.3.1 Rainbows

Rainbows have traditionally been considered a fascinating topic, from scientists to philosophers, and are arguably one of the most beautiful displays of nature (Gre90; Min93; LL01). These interesting and beautiful spectral properties of rainbows are owed to the very special phase function of a medium made of water drops suspended on air.

Different theories have been developed over the centuries, and some of them have been adopted by the computer graphics community to simulate rainbows with varying degrees of realism. Some works provide a simplification of the process, in order to achieve interactive frame rates, while others present rainbow simulations in the context of atmospheric modeling. However, the complete (and quite complex) physics of rainbow formation has not been fully researched in the field of computer graphics.

Simple models include the work by Musgrave (Mus89), which follows Descartes' model from a classical geometrical optics perspective. Frisvad and colleagues presented a real-time simulation using Aristotle's rainbow formation theory based on reflections in clouds (FCF07). Although these models may provide intuitive explanations about rainbow formation, geometrical optics by itself fails to capture more complex aspects such as supernumerary arcs.

Lorenz-Mie theory (Lor90; Mie08) provides an exact solution for scattering by spherical particles in non-absorbing media. Given its computational complexity, it was not deemed useful until van de Hulst published results as tabulated data (van57). Unfortunately, this work was limited to very small spheres, and thus not suitable for all kinds of rainbows. This theory was later introduced to graphics by Rushmeier (Rus95). Furthermore, there's a huge on-line database related to research about rainbows by Philip Laven (Lavb), including Mieplot (Lava), a software able to simulate phase function of rainbows based on Mie theory.

Our work (see section 4.4) is free from the restriction of simulating spherical particles, and can include arbitrary geometries.

3. PHYSICALLY - BASED MODELING

3.1.4 BSSRDFs

Translucent materials pose a particularly hard problem because light can travel through and be reflected from a point inside the material re-emerging at a different location on the surface. Consequently a BSSRDF has a higher dimensionality compared to reflectance functions describing opaque surfaces making them very large. Hence analytical BSSRDF models are not commonly used.

One common way of defining a BSSRDF is by splitting its definition into a transparent surface with a participating medium on the inside, modeled with the corresponding coefficients and phase function (as in section 3.1.3). Although this approach constraints the general BSSRDF form, it also facilitates a more intuitive modeling of complex sub-surface properties.

A landmark method (JMLH01) is based on the observation that the appearance of translucent materials can be approximated with the concept of diffusion. In other words considering light as spreading away from the point at which it enters the surface. It approximates sub-surface scattering by a single scattering and a multiple scattering addends. Furthermore, within optically thick materials, the effects of single scattering in the final light transport are negligible (JB02). Light distribution can be considered isotropic. This allows us to express multiple sub-surface scattering as:

$$L_{ss}(\vec{r}_o, \vec{\omega}_o) = \frac{1}{\pi} F_t(\eta, \vec{\omega}_o) \int_A R_d(\|\vec{r}_o - \vec{r}_i\|) E(\vec{r}_i) dA(\vec{r}_i) \quad (3.6)$$

$F_t(\eta, \omega)$ is the Fresnel transmission coefficient ($\eta = n_t/n_i$ being the relative index of refraction) that has been defined on section 2.2.3. $R_d(\|x_{out} - x_{in}\|)$ is called the *diffuse reflectance function*, and depends on the distance between the incident and exitant points and the properties of the corresponding translucent material (eg. absorption coefficient, scattering coefficient, albedo or phase function). It is defined as follows:

$$R_d(r) = \frac{\Lambda'}{4\pi} \left[z_r \left(\sigma_{tr} + \frac{1}{d_r} \right) \frac{e^{-\sigma_{tr} d_r}}{\sigma'_t d_r^2} + z_v \left(\sigma_{tr} + \frac{1}{d_v} \right) \frac{e^{-\sigma_{tr} d_v}}{\sigma'_t d_v^2} \right] \quad (3.7)$$

where

- $\sigma_t = \sigma_a + \sigma_s$ is the extinction coefficient.
- $\sigma'_s = (1 - g)\sigma_s$ is the reduced scattering coefficient.
- $\sigma'_t = \sigma_a + \sigma'_s$ is the reduced extinction coefficient.
- $\Lambda' = \frac{\sigma'_s}{\sigma'_t}$ is the reduced albedo.
- $\sigma_{tr} = \sqrt{3\sigma_a\sigma'_t}$ is the effective extinction coefficient.
- $F_{dr} = -1.440\eta^{-2} + 0.710\eta^{-1} + 0.668 + 0.0636\eta$ is the diffusion fresnel term.
- $A = \frac{1+F_{dr}}{1-F_{dr}}$.
- $z_r = 1/\sigma'_t$ and $z_v = 1 + \frac{4}{3}A$.
- $d_r = \sqrt{r^2 + z_r^2}$ and $d_v = \sqrt{r^2 + z_v^2}$.

$E(x_{in})$ is the irradiance at a given point on the surface, expressed as:

$$E(\vec{r}_i) = \int_{\Omega} F_t(\eta, \vec{\omega}_i) L(\vec{r}_i, \vec{\omega}_i) (\vec{n} \cdot \vec{\omega}_i) d\vec{\omega}_i \quad (3.8)$$

Equations 3.9 and 3.8 define a sub-surface scattering behavior, as it can be realized from their similarity to equation 2.33. Note that, since we are assuming optically thick materials, single sub-surface scattering can be neglected and thus L_{ss} defines all the outgoing radiance.

Many research projects are based on this dipole approximation, either by extending it to multi-layered materials (DJ05; dLE07) or by exploiting the properties of this approximation for efficient translucency rendering (dLE07; CLH+08; WZT+08).

Capturing BSSRDFs for real-world translucent materials generally requires the use of relatively complex measuring equipment (MPZ+02; DHT+00; GLL+04; WMP+06). Camera-projector setups have also been used to measure reflectance of small material samples (PvBM+06; TGL+06). More recent image-based approaches attempt to capture bssrdf models using more practical camera equipment (DWD+08; GHP+08; WZT+08). The first of these uses a single multi-spectral image (DWD+08) to measure skin reflectance, but this approach imposes stringent constraints on the capture setup and requires samples to be taped in front of the equipment. The second approach exploits cross-polarization photography and uses 20 photographs from a single viewpoint to acquire a layered reflectance model of skin (GHP+08). The third is similar in spirit to our approach and estimates sub-surface scattering parameters from a single image under uncontrolled illumination, but the technique requires that the 3D location of the camera and geometry of the target is known a-priori (MSY08). The final example in this class of approaches requires sampled material to be a unit cube, and constrains the position of the camera and light source (WZT+08).

An alternative model-less approach separates the sub-surface scattering component of a BSSRDF, starting from a single image together with a set of diffuse priors (WT06). Since no specific reflectance model parameters are estimated, Wu and Tang’s (WT06) approach is applicable only where re-lighting is unnecessary, and insignificant specularities and reflections in the input image exist.

Part of the research of this PhD is related to the acquisition of translucent materials by using generic algorithms (see section 3.2) from a single render. Genetic algorithms have been successfully used before in the field of computer graphics for texture synthesis, analysis and parameterization (Sim91; STW99; QY02), image-based simulation of facial aging (HCG08), image recognition (KT94; KA06), or extraction of geometric primitives (RL94). However, genetic algorithms are not practical because they require several full renders per iteration, thus converging slowly even with fast rendering methods, and require known lighting and known camera position.

In summary the current state of the art methods to capture or approximate a BSSRDF from a single image impose specific restrictions on either the material sample, lighting conditions, camera position and/or require additional scene specific information. While this would be a daunting task and a very underconstrained and ill-posed problem, it is possible to attempt to approximate visually plausible BSSRDFs instead of physically correct ones. Recent image-based techniques that aim to achieve visually plausible results (KRFB06; GWJ+08) do not capture a full reflectance model. Interestingly however, these methods show that certain kinds of errors in estimated geometry, can still give rise to models that can be plausibly relit with entirely novel lighting; this supports our argument that approximate geometry and lighting estimates can lead to visually plausible results.

Another contribution of this PhD is a method to approximate plausible physically-based BSSRDFs, from single HDR pictures, without any prior information about geometry nor lighting. Details can be found on section 3.3.

3.2 Inverse rendering using genetic algorithms

Our first approach for estimating translucency from images is through minimization of the difference between the target image and a render using the specified parameter models. The chosen optimization method is genetic algorithms. Although this approach is slow due to the need of rendering the whole image for every parameter evaluation, we prove that it serves adequately for inverse rendering. Find the description of this algorithm below in the text and the results are on the *Results* chapter (5), section 5.1. This research has resulted into two publications (MMT⁺09; MMTG09).

3.2.1 Brief introduction about genetic algorithms

Genetic algorithms are probabilistic heuristic algorithms for search and optimization which apply the concepts of biological evolution: natural selection, reproduction and mutation. As any other optimization method, the algorithm tries to find a set of variables, (x_1, x_2, \dots, x_n) , so that the objective function, $F(x_1, x_2, \dots, x_n)$, reaches its maximum (or minimum). This section gives an overview of how these algorithms work, but we refer the reader to Winter and colleagues' work (WPGC95) for a more comprehensive explanation on genetic algorithms and their application.

Given that each possible set of input variables (x_1, x_2, \dots, x_n) is equivalent to a chromosome (and to an individual) and each parameter x_i is denominated *gene_i*, the analogy with the theory of evolution is immediate: starting from a population of n chromosomes, each of them delivers a solution to the problem, and only the chromosomes yielding the better solutions survive to produce the next generations and perpetuate their *genetic material*. Genetic diversity is completed by sexual reproduction and random mutations.

The algorithm consists of four steps: initialization, selection, reproduction and termination. Selection and reproduction are iterated until the condition for termination is reached.

Initialization. The first step implies the creation of an initial population of individuals (or sets of variables corresponding to the parameters we want to estimate). The genes of these individuals are generated randomly within the search space. Both the size of the initial population and the limits of the search space are input parameters to the algorithm and have a great influence in its performance, as analyzed in Section 3.2.2.3.

Selection. In order to apply the principle of natural selection, it is necessary to evaluate the performance of each generated individual. To do this, each individual is assigned a rating, called *fitness*, representing the proximity of that individual to the solution. Chromosomes are then ordered according to their *fitness* and the ones with the lowest *fitness* values are eliminated and substituted by the descendants of the surviving chromosomes (*based-on-rank selection*). This way only the genetic material delivering the best results is perpetuated.

Reproduction. This step entails the creation of the next generation using two genetic operators: *crossover* and *mutation* (see Figure 3.5). Crossover is a genetic operator used for exchange of genetic material, in which two chromosomes are randomly selected and an exchange of genes between them is performed. Mutation, on the other hand, ensures genetic diversity from one generation of individuals to the next by randomly modifying the value of some genes.

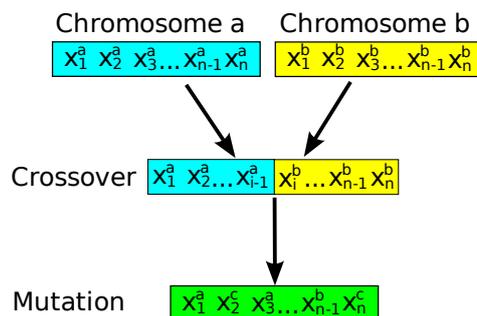


Figure 3.5: Crossover and mutation.

Termination. Typical termination conditions of the iterative process are: a solution is found which satisfies a certain minimum criterion, the specified maximum number of generations has been reached or the solution found cannot be improved any further.

In the following section, we present our adaptation of the genetic algorithms approach to the problem of appearance acquisition, and comment on some implementation details.

3.2.2 Appearance acquisition

3.2.2.1 Variables and objective function

To be able to run genetic algorithms for appearance acquisition, we first need to define the variables and the objective function. Our method works both for opaque and translucent materials, and the objective function is an error function, defined in both cases as the difference between the input image and the image rendered in each iteration with the estimated parameters. However, the reflectance parameters being sought, i.e., the variables of our optimization problem, differ in each case. In order to reduce the dimensionality of the problem, we assume that other parameters such as the lighting or the geometry of the scene are known. It should be noted that whilst the parameters are calculated in RGB, the error function, i.e. the difference between the original and the successively rendered images, is computed in the YCbCr color space, which is a perceptual color space (Poy96). This yields better results than comparing the images in non-perceptual color spaces such as RGB.

The two models chosen for this paper have been the Phong BRDF (see section 3.1.2) for opaque materials and the dipole model (see section 3.1.4) for translucent materials. Of course other models could be chosen and trivially adapted to our approach. The parameter space to be explored for the Phong model consists on five parameters to be optimized:

- k_d - diffuse coefficient, which is a RGB vector (three parameters).
- k_s - specular coefficient, which is a scalar (we consider that the specular coefficient does not present spectral dependence)
- b - specular angle, which is a scalar (no spectral dependence for this angle)

The dipole model consists on a set of eight parameters to be optimized:

3. PHYSICALLY - BASED MODELING

- σ_a - absorption coefficient, which is a RGB vector.
- σ_s - scattering coefficient, which is a RGB vector.
- g - phase function cosine integral, which is a scalar.
- η - relative index of refraction, which is a scalar.
- b - specular angle, which is a scalar (helps to model specular highlight on the surface)

All the scalars in previous parameter sets can easily be extended to RGB vectors and included into the algorithm, increasing the parameter space. However our tests show how the spectral dependency on those did not add that much potential and variability, while effectively reducing the convergence time of the algorithm.

3.2.2.2 Implementation of the algorithm

We provide here some insight on how the genetic algorithms framework maps to our appearance acquisition problem. A discussion of the influence of the specific parameters is provided in subsection 3.2.2.3.

The first step of any genetic algorithm is *initialization*. A set of chromosomes consisting of strings of reflectance parameters (eight in the case of translucent and four in the case of opaque materials, as explained before) are set. In this first generation the parameters could take random values within the search space, but, in order to accelerate convergence, we fixed the initial estimations of the different parameters to common values (that experimentally work well), shown in table 3.1. The number of chromosomes created is a configuration parameter of the algorithm.

Phong model			Diffuse approximation model				
$k_{d,(R,G,B)}$	k_s	b	η	g	b	$\sigma_{s,(R,G,B)}$	$\sigma_{a,(R,G,B)}$
0.5	0.5	10	1.3	0	10	1	1

Table 3.1: Initialization values of the sought parameters

An image is then rendered for each of the chromosomes created in each generation to calculate the *fitness* value of each chromosome and thus perform the *selection* step. This *fitness* value is calculated with a per-pixel least squares function measuring the difference between the individual channels in the original and the rendered images¹. The set of parameters delivering the most approximate solution are used to create the next generation. The number of chromosomes being replaced conforms another configuration parameter of the algorithm.

Once the best chromosomes have been selected, *reproduction*, involving crossover and mutation, takes place. In our implementation crossover is performed at only one point of the chromosome, which has proven enough for our objectives, but more complex crossover procedures are also possible. During mutation gene values vary between $\pm 0-30\%$ of their original value. Our research shows that greater variations introduce a too random behavior and control over the evolution of the algorithm is easily lost, whereas very small variations need many generations for the algorithm to reach a valid solution.

The processes of selection and reproduction continue iteratively until the termination condition is met. Given that our goal is to study the effectiveness of the algorithm and the influence of its configuration parameters on the final result, we simply define our termination condition as a fixed number of generations. This suffices in our context, although changing the termination condition to an error threshold is straightforward.

¹Our experiments showed no significant difference working on RGB or YCrCb color spaces; the images shown in the paper have been obtained in RGB space.

3.2.2.3 Parameter space

Genetic algorithms have a series of input parameters (initial number of individuals, crossover and mutation probabilities, etc), which affect the final result. In order to select the most adequate values for these parameters, we have performed a series of tests, taking into account both the accuracy of the final solution and the computation time required. The results of these tests for the most relevant configuration parameters are discussed here. The accuracy was measured as percentage of error between the real ground truth values and the values obtained by the algorithm.

Probability of replacement. The probability of replacement accounts for the percentage of individuals which are eliminated in each selection process. Following the evolution simile, the higher this probability is, the faster the population evolves. However, running times also increase significantly, as all the chromosomes and their corresponding image need to be created for each generation. As seen on Figure 3.6 (top) it is one of the most influential parameters of the algorithm, both in time and in accuracy of the result.

Probability of crossover. As explained in Section 3.2.1, crossover represents sexual reproduction and takes place after the selection and replacement process. The probability of crossover represents the percentage of individuals which are the result of combining the genes of two survivor chromosomes. The second row of Figure 3.6 shows how an increase in *sexual reproduction* (hence favoring genetic diversity) causes the percentage of error to decrease slightly. As the resulting images of more combinations of genes need to be calculated, execution time increases slowly.

Probability of mutation. Representing the percentage of genes which mutate from one generation to the next, this probability is critical when working with a small number of individuals per generation. Variations in the initial genes are crucial to progressively reach the optimal solution, and the higher this probability, the lower the percentage of error with minimum time penalty (see Figure 3.6, third row).

Number of generations. The number of generations is undoubtedly, and together with the number of individuals per generation discussed below, the parameter with the greatest influence. It indicates the number of generations which are created before the algorithm terminates and delivers a solution (alternatively, an error threshold can be trivially set as termination parameter). Figure 5.2 shows how the solution progressively evolves along generations. With an infinite number of generations, the solution would perfectly match the original. In practice, a compromise has to be found between execution time and accuracy of the solution, determined by the number of generations.

Number of individuals per generation. The effect of the number of individuals of each generation in the performance of the algorithm is straightforward: the more individuals, the least the percentage of error, as more possibilities are evaluated. However, there is a substantial increase in the execution time, as shown in Figure 3.6 (last row).

This approach for inverse rendering has been published in two international conference proceedings (MMT⁺09; MMTG09).

3. PHYSICALLY - BASED MODELING

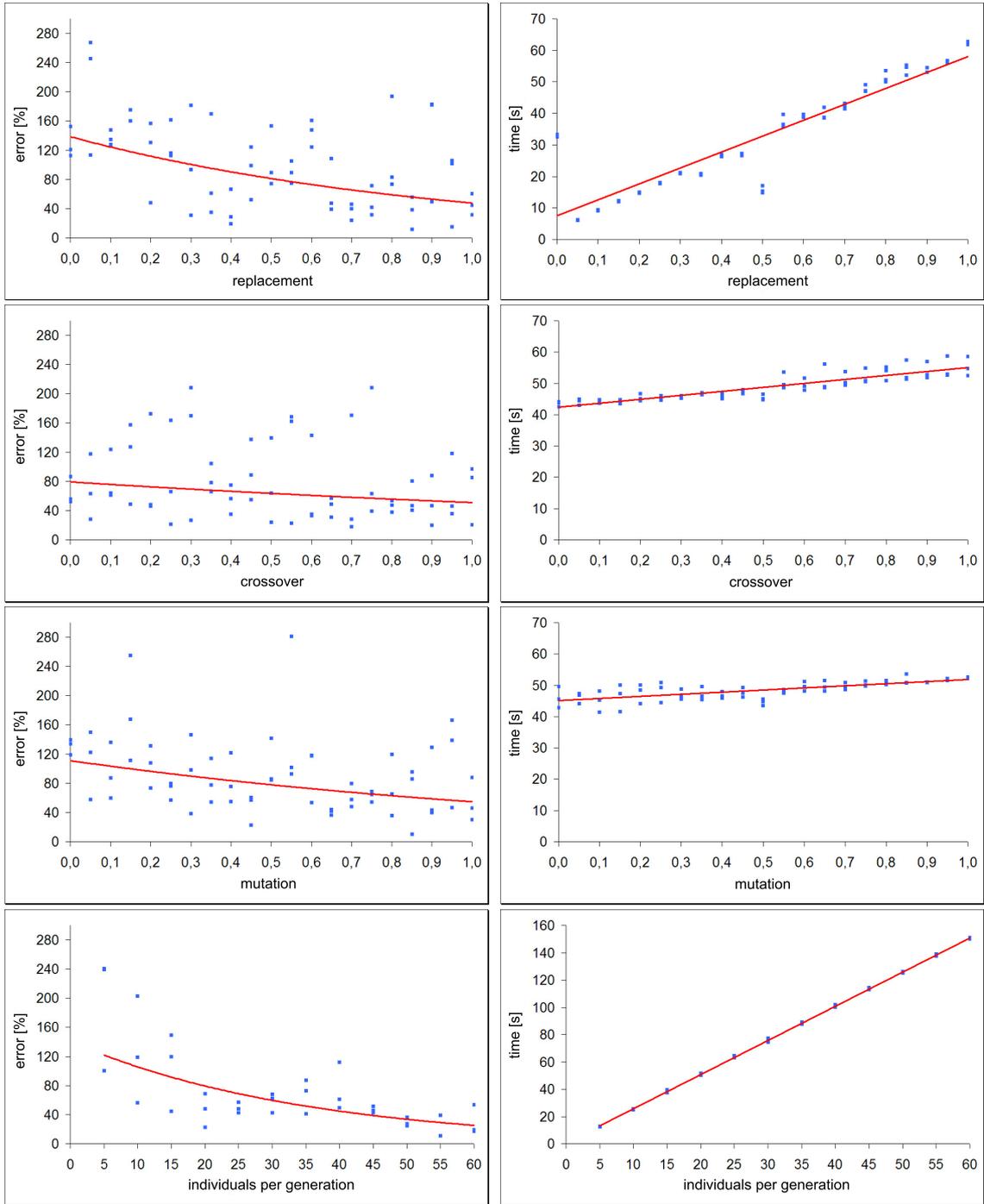


Figure 3.6: Percentage of error (*left column*) and execution time (*right column*) as a function of, *from top to bottom*, the probability of replacement, the probability of crossover, the probability of mutation and the number of individuals in each generation. Data obtained for the *Phong* model.

3.3 Estimating BSSRDFs from single HDR photographs

We aim to obtain a model of the BSSRDF of a translucent object from a single image, without any prior information. Assuming that the object is made up of an optically thick, homogeneous material, we can neglect the effects of single scattering. Our BSSRDF estimation is based on the diffusion approximation (JMLH01); the diffusion profile is expressed as a linear combination of piecewise constant functions, resulting into a linear system that can be efficiently solved, applying the Quasi-Minimal Residual method (BBC+94). The obtained profile is subsequently smoothed by fitting it to a dipole model.

In this section we introduce our approach for controlled environments, where both the geometry and the main light direction in the scene are known. This allows us to demonstrate the validity of our BSSRDF approximation algorithm. We then generalize our algorithm in the next section for the ill-posed case of single images, showing how to leverage rough estimates of both shape and light direction.

3.3.1 Algorithm

Given an alpha matte \mathbb{O} of the image object, we first discard pixels representing highlights by simply assuming that the minimum of the histogram’s derivative indicates the start of the highlight (KRFB06). This defines $\mathbb{I} \subseteq \mathbb{O}$ as the set of object pixels containing useful subsurface light transport information. Inspired by a recent texture-editing technique (FH04), we subsequently minimize the effect of indirect lighting by finding the pixel in \mathbb{O} with the lowest luminance and subtracting that value from the pixels in \mathbb{I} . These simple operations help increase the accuracy of the final result.

Our BSSRDF estimation process leverages the fact that within optically thick materials, single scattering effects are negligible (JB02). Light distribution can be considered isotropic and thus we can expect the dipole diffusion approximation to hold (JMLH01). This allows us to express multiple subsurface scattering as:

$$L(x_{out}, \omega_{out}) = \frac{1}{\pi} F_t(\eta, \omega_{out}) \int_A R_d(\|x_{out} - x_{in}\|) E(x_{in}) dA(x_{in}) \quad (3.9)$$

where $L(x_{out}, \omega_{out})$ refers to the outgoing radiance at a specific point x_{out} in a specific direction ω_{out} , $F_t(\eta, \omega)$ is the Fresnel transmission coefficient (η being the relative index of refraction). $R_d(\|x_{out} - x_{in}\|)$ is called the *diffuse reflectance function* (see section 3.1.4), and depends on the distance between the incident and outgoing points and the properties of the corresponding translucent material (e.g. absorption coefficient, scattering coefficient, albedo or phase function).

Given that we have roughly eliminated highlights and indirect illumination from the object matte, the outgoing radiance is mainly owed to subsurface scattering. The pixel values in \mathbb{I} are therefore taken as a good estimator for the radiance L in Equation 3.9.

The two terms in Equations 3.9 and 3.8 that define the properties of the translucent material are the index of refraction η and the diffuse reflectance function $R_d(\|x_{out} - x_{in}\|)$. We use a standard value of $\eta = 1.3$ (JMLH01; XGL+07; WZT+08). Consequently, the only unknown in our model is $R_d(\|x_{out} - x_{in}\|)$. Different formulations for this function have been previously proposed. Note that, although we use the dipole approximation for demonstration purposes, our method is independent of the specific definition of this function. By assuming directional light sources (an approach that

3. PHYSICALLY - BASED MODELING

has been exploited in related work (WZT+08)), based on Equation 3.8, we can define the per-pixel irradiance maps (front E and back E_b) as:

$$E_i = \sum_k F_t(\eta, \omega_k) L_k(p_i, \omega_k) (n_i \cdot \omega_k) \quad (3.10)$$

$$E_{b,i} = \sum_k F_t(\eta, \omega_k) L_k(p_{b,i}, \omega_k) (n_{b,i} \cdot \omega_k) \quad (3.11)$$

where $i \in \mathbb{O}$ represents the pixel index, ω_k is the direction of each light, p_i represents a point on the object's 3D surface and $p_{b,i}$ is the corresponding point along the Z direction at the back of the object (see Figure 3.7). n_i and $n_{b,i}$ are the respective normals at those points. The irradiance maps are defined per color channel in RGB space, and our algorithm is applied to each channel independently.

For an orthogonal projection, the view vector c for each point p is $c = (0, 0, 1)$ (a perspective projection could be used instead, yielding different per-pixel values of c). Considering $\omega_{out} = c$ in Equation 3.9, this yields $L_i = L(p_i, c)$ for each pixel in \mathbb{I} . Therefore we can now express Equation 3.9 in terms of depth, surface normals, camera and irradiance maps as follows:

$$L_i = \frac{1}{\pi} F_t(\eta, c) \sum_{j \in \mathbb{O}} (R_d(r) E_j \Delta A + R_d(r_b) E_{b,j} \Delta A_b) \quad (3.12)$$

where L_i represents the color of a pixel, $\Delta A = |c \cdot n_i|^{-1}$ is a factor related to the screen-space projection of the area of the object in a single pixel (similarly for ΔA_b), and r and r_b represent Euclidean distances in 3D space from point p_i on the front of the object to points p_j and $p_{b,j}$ (see Figure 3.7).

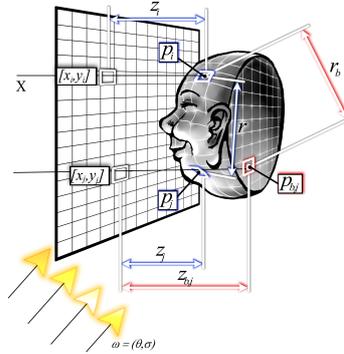


Figure 3.7: Definition of distances r and r_b .

Approximating the diffuse reflectance function: The only unknown in Equation 3.12 is the diffuse reflectance function R_d , which defines the properties of a translucent material. To solve this equation, standard minimization-optimization algorithms could be used. However such algorithms would be very time consuming, would require defining a specific model for the R_d function and might not converge to a plausible solution. Therefore, we opt for a more efficient method, and approximate R_d by a linear combination of a set of basis functions. This linear combination enables us to apply Equation 3.12 for each pixel $i \in \mathbb{I}$. We first rewrite Equation 3.12 as:

3.3 Estimating BSSRDFs from single HDR photographs

	Piecewise constant			Haar wavelets			Zero-mean gaussian			Hermite polynomials			Legendre polynomials		
Number of functions	10	20	30	10	20	30	10	20	30	10	20	30	10	20	30
Estimation time	28 s	30 s	38 s	67 s	97 s	122 s	189 s	273 s	388 s	49 s	139 s	480 s	82 s	297 s	743 s
Condition number	$1.5 \cdot 10^4$	$1.1 \cdot 10^4$	$1.3 \cdot 10^5$	$4.0 \cdot 10^5$	$4.1 \cdot 10^5$	$4.3 \cdot 10^5$	$2.9 \cdot 10^8$	$8.3 \cdot 10^8$	$2.7 \cdot 10^9$	$1.8 \cdot 10^{12}$	$2.0 \cdot 10^{13}$	-	$4.7 \cdot 10^8$	$1.3 \cdot 10^9$	$6.4 \cdot 10^9$
Error	2.98%	2.51%	2.43%	2.98%	2.66%	2.51%	2.97%	3.19%	5.97%	3.22%	18.6%	-	2.73%	2.5%	2.51%

Table 3.2: Results of our basis functions tests for the whole milk material. For an increasing number of basis functions, the table shows estimation time, condition number of the matrices and average relative error of the result. With 30 Hermite polynomials the system does not converge.

$$L_i = \sum_{j \in \mathbb{I}} (K_j R_d(r) + K_{b,j} R_d(r_b)) \quad (3.13)$$

where $K_j = \pi^{-1} F_t(\eta, c) E_j \Delta A$ (with a similar definition for $K_{b,j}$). Next, we estimate R_d by a linear combination of m basis functions:

$$R_d(r) \approx \sum_{h=1}^m \hat{w}_h B_h(r) \quad (3.14)$$

where $B_h(r)$ represents the basis functions (discussed at the end of this section) and \hat{w}_h are the weights assigned to each basis function. Equation 3.13 now yields:

$$L_i = \sum_{j \in \mathbb{I}} \left(K_j \sum_{h=1}^m \hat{w}_h B_h(r) + K_{b,j} \sum_{h=1}^m \hat{w}_h B_h(r_b) \right) \quad (3.15)$$

This equation applies to every pixel $i \in \mathbb{I}$. However, we have found that scaling \mathbb{I} down to a resolution of around 200x200 (preserving the aspect ratio of the input image) yields valid approximations for R_d while greatly reducing computation times. Applying the equation to each pixel of the scaled \mathbb{I} we get a linear system defined by the matrix product $\mathbf{A} \cdot \mathbf{X} = \mathbf{B}$, for n pixels and m basis functions, with:

$$a_{ih} = \sum_{j \in \mathbb{O}} (K_j B_h(r) + K_{b,j} B_h(r_b)) \quad (3.16)$$

$$\mathbf{X}_{m \times 1}^T = (\hat{w}_1 \quad \hat{w}_2 \quad \dots \quad \hat{w}_m) \quad (3.17)$$

$$\mathbf{B}_{n \times 1}^T = (L_1 \quad L_2 \quad \dots \quad L_n) \quad (3.18)$$

Resolution method: To solve the equivalent system $(\mathbf{A}^T \mathbf{A}) \mathbf{X} = (\mathbf{A}^T \mathbf{B})$ we note that some columns in \mathbf{A} may contain values close to zero. This leads to a highly ill-conditioned matrix, while the related basis functions have negligible influence in the final solution. We thus set the associated weights \hat{w}_h to 0 and remove the corresponding columns from \mathbf{A} . Although this approximation reduces the condition number, the system is still ill-conditioned; we improve it further by using a Jacobi pre-conditioner for $(\mathbf{A}^T \mathbf{A})$, and solve the system using the Quasi-Minimal Residual (QMR) method (BBC⁺94).

3. PHYSICALLY - BASED MODELING

Basis functions: To choose an appropriate set of basis functions, we rendered translucent objects using materials defined in (JMLH01), and approximated a diffusion profile testing different options: piecewise constant functions, zero-mean Gaussians (inspired by the work of d'Eon et al (dLE07)), Hermite and Legendre polynomials and Haar wavelets. An alternative might have been to try piecewise polynomials (XGL⁺07) as well. We choose not to do it due to the strict continuity conditions imposed on the linear system.

Zero-mean gaussian functions show extremely high condition numbers, thus yielding higher errors. Hermite polynomials also result into high errors (with 30 basis functions the QMR iterative process does not even converge). Legendre polynomials, Haar wavelets and piecewise constant functions present similar behavior in terms of error. We choose piecewise constant functions given their lower condition numbers and estimation times. A good compromise between detail in the estimation and system stability is reached by using between 20 and 30 piecewise constant basis functions. Table 3.2 shows the results for the case of the *whole milk* material from (JMLH01).

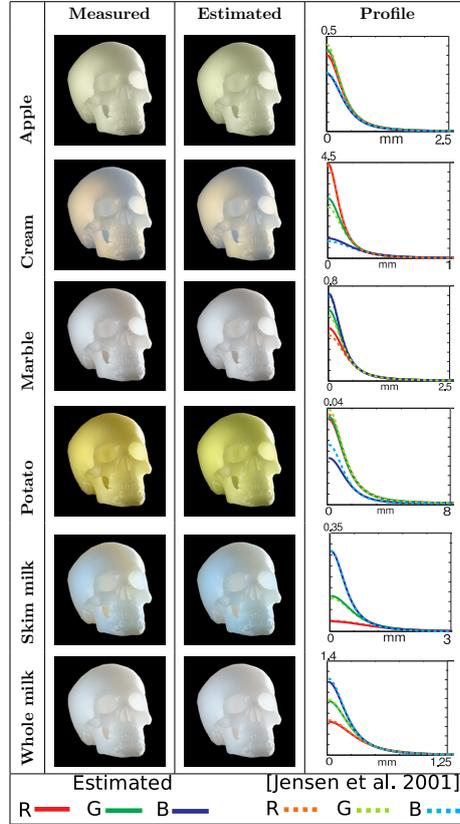


Figure 3.8: Left two columns: Comparison between renders using materials from [Jensen et al. 2001] and our estimated diffusion profiles. Right column: Comparison of diffusion profiles.

Fitting: The piecewise constant result is then fitted to a continuous, differentiable, monotonically decreasing function. This helps eliminate noise and avoid discontinuities in the rendering, while keeping the function physically plausible. Although our algorithm imposes no particular model for this function, we use the dipole approximation given its ease of use and its wide acceptance in the graphics community. This has the additional advantage of allowing us to compare our results with physical measures of real materials from (JMLH01), to demonstrate the validity of our approach.

Validation: To validate our BSSRDF estimation algorithm under known geometry and lighting, we rendered objects with different material parameters from (JMLH01). We then used the resulting images as input to our algorithm. The estimated diffusion profiles are fitted to the dipole approximation. Both the profiles and the overall look of the images rendered with them are very similar to the ground truth (see Figure 3.8). The differences are due to the coarse modeling of the R_d function by a limited number of basis functions, given the intrinsic trade-off between this number and the conditioning of the linear system.

Given that we use the dipole approximation, we can also derive albedo and extinction coefficients for the estimated materials. Table 3.3 shows how our results match physically measured data from (JMLH01), with very small amounts of residual error for most materials.

	Reduced albedo									Reduced extinction (mm^{-1})								
	(JMLH01)			Estimated			Error			(JMLH01)			Estimated			Error		
	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
Apple	0.9987	0.9986	0.9772	0.9969	0.9985	0.9686	0.18%	0.01%	0.88%	2.2930	2.3934	2.0160	2.2428	2.3216	2.0202	2.19%	3.00%	0.21%
Cream	1.0000	0.9995	0.9949	1.0000	1.0000	0.9967	0.00%	0.05%	0.18%	7.3802	5.4728	3.1663	7.4580	5.9233	3.4267	1.05%	8.23%	8.22%
Marble	0.9990	0.9984	0.9976	1.0000	1.0000	1.0000	0.10%	0.16%	0.24%	2.1921	2.6241	3.0071	2.3543	2.7351	3.0359	7.40%	4.23%	0.96%
Potato	0.9965	0.9873	0.8209	1.0000	0.9999	0.9145	0.35%	1.27%	11.40%	0.6824	0.7090	0.6700	0.6690	0.6806	0.5651	1.97%	4.00%	15.65%
Skim milk	0.9980	0.9980	0.9926	0.9898	1.0000	0.9981	0.82%	0.20%	0.56%	0.7014	1.2225	1.9142	0.6875	1.2602	1.8943	1.99%	3.08%	1.04%
Whole milk	0.9996	0.9993	0.9963	1.0000	1.0000	0.9818	0.04%	0.07%	1.46%	2.5511	3.2124	3.7840	2.4968	3.1725	3.7553	2.13%	1.24%	0.76%

Table 3.3: Comparison between the measured properties of several materials (JMLH01) and the estimated properties resulting from our method.

3.4 Generalizing to Single Images

Once we have demonstrated the suitability of our method in controlled environments, we wish to extend our approach to a much more challenging scenario: approximating diffusion profiles from single images. This is a heavily ill-posed problem, given that neither the light direction nor the geometry are known in this case.

We follow the same approach of the BSSRDF hallucination technique of the SubEdit system (STPP09): instead of trying to recover an exact physically-based BSSRDF (which is obviously impossible), we aim to estimate a plausible representation that will yield results similar to the material in the input image. We do not fit a dipole diffusion profile, but relax our fitting to a function defined by a set of piecewise cubic Hermite polynomials instead. We only impose the restriction of being differentiable and monotonically decreasing, to ensure physical plausibility, but eliminate the constraints on its second derivative. This degree of freedom allows us to obtain solutions that better represent the subtle heterogeneities and irregularities of real-world materials, which translates into more realistic renderings.

We leverage recent findings by Fleming and colleagues (FB05), who conclude that humans do not estimate translucency through accurate inverse optics, but instead perceive the overall look of translucent materials based on simple image heuristics. This suggests that a suitable *approximation* of both the object’s shape and incident light direction may suffice for our purposes, an approach that has been recently employed in the context of material editing (KRFB06) or the simulation of caustics (GSLM+08). We therefore adapt existing techniques, a process we summarize in the following paragraphs.

Estimating shape: Estimating shape from an image of an opaque object is a very under-constrained problem by itself. We note that this estimation is even harder if the object is translucent, given the softening effects of subsurface scattering. We leverage three sources of information: pixels in

3. PHYSICALLY - BASED MODELING

the contour (which we assume to lie on the image plane at $Z=0$), shading information across its surface, and our additional assumption of global convexity. Inspired by previous approaches (KRFB06; Joh02), we reconstruct the depth map Z of an object as the weighted sum of a base layer (which encodes global convexity) and a detail layer (which encodes high frequency), both obtained by means of the bilateral filter. We rely on additional non-linear spline functions to reshape the base layer and boost its apparent "inflation", similar to (KRFB06). Given the bass-relief ambiguity (BKY99; Kv01), we reverse the resulting signal if necessary to comply with our global convexity assumption, which yields our final depth map Z . A normal map N is subsequently computed from Z . Additionally, a *back depth map* Z_b plus the corresponding *back normal map* N_b are generated; we make the simplifying assumption that the back of the object can be approximated by mirroring Z . While this is a strong simplification to circumvent the fact that we do not have information about the back portion of the object in the image, this straightforward operation suffices to produce good results. We use values of $\sigma_{spatial} \in [0.08..0.1]$ and $\sigma_{intensity} \in [0.3..0.5]$ for the bilateral filter, while the weights for adding the base and detail layers are usually 0.8 and 0.2 respectively (thus favoring our assumption of global convexity over details).

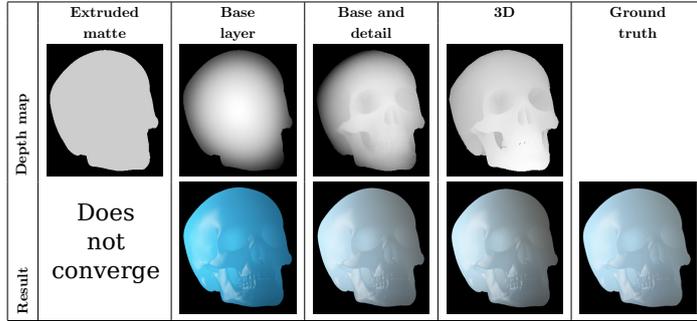


Figure 3.9: Different depth estimation techniques, and the corresponding results when used in our method. The simplest method does not converge, whereas using only a base layer may lead to unsatisfactory results. A good balance between visual accuracy and simplicity is better achieved with the combination of base and detail layers.

Some recent works rely on user interaction to help find a good estimate of normals from a single image, such as the work by Wu et al. (WSTS08). While their technique could certainly be employed here, our approach works sufficiently well in our context. On the other hand, it could be argued that even a simpler approach could be used instead, but, as Figure 3.9 demonstrates, that may lead to wrong results in the final subsurface light transport. We nevertheless restrict our estimations to simple geometries in order to minimize the impact of this error on the BSSRDF estimation, leaving the field of depth estimation of complex translucent geometries still open for further research. Future more accurate techniques could be trivially included at this stage.

Estimating light direction: Several existing methods can estimate light source directions from a single image, but usually at the expense of assuming some previous knowledge or including a calibration object in the scene (ZY01; WS02). In contrast, our goal is to obtain the dominant light direction starting with a single, off-the-shelf image, and thus we cannot impose such restrictions to our inputs.

We apply the method recently proposed by Lopez-Moreno et al. (LMHRG09), which performs a two-step analysis of an object’s luminance channel: first, the pixels of the contour \mathcal{O} are clustered by a k-means algorithm to identify the number of light sources in the scene, as well as their azimuth θ_i direction (in image-space) and relative intensities. Second, zenith angles ϕ_i are approximated for each light direction by analyzing gradients in the interior of the object. The pair (θ_i, ϕ_i) defines the recovered 3D direction for each light.

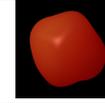
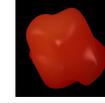
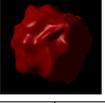
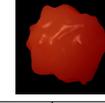
	Opaque		Translucent		Very translucent	
Geometry 1						
	θ	ϕ	θ	ϕ	θ	ϕ
	145.0°	52.1°	142.0°	51.0°	142.7°	50.5°
Error	0.0°	7.1°	-3.0°	6.0°	-2.3°	5.5°
Geometry 2						
	θ	ϕ	θ	ϕ	θ	ϕ
	152.5°	60.5°	149.9°	60.1°	156.2°	52.5°
Error	7.5°	15.5°	4.9°	15.1°	11.2°	7.5°
Geometry 3						
	θ	ϕ	θ	ϕ	θ	ϕ
	155.3°	61.3°	149.9°	60.1°	154.3°	59.8°
Error	10.3°	16.3°	4.9°	15.1°	9.3°	14.8°
Geometry 4						
	θ	ϕ	θ	ϕ	θ	ϕ
	159.2°	50.3°	149.0°	51.8°	153.4°	56.9°
Error	14.2°	5.3°	4.0°	6.8°	8.4°	11.9°

Figure 3.10: Performance of the light detection method for varying degrees of translucency and geometric complexity. The images have been rendered with a directional light source at $(\theta, \phi) = (145^\circ, 45^\circ)$. The detected light directions are shown below each image, along with the error. The error is always $\epsilon < 20^\circ$, which is below perceptual threshold (LMHRG09).

As in the case of the shape estimation algorithm, the original light detection algorithm was designed for opaque objects, where it was proved by the authors to yield satisfactory results. In order to assess how well its extension to translucent objects works, we tested it in a controlled scene with incident light at $(\theta, \phi) = (145^\circ, 45^\circ)$ over an object with varying degrees of translucency. As Figure 3.10 shows, the error of the algorithm even with translucent objects is always less than 20° , which was proven by the authors to be usually undetected by human observers.

To test how the total error in the light detection affects the BSSRDF estimation algorithm, we manually introduced light direction estimations $(\hat{\theta}, \hat{\phi})$, increasing errors in the angles, and compared our results with rendered images using the diffusion approximation. Figure 3.11 shows how the error on the estimated BSSRDF increases as the error of the light detection increases. Our experiments (and results shown in this work) confirm that satisfactory results are obtained with errors of up to 20° (see Figure 3.10).

Section 5.2 show the results of this approach for BSSRDF estimation.

3. PHYSICALLY - BASED MODELING

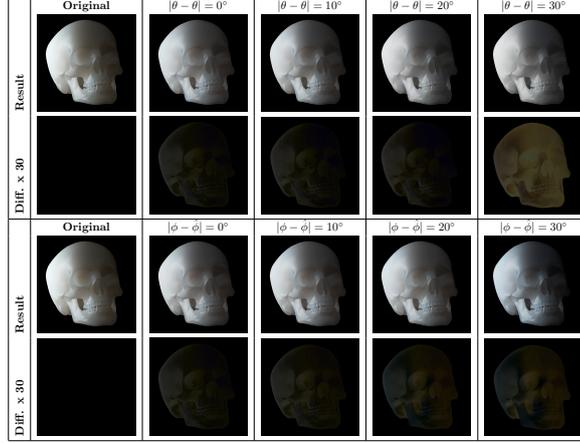


Figure 3.11: Behavior of the BSSRDF estimation algorithm according to the error on the light estimation (both on azimuth and zenith). The resulting renderings are visually accurate up to an error of 20° .

3.5 The ocean: building a model from measured data and standards

In previous sections the modeling of the corresponding media was done from inverse rendering or by approximating data from a photograph. However, in this section we follow a different approach. There is plenty of data for specific participating media that can be found on specialized literature or standardization consortiums, because it has been measured using very specific equipment. This is the case of ocean water. In this section a model of ocean water can be found. This model includes inelastic scattering (see section 2.3.3) that enables energy transfers between different wavelengths. Later in the text, section 4.2 presents a novel algorithm that is able to simulate this complex ocean water model, and section 5.3 shows the results and conclusions of applying the rendering algorithm to this bio-optical model. The result of this work has been published in the Computer Graphics Forum (GSMA08).

Apart from the water molecules themselves, natural water is made up of a high variety of optically influent constituents that are usually classified into *dissolved* (matter with a diameter less than $0.4 \mu\text{m}$) and *particulate* matter, which can be organic or inorganic. The presence and concentrations of these constituents determine the optical properties of water. These optical properties are divided in two classes: *inherent* and *apparent*. Together with the incident light, they determine the underwater light field. Other factors like the state of the water body (still, turbulent, etc.) or the bottom conditions may also alter this light field, but these are not included in our model.

Inherent optical properties (IOP) only depend on the constituents of the water. The two fundamental IOP are the absorption coefficient α and the volume scattering function β , or VSF, which characterizes the intensity of scattering as a function of angle. Other properties commonly derived from these include the scattering coefficient σ , calculated by integrating β over all angles, the extinction coefficient κ and the albedo Λ . Works on scattering often refer to the scattering phase function p , which is related to the VSF by $p = \beta/\sigma$. The *apparent optical properties* (AOP) are not properties of the aquatic medium itself, although they do depend on its characteristics. Some of the AOP include irradiance reflectance (defined as the ratio of the upward to the downward irradiance at a given point), attenuation coefficients or the average cosines (Pre76). Both IOP and AOP provide the basis of a widely used classification of water bodies: Case I and Case II waters (Mob94). Case I waters

3.5 The ocean: building a model from measured data and standards

are usually found in open sea, whereas Case II refers to coastal waters and tend to be murkier. Our bio-optical model covers both cases without the need for any special reformulation, and thus we will not make any further distinctions between the two in the paper.

Scattering in water is caused by interactions of light at molecular level and with organic and inorganic particles (Mob94). It can be classified in two broad categories: *elastic* or *inelastic* scattering, depending on whether the scattered photon maintains or changes its energy in the process. The inelastic scattering events can be further subclassified according to the nature of the energy transfer: *Stokes* scattering, when a molecule of the medium absorbs the photon and re-emits it with a lower energy (having transferred energy to the medium), and *anti-Stokes* scattering, when the re-emitted photon has a higher energy (absorbed from the medium). Both cases are also covered by our model. The process implies an energy transfer from wavelength λ' to λ , with λ' being the excitation wavelength and λ the re-emitted wavelength. The former case implies a shift towards longer wavelengths, whereas in the latter the scattered photon has a shorter wavelength. Major forms of elastic events in water include Einstein-Smoluchowski scattering (see Section 3.5.3), whereas for inelastic events, *Raman* scattering and *fluorescence* are the two most prominent and are described in Section 3.5.4.

3.5.1 The Bio-Optical Model

The various constituents of ocean water have a great influence in its optical properties. In order to solve the forward problem in ocean optics, the IOP have to be modeled and used in the FRTE. The values of these IOP can be obtained as the sum of the contributions of pure water and the dissolved particles and particulate matter present in the water, as proposed in (Mob94). Optically pure water is devoid of any dissolved or suspended matter, and thus there is no scattering or absorption owed to particles or organic material (Mor74). For *saline* pure water the salt concentration (35 to 39 parts per thousand) does influence the scattering and absorption functions. In particular it absorbs most wavelengths except for blue, with the absorption coefficient peaking at 760 nm, and reaching a minimum at 430 nm. The relevance of the wavelength λ for the properties of water starts to become clear. For the rest of this section we include the dependencies on λ for clarity purposes (while on section 2.4.1 it was omitted for brevity).

Table 3.4: Absorption coefficient for a clear water body $\sigma_{a,w}$ (after Smith and Baker (SB81)) and specific absorption coefficient for phytoplankton $\sigma_{a,p}^*$ (after Sathyendranath, Lazzara and Prieur (SLP87)).

$\sigma_{a,w}$	$[cm^{-1}]$	0.00022	0.000145	0.000257	0.000638	0.00289	0.0043	0.01169	0.0236
$\sigma_{a,p}^*$	$[m^2 \cdot mg^{-1}]$	0.025	0.035	0.02	0.01	0.007	0.015	0.001	0.0001
λ	$[nm]$	380	440	500	550	610	670	720	780

We develop our bio-optical model from three main IOP, with others like the extinction coefficient or the albedo derived from those three. These IOP are the absorption coefficient (3.19), the scattering coefficient (3.20) and the phase function (3.21), which for the elastic case can be written as (see Table 4.2 for a more detailed description of the functions used, including both the elastic and inelastic cases):

$$\sigma_a(\lambda) = \sigma_{a,w}(\lambda) + \sum_i \sigma_{a,i}(\lambda) \quad (3.19)$$

$$\sigma_s(\lambda) = \sigma_{s,w}(\lambda) + \sum_i \sigma_{s,i}(\lambda) \quad (3.20)$$

$$p(\lambda, \theta) = \frac{\sigma_{s,w}(\lambda)}{\sigma(\lambda)} p_w(\lambda, \theta) + \sum_i \frac{\sigma_{s,i}(\lambda)}{\sigma(\lambda)} p_i(\lambda, \theta) \quad (3.21)$$

3. PHYSICALLY - BASED MODELING

where θ is the angle between the incoming $\vec{\omega}_i$ and outgoing $\vec{\omega}_o$ directions, the subscript w stands for the contribution of the pure water (fresh or salty) and the subscript i stands for the constituents in the water body such as biological particles or dissolved substances. We include three types of such constituents in our model, namely CDOM (Colored Dissolved Organic Matter, also known as yellow matter, present mainly in shallow ocean waters and harbors), phytoplankton (microscopic plants rich in chlorophyll) and minerals and organic detritus. The rest of this section will characterize the three main IOP (with elastic and inelastic scattering treated separately) for pure water and the three constituents. The next section will show how radiative transfer theory is applied to simulate the light field (which define the AOP) and render the final images.

3.5.2 Modeling Absorption

For the spectral absorption function of pure water $\sigma_{a,w}(\lambda)$ we rely on the work of Smith and Baker (SB81), whose tabulated values are well known in oceanography studies (shown in Table 3.4). Following further studies by Pope and Fry (PF97), we use those values as an upper bound, to account for the fact that the true absorption can be, in fact, lower. The function shows that absorption is more prominent both in the UV and red ends of the spectrum. (PF97) also shows that absorption by salt in oceanic water is negligible.

Based on the data by Bricaud, Morel and Prieur (BMP81), we model absorption by CDOM by fitting an exponential curve of the form:

$$\alpha_y(\lambda) = \sigma_{a,y}(\lambda_0) e^{-S_y(\lambda-\lambda_0)} \quad (3.22)$$

where the subscript y denotes the constituent CDOM. λ_0 is a reference wavelength, often chosen to be 440 nm for yellow matter, and S_y is the slope of the semilogarithmic absorption curve (Kir94). S_y is usually taken to be constant, with a value of 0.014 nm⁻¹, but has been found to vary both geographically and temporally, and is also dependent on the wavelength range over which it is calculated (BMP81). The values of absorption $\sigma_{a,y}(\lambda_0)$ at reference wavelengths also vary in a range between 0.01 m⁻¹ to 20 m⁻¹, as a function of turbidity (Kir94).

Phytoplankton absorbs a great amount of visible light, due to its chlorophyll pigment. The absorption function for chlorophyll peaks strongly at 430 nm and 670 nm, being very weak in the mid range of the visible spectrum (thus the more phytoplankton the greener the hue of the water). The concentration of the chlorophyll in the water usually ranges from 0.01 mg/m³ for open waters to 100 mg/m³. The spectral absorption coefficient of the phytoplankton is usually expressed as a function of this concentration C as:

$$\sigma_{a,p}(\lambda) = C \sigma_{a,p}^*(\lambda) \quad (3.23)$$

where C can be defined as the concentration of the main pigment chlorophyll- a (Chl _{a}) or as the sum of the concentrations of Chl _{a} and its degradation products, the pheopigments. $\sigma_{a,p}^*$ is the *specific spectral absorption coefficient* (the absorption per unit of concentration) for a particular species of phytoplankton, given in m²/mg. Typical values for specific absorptions of different species of phytoplankton can be found in the work of Sathyendranath, Lazzara and Prieur (SLP87) (see Table 3.4). A rough correspondence between chlorophyll concentrations and several oceanic water types is given by Morel (Mor88).

3.5 The ocean: building a model from measured data and standards

The absorption owed to organic detritus and minerals can be approximated by an exponential function, according to Roesler, Perry and Carder(RPC89):

$$\sigma_{a,d}(\lambda) = \sigma_{a,d}(\lambda_0) e^{-S_d(\lambda-\lambda_0)} \quad (3.24)$$

Here the reference wavelength 400 nm is selected for λ_0 and typical values for the exponent coefficient S_d will be in the range between 0.006 nm⁻¹ to 0.014 nm⁻¹, although 0.011 nm⁻¹ is the most common value(RPC89). Further studies confirm that the absorption spectra of minerals and detritus is well described by an exponential function with an average slope S_d of 0.0123 nm⁻¹, with slightly lower values than predicted at wavelengths below 440 nm(BSF+03).

3.5.3 Modeling Elastic Scattering

Table 3.5: Scattering coefficient for detritus $\sigma_{s,dt}$ and minerals $\sigma_{s,m}$ (After Stramski et al.(SBM01)).

detritus	$\sigma_{s,dt}$	[m ⁻¹]	0.045	0.0375	0.0325	0.03	0.0285	0.0275	0.027	0.027
minerals	$\sigma_{s,m}$	[m ⁻¹]	0.0675	0.0525	0.05	0.045	0.04	0.036	0.034	0.032
total	$\sigma_{s,d}$	[m ⁻¹]	0.1125	0.09	0.0825	0.075	0.0685	0.0635	0.061	0.059
	λ	[nm]	380	440	500	550	610	670	720	780

For the pure water term we use the volume scattering function defined by the Einstein-Smoluchowski theory(Maz02), which models scattering at molecular level as small-scale fluctuations. Most of the related works use Rayleigh's scattering instead, which is just an approximation for scattering owed to water molecules. However, Einstein-Smoluchowski scattering is well defined and imposes no overheads in the simulations, while providing more accurate results. Its scattering coefficient and phase function are given by:

$$\sigma_{s,w}(\lambda) = 16.06 \beta_w(\lambda_0, 90^\circ) \left(\frac{\lambda_0}{\lambda} \right)^{4.32} \quad (3.25)$$

$$p_w(\theta) = 0.06225 (1 + 0.835 \cos^2 \theta) \quad (3.26)$$

Typical values for $\beta_w(\lambda_0, 90^\circ)$ for both fresh and saline pure water are given in(Mor74). These values range from $14.1 \cdot 10^{-4} m^{-1}$ to $134.5 \cdot 10^{-4} m^{-1}$.

All the scattering produced by CDOM has inelastic nature and thus will be described in next section.

Gordon and Morel(GM83) found that phytoplankton, even in small concentrations, also contribute to the total elastic scattering in the water. Its contribution is given by:

$$\sigma_{s,p}(\lambda) = \left(\frac{550}{\lambda} \right) 0.30 C^{0.62} \quad (3.27)$$

3. PHYSICALLY - BASED MODELING

where the constant 0.30 is selected to fit the data collected from many types of waters. The actual upper bound for this constant has a value of 0.45(GM83). The phase function due to phytoplankton is given by an isotropic function ($p_p = 1/\pi$).

The elastic scattering caused by organic detritus and minerals can be modeled based on Mie theory(GSO03). The Henyey-Greenstein phase function models forward scattering fairly well but fails to reproduce backscattering with the same precision. We found that we can achieve a better fit by using a Two-Terms Henyey-Greenstein phase function (TTHG)(HG41):

$$p_d(\theta, \zeta, g_f, g_b) = \tag{3.28}$$

$$\zeta p_{HG}(\theta, g_f) + (1 - \zeta) p_{HG}(\theta, g_b) \tag{3.29}$$

where ζ is a weighting function between zero and one. This common way of utilizing this combination defines a forward scattering lobe (first term), plus a backscattering lobe (second term), with $g_f \in [0..1]$ and $g_b \in [-1..0]$. p_{HG} represents a simple Henyey-Greenstein phase function (see section 3.1.3).

The TTHG function not only models backscattering more precisely, but it can describe more complex particle scattering models, improving the fit at large and small angles as well. The shape of each of the two HG functions can be approximated by an ellipsoid, avoiding the relatively expensive exponent in its evaluation. The observation was first introduced by Schlick(BLSS93). Due to the great variety of particulate matter, the scattering coefficient $\sigma_{s,d}$ can adopt a wide range of values. Table 3.5 shows typical values of this function (data after Stramski et al.(SBM01)).

3.5.4 Modeling Inelastic Scattering

For inelastic scattering, we need to model the possibility of an absorbed photon being re-emitted at a different wavelength. The function $f_\lambda(\lambda_i, \lambda_o)$, known as *wavelength redistribution function*, represents the efficiency of the energy transfer between wavelengths. It is defined as the quotient between the energy of the emitted wavelength and the energy of the absorbed excitation wavelength, per wavelength unit (see section 2.4.3).

The wavelength redistribution function f_λ can be seen as a re-radiation matrix. The three functions $\Gamma(\lambda')$, $f(\lambda', \lambda)$ and $\gamma(\lambda', \lambda)$, depend on both the medium and the type of inelastic event. The two inelastic events with more influence in the in-water light field are fluorescence and Raman scattering. Phytoplankton and CDOM are important fluorescence sources, whilst Raman scattering is produced by pure water; minerals and detritus, on the other hand, do not produce any inelastic event.

3.5.4.1 Fluorescence

Fluorescence occurs when a molecule absorbs a photon of wavelength λ_i , and re-emits it at a longer wavelength λ_o according to the *fluorescence efficiency function* $\gamma_F(\lambda_i, \lambda_o)$. The time lapse between the two events is 10^{-11} to 10^{-8} seconds, so it can be taken as an instantaneous process for computer graphics. For the two main sources of fluorescence (phytoplankton and CDOM), re-emission follows an isotropic phase function. For phytoplankton, the wavelength of the re-emitted photons is independent of the excitation wavelength, although the intensity does show wavelength dependency (Mob94).

It is very common in ocean waters to see a color shift ranging from greenish to very bright green, or even yellowish. These hue shifts are mainly due to the variation in the concentration and type of the suspended microorganisms, specially phytoplankton and its related chlorophyll concentration, which presents an absorption function peaking at 350 nm and rapidly decaying to almost zero beyond 500 nm . Only wavelengths between 370 and 690 nm can trigger fluorescence due to phytoplankton. This can be modeled as a dimensionless function $g_p(\lambda')$ so that:

$$g_p(\lambda_i) \equiv \begin{cases} 1 & \text{if } 370 \leq \lambda_i \leq 690\text{ nm} \\ 0 & \text{otherwise} \end{cases} \quad (3.30)$$

The wavelength-independent quantum yield for phytoplankton $\Gamma_p(\lambda_i)$ ranges from 0.01 to 0.1. Using (2.35) and (3.30), the relationship between the wavelength redistribution function $f_{\lambda,p}(\lambda_i, \lambda_o)$ and the spectral quantum efficiency function $\gamma_p(\lambda_i, \lambda_o)$ is:

$$f_{\lambda,p}(\lambda_i, \lambda_o) = \gamma_p(\lambda_i, \lambda_o) \frac{\lambda_i}{\lambda_o} \equiv \Gamma_p g_p(\lambda_i) h_p(\lambda_o) \frac{\lambda_i}{\lambda_o} \quad (3.31)$$

where $h_p(\lambda)$ is the *fluorescence emission function* per unit wavelength, and can be approximated by a gaussian (Mob94):

$$h_p(\lambda) = \frac{1}{\sqrt{2\pi} \lambda_\sigma} \exp \left\{ -\frac{(\lambda - \lambda_0)^2}{2(\lambda_\sigma)^2} \right\} \quad (3.32)$$

$\lambda_0 = 685\text{ nm}$ is the wavelength of maximum emission and $\lambda_\sigma = 10.6\text{ nm}$ represents the standard deviation.

The other important source of fluorescence in water is CDOM. For relatively high concentrations of CDOM, its quantum yield $\Gamma_y(\lambda')$ varies between 0.005 and 0.025. Following the work of Hawes (Haw92) we use the following formula to describe its spectral fluorescence quantum efficiency function:

$$f_{\lambda,y}(\lambda_i, \lambda_o) = \gamma_y(\lambda_i, \lambda_o) \frac{\lambda_i}{\lambda_o} \equiv \quad (3.33)$$

$$A_0(\lambda_i) \exp \left\{ -\left(\frac{\frac{1}{\lambda_o} - \frac{A_1}{\lambda_i} - B_1}{0.6 \left(\frac{A_2}{\lambda_i} + B_2 \right)} \right)^2 \right\} \frac{\lambda_i}{\lambda_o} \quad (3.34)$$

where A_0, A_1, A_2, B_1 and B_2 are empirical parameters whose values depend on the specific composition of the CDOM and can be found in (Mob94) (see Table 4.3). A_1 and A_2 are dimensionless, whereas the rest are given in nm^{-1} .

Our model can be easily extended to account for phosphorescence phenomena, which are intrinsically similar to fluorescence and are governed by the *phosphorescence efficiency function*. The only difference is that the re-emitted energy declines with time according to a function $d(t)$.

3. PHYSICALLY - BASED MODELING

3.5.4.2 Raman scattering

Raman scattering influences the in-water light field, specially at great depths where sun irradiance becomes zero and only Raman radiance remains. It occurs when vibration and rotation in water molecules exchange energy with incoming photons, re-emitting them with approximately the same wavelength, but allowing for small shifts towards longer or shorter wavelengths. It can also be considered a spontaneous process. To isolate Raman inelastic events from fluorescence and other scattering events, it is usually studied in pure water, filtered several times, so that the second term in (3.20) becomes zero.

The Raman wavelength redistribution function $f_{\lambda,w}(\lambda_i, \lambda_o)$ is usually described in terms of a sum of four Gaussian functions(Mob94):

$$f_{\lambda,w}(\lambda_i, \lambda_o) = \tag{3.35}$$

$$\frac{10^7 \sum_{j=1}^4 A_j \frac{1}{\Delta \tilde{\nu}_i} \exp \left\{ - \frac{[10^7 (\frac{1}{\lambda_i} - \frac{1}{\lambda_o}) - \tilde{\nu}_i]^2}{\Delta \tilde{\nu}_i^2} \right\}}{\lambda_i^2 \sqrt{\frac{\pi}{4 \ln 2}} \sum_{j=1}^4 A_j} \tag{3.36}$$

where $\tilde{\nu}$ is the wavenumber ($\tilde{\nu} = 10^7/\lambda$) given in cm^{-1} . Typical parameter values A_i , $\tilde{\nu}_i$ and $\Delta \tilde{\nu}_i$ for the Raman redistribution function are given by Walrafen(Wal69) and are shown in Table 4.3.

The algorithm able to render all these complex Stokes and anti-Stokes inelastic behavior on participating media is also a contribution of this PhD, and can be found on section 4.2. Renders and results are in section 5.3. This work has been published in the Computer Graphics Forum (GSMA08).

3.6 Summary

This chapter has introduced some concepts about the modeling of materials and participating media for simulation of light transport. Later, a contribution of this PhD has been presented: an algorithm that estimates the materials of objects from single pictures, as an example of how material properties can be deduced from images. The first presented approach to material capture is using genetic algorithms for inverse rendering (see section 3.2), and has already been published (MMT⁺09; MMTG09). In section 3.3, a new more efficient and practical solution for translucent material estimation (using linear systems) is proposed.

Although our first approach for modeling is estimation from images, section 3.5 has introduced another approach and another contribution of this paper. Instead of estimating properties from pictures, the physical models can be obtained from literature that have measured them in fields and all of them can be merged into a comprehensive dataset. This approach has been applied to the creation of a bio-optical model of ocean water, that includes complex phenomena such as inelastic scattering. This model is used for rendering with a specific algorithm that also is a contribution of this PhD, presented in section 4.2. The result of this work has been published in the Computer Graphics Forum (GSMA08).

Chapter 4 also introduces another approach for material modeling, which is simulating from its fundamental phenomena. Section 4.4 presents one of the most interesting phenomena in nature: the rainbow, and how can it be simulated from the fundamental interactions between electromagnetic waves and water particles. It has been included into the simulation chapter because it consists mainly on simulating the interactions, although the result is a phase function (and thus could be also considered as modeling).

Chapter 5 gathers all the results of these contributions.

References

- [APS00] Michael Ashikmin, Simon Premože, and Peter Shirley, *A microfacet-based brdf generator*, SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques (New York, NY, USA), ACM Press/Addison-Wesley Publishing Co., 2000, pp. 65–74. [35](#)
- [BBC⁺94] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, R. Pozo, V. Eijkhout, H. Van der Vorst, and C. Romine, *Templates for the solution of linear systems: Building blocks for iterative methods, 2nd edition*, SIAM, 1994. [45](#), [47](#)
- [BKY99] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille, *The bas-relief ambiguity*, International Journal of Computer Vision **35** (1999), no. 1, 33–44. [50](#)
- [Bli77] James F. Blinn, *Models of light reflection for computer synthesized pictures*, SIGGRAPH '77: Proceedings of the 4th annual conference on Computer graphics and interactive techniques (New York, NY, USA), ACM, 1977, pp. 192–198. [35](#)
- [BLSS93] P. Blasi, B. Le Saec, and C. Schlick, *A rendering algorithm for discrete volume density objects*, Computer Graphics Forum (Eurographics 93) **12** (1993), no. 3, 201–210. [56](#)
- [BMP81] A. Bricaud, A. Morel, and L. Prieur, *Absorption by dissolved organic matter of the sea (yellow substance) in the uv and visible domains*, Limnology and Oceanography **26** (1981), no. 1, 43–53. [54](#)
- [BSF⁺03] M. Babin, D. Stramski, G. M. Ferrari, H. Claustre, A. Bricaud, G. Obolensky, and N. Hoepffner, *Variations in the light absorption coefficients of phytoplankton, nonalgal particles, and dissolved organic matter in coastal waters around europe*, J. Geophys. Res. **108(C7)** (2003), no. 3211. [55](#)
- [CLH⁺08] C-W. Chang, W-C. Lin, T-C. Ho, T-S. Huang, and J-H. Chuang, *Real-time translucent rendering using gpu-based texture space importance sampling*, Computer Graphics Forum **27** (2008), no. 2, 517–526. [39](#)
- [CT81] Robert L. Cook and Kenneth E. Torrance, *A reflectance model for computer graphics*, SIGGRAPH '81: Proceedings of the 8th annual conference on Computer graphics and interactive techniques (New York, NY, USA), ACM, 1981, pp. 307–316. [35](#)
- [DHT⁺00] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, , and Mark Sagar, *Acquiring the reflectance field of a human face*, SIGGRAPH, ACM, 2000. [39](#)
- [DJ05] C. Donner and H. W. Jensen, *Light diffusion in multi-layered translucent materials*, SIGGRAPH (Los Angeles, California), ACM, 2005, pp. 1032–1039. [39](#)

REFERENCES

- [dLE07] E. d'Eon, D. Luebke, and E. Enderton, *Efficient rendering of human skin*, Eurographics Symposium on Rendering (Grenoble, France) (J. Kautz and S. Pattanaik, eds.), Eurographics Association, 2007, pp. 147–157. [39](#), [48](#)
- [DWd⁺08] Craig Donner, Tim Weyrich, Eugene d'Eon, Ravi Ramamoorthi, and Szymon Rusinkiewicz, *A layered, heterogeneous reflectance model for acquiring and rendering human skin*, (New York, NY, USA), vol. 27, ACM, 2008, pp. 1–12. [39](#)
- [FB05] R. W. Fleming and H. H. Bühlhoff, *Low-level image cues in the perception of translucent materials*, ACM Transactions on Applied Perception (TAP) **2** (2005), no. 3, 346–382. [49](#)
- [FCF07] Jeppe Revall Frisvad, Niels Jørgen Christensen, and Peter Falster, *The aristotelian rainbow: from philosophy to computer graphics*, GRAPHITE, ACM, 2007, pp. 119–128. [37](#)
- [FCJ07] Jeppe Revall Frisvad, Niels Jørgen Christensen, and Henrik Wann Jensen, *Computing the scattering properties of participating media using lorenz-mie theory*, SIGGRAPH '07: ACM SIGGRAPH 2007 papers (New York, NY, USA), ACM, 2007, p. 60. [37](#)
- [FH04] Hui Fang and John C Hart, *Textureshop: texture synthesis as a photograph editing tool*, ACM Transactions on Graphics **23** (2004), no. 3, 354–359. [45](#)
- [ftPoWS] The International Association for the Properties of Water and Steam, *Release on the refractive index of ordinary water substance as a function of wavelength, temperature and pressure*, <http://www.iapws.org/relguide/rindex.pdf>. [34](#)
- [GHP⁺08] Abhijeet Ghosh, Tim Hawkins, Pieter Peers, Sune Frederiksen, and Paul Debevec, *Practical modeling and acquisition of layered facial reflectance*, (New York, NY, USA), vol. 27, ACM, 2008, pp. 1–10. [39](#)
- [GLL⁺04] Michael Goesele, Hendrik P. A. Lensch, Jochen Lang, Christian Fuchs, and Hans-Peter Seidel, *Disco: acquisition of translucent objects*, SIGGRAPH (New York, NY, USA), ACM, 2004, pp. 835–844. [39](#)
- [GM83] H. R. Gordon and A. Morel, *Remote assessment of ocean color for interpretation of satellite visible imagery: A review*, Lecture Notes on Coastal and Estuarine Studies, vol. 4, Springer-Verlag, New York, 1983. [55](#)
- [GMN94] Jay S. Gondek, Gary W. Meyer, and Jonathan G. Newman, *Wavelength dependent reflectance functions*, SIGGRAPH '94 (New York, NY, USA), ACM, 1994, pp. 213–220. [35](#)
- [GNG⁺08] Jinwei Gu, Shree Nayar, Eitan Grinspun, Peter Belhumeur, and Ravi Ramamoorthi, *Compressive structured light for recovering inhomogeneous participating media*, Computer Vision ECCV 2008, 2008, pp. 845–858. [37](#)
- [Gre90] R. Greenler, *Rainbows, halos, and glories*, Cambridge University Press, 1990. [37](#)
- [GSLM⁺08] Diego Gutierrez, Francisco J. Seron, Jorge Lopez-Moreno, Maria P. Sanchez, Jorge Fandos, and Erik Reinhard, *Depicting procedural caustics in single images*, SIGGRAPH Asia (New York, NY, USA), ACM, 2008. [49](#)
- [GSMA06] Diego Gutierrez, Francisco J. Serón, Adolfo Muñoz, and Oscar Anson, *Simulation of atmospheric phenomena.*, Computers & Graphics **30** (2006), no. 6, 994–1010. [34](#)
- [GSMA08] Diego Gutierrez, Francisco J. Serón, Adolfo Muñoz, and Oscar Anson, *Visualizing underwater ocean optics*, Computer Graphics Forum **27** (2008), no. 2, 547–556.

-
- [GSO03] Rebeca E. Green, Heidi M. Sosik, and Robert J. Olson, *Contributions of phytoplankton and other particles to inherent optical properties in new england continental shelf waters*, *Limnol. Oceanogr.* **48** (2003), no. 6, 2377–2391. 56
- [GWJ+08] M. Glencross, G. J. Ward, C. Jay, J. Liu, F. Melendez, and R. Hubbard, *A perceptually validated model for surface depth hallucination*, To Appear in SIGGRAPH (Los Angeles, California), ACM, 2008. 39
- [Haw92] S.K. Hawes, *Quantum fluorescence efficiencies of marine fulvic and humic acids*, Ph.D. thesis, Dept. of Marine Science, Univ. of South Florida, 1992. 57
- [HCG08] D. Hubball, M. Chen, and P.W. Grant, *Image-based aging using evolutionary computing*, *Computer Graphics Forum*, **27** (2008), no. 2, 607–616. 39
- [HED05] Tim Hawkins, Per Einarsson, and Paul Debevec, *Acquisition of time-varying participating media*, *ACM Trans. Graph.* **24** (2005), no. 3, 812–815. 37
- [HG41] L.G. Henyey and J.L. Greenstein, *Diffuse radiation in the galaxy*, *Astrophysics Journal* **93** (1941), 70–83. 36, 56
- [Jam07] John James, *Spectrograph design fundamentals*, Cambridge University Press, 2007. 34
- [JB02] H. W. Jensen and J. Buhler, *A rapid hierarchical rendering technique for translucent materials*, SIGGRAPH (San Antonio, Texas), ACM, 2002, pp. 576–581. 38, 45
- [JMLH01] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan, *A practical model for subsurface light transport*, SIGGRAPH (Los Angeles, California), ACM, 2001, pp. 511–518. 38, 45, 48, 49
- [Joh02] Scott F. Johnston, *Lumo: illumination for cel animation*, NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering (New York, NY, USA), ACM, 2002, pp. 45–52. 50
- [KA06] J. Koljonen and J.T. Alander, *Genetic algorithm for optimizing fuzzy image pattern*, Proceedings of the Ninth Scandinavian Conference on Artificial Intelligence (SCAI 2006) (2006), 46–53. 39
- [Kir94] John T.O. Kirk, *Light and photosynthesis in aquatic ecosystems*, Cambridge University Press, New York, 1994. 54
- [KRFB06] E. A. Khan, E. Reinhard, R. W. Fleming, and H. H. Bühlhoff, *Image-based material editing*, SIGGRAPH (Boston, Massachusetts), ACM, 2006, pp. 654–663. 39, 45, 49, 50
- [KT94] A.J. Katz and P.R. Thrift, *Generating image filters for target recognition by genetic learning*, *PAMI* **16** (1994), no. 9, 906–910. 39
- [Kv01] J. J. Koenderink and A. J. van Doorn, *Shading in the case of translucent objects*, *Human Vision and Electronic Imaging* (San Jose, California), SPIE, 2001, pp. 312–320. 50
- [Lava] Philip Laven, *Mieplot* - <http://www.philiplaven.com/mieplot.htm>. 37
- [Lavb] Philip Laven, <http://www.philiplaven.com/>. 37
- [LFTW06] H. Li, S.C. Foo, K.E. Torrance, and S.H. Westin, *Automated three-axis gonioreflectometer for computer graphics applications*, *Optical Engineering* **45** (2006), no. 4, 1–11. 35
- [LL01] D. Lynch and W. Livingstone, *Color and light in nature (second edition)*, Cambridge University Press, 2001. 37

REFERENCES

- [LMHRG09] Jorge Lopez-Moreno, Sunil Hadap, Erik Reinhard, and Diego Gutierrez, *Light source detection in photographs*, Congreso Español de Informatica Grafica (CEIG), 2009, pp. 161–169. [50](#), [51](#)
- [Lor90] L. Lorenz, *Lysbevælgelser i og uden for en af plane lysblger belyst kugle. det kongelig danske videnskabernes selskabs skrifter*, Rkke, Naturvidenskabelig og Matematisk Afdeling **6** (1890), no. 1, 2–62. [37](#)
- [Maz02] Robert M. Mazo, *Brownian Motion: Fluctuations, Dynamics and Applications*, International Series of Monographs on Physics, vol. 112, ch. Einstein-Smoluchowski Theory, pp. 46–62, International Series of Monographs on Physics, Oxford University Press Inc., Great Clarendon Street, Oxford, 2002, pp. 46–62. [55](#)
- [Mie08] G. Mie, *Beitr age zur optik tr uber medien, speziell kolloidaler metall osungen*, Annalen der Physik **25** (1908), no. 3, 377–445. [37](#)
- [Min93] M. Minnaert, *Light and colour in the outdoors*, Springer-Verlag, 1993. [37](#)
- [MMT⁺09] Belen Masia, Adolfo Munoz, Alfonso Tolosa, Oscar Anson, Jorge Lopez-Moreno, Jorge Jimenez, and Diego Gutierrez, *Genetic algorithms for estimation of reflectance parameters*, Poster Proceedings of Spring Conference on Computer Graphics, 2009, pp. 39–44. [40](#), [43](#), [58](#)
- [MMTG09] Adolfo Munoz, Belen Masia, Alfonso Tolosa, and Diego Gutierrez, *Single-image appearance acquisition using genetic algorithms*, Proceedings of Computer Graphics, Visualization, Computer Vision and Image Processing, 2009, pp. 24–32. [40](#), [43](#), [58](#)
- [Mob94] Curtis D. Mobley, *Light and water: Radiative transfer in natural waters*, Academic Press, Inc., San Diego, 1994. [36](#), [52](#), [53](#), [56](#), [57](#), [58](#)
- [Mor74] A. Morel, *Optical aspects of oceanography*, Academic Press, New York, 1974. [53](#), [55](#)
- [Mor88] A. Morel, *Optical modeling of the upper ocean in relation to its biogenous matter content (case i waters)*, Journal of Geophysical Research **93** (1988), no. C9, 10749–10768. [54](#)
- [MPZ⁺02] W. Matusik, H. Pfister, R. Ziegler, A. Ngan, and L. McMillan, *Acquisition and rendering of transparent and refractive objects*, Thirteenth Eurographics Workshop on Rendering, EG, 2002, pp. 267–277. [39](#)
- [MSY08] Y. Mukaigawa, K. Suzuki, and Y. Yagi, *Analysis of subsurface scattering under generic illumination*, ICPR, 2008. [39](#)
- [Mus89] F. Kenton Musgrave, *Prisms and rainbows: A dispersion model for computer graphics*, Graphics Interface '89, June 1989, pp. 227–234. [37](#)
- [NGD⁺06] S. G. Narasimhan, M. Gupta, C. Donner, R. Ramamoorthi, S. K. Nayar, and H. W. Jensen, *Acquiring scattering properties of participating media by dilution*, SIGGRAPH (Boston, Massachusetts), ACM, 2006, pp. 1003–1012. [37](#)
- [NPL] UK National Physical Laboratory, *Refractive index of optical materials*, http://www.kayelaby.npl.co.uk/general_physics/2_5/2_5_8.html. [34](#)
- [PF97] Robin M. Pope and Edward S. Fry, *Absorption spectrum (380–700 nm) of pure water. ii. integrating cavity measurements*, Applied Optics **36** (1997), no. 33, 8710–8723. [54](#)
- [Pho73] B.T. Phong, *Illumination of computer-generated images*, Ph.D. thesis, University of Utah, July 1973. [35](#)

-
- [Poy96] Charles A. Poynton, *A technical introduction to digital video*, John Wiley & Sons, Inc., New York, NY, USA, 1996. 41
- [Pre76] R. W. Preisendorfer, *Introduction*, Hydrologic Optics, vol. 1, National Technical Information Service, Springfield, IL, 1976. 52
- [PvBM⁺06] Pieter Peers, Karl vom Berge, Wojciech Matusik, Ravi Ramamoorthi, Jason Lawrence, Szymon Rusinkiewicz, and Philip Dutré, *A compact factored representation of heterogeneous subsurface scattering*, SIGGRAPH (New York, NY, USA), ACM, 2006, pp. 746–753. 39
- [QY02] X. Qin and Y.H. Yang, *Estimating parameters for procedural texturing by genetic algorithms*, Graphical Models **64** (2002), no. 1, 19–39. 39
- [RL94] G. Roth and M.D. Levine, *Geometric primitive extraction using a genetic algorithm*, PAMI **16** (1994), no. 9, 901–905. 39
- [RPC89] C. S. Roesler, M. J. Perry, and K. L. Carder, *Modeling in situ phytoplankton absorption from total absorption spectra in productive inland marine waters*, Limnology and Oceanography **34** (1989), no. 8, 1510–1523. 54, 55
- [Rus95] H. Rushmeier95, *Input for participating media*, Realistic Input for Realistic Images, ACM SIGGRAPH 95 Course Notes. Also appeared in the ACM SIGGRAPH 98 Course Notes - A Basic Guide to Global Illumination, 1995. 37
- [SB81] R. C. Smith and K. S. Baker, *Optical properties of the clearest natural waters (200–800 nm)*, Appl. Opt. **20** (1981), 177–184. 53, 54
- [SBM01] D. Stramski, A. Bricaud, and A. Morel, *Modeling the inherent optical properties of the ocean based on the detailed composition of planktonic community*, Applied Optics **40** (2001), 2929–2945. 55, 56
- [Sim91] K. Sims, *Artificial evolution for computer graphics*, SIGGRAPH '91 (New York, NY, USA), ACM, 1991, pp. 319–328. 39
- [SLP87] Shubha Sathyendranath, Luigi Lazzara, and Louis Prieur, *Variations in the spectral values of specific absorption of phytoplankton*, Limnology and Oceanography **32** (1987), no. 2, 403–415. 53, 54
- [STPP09] Y. Song, X. Tong, F. Pellacini, and P. Peers, *Subedit: A representation for editing measured heterogeneous subsurface scattering*, ACM Transactions on Graphics (SIGGRAPH) **28** (2009), no. 3, 1–10. 49
- [STW99] P. Salek, J. Tarasiuk, and K. Wierzbanski, *Application of genetic algorithms to texture analysis*, Crystal Research and Technology **34** (1999), 1073–1079. 39
- [TGL⁺06] S. Tariq, A. Gardner, I. Llamas, A. Jones, P. Debevec, and G. Turk, *Efficient estimation of spatially varying subsurface scattering parameters*, Workshop on Vision, Modeling, and Visualization (VMW) (Aachen, Germany), 2006. 39
- [USA76] *U.s. standard atmosphere*, U.S. Government Printing Office, Washington, D.C., 1976. 34
- [van57] H. C. van de Hulst, *Light scattering by small particles*, Dover Publications Inc., 1957. 37
- [Wal69] G. E. Walrafen, *Continuum model of water—an erroneous interpretation*, Journal of Chemical Physics **50** (1969), no. 1, 567–569. 58
- [War92a] G. J. Ward, *Measuring and modeling anisotropic reflection*, SIGGRAPH (Chicago, Illinois), ACM, 1992, pp. 265–272. 35

REFERENCES

- [War92b] Gregory J. Ward, *Measuring and modeling anisotropic reflection*, SIGGRAPH Comput. Graph. **26** (1992), no. 2, 265–272. [35](#)
- [WAT92] Stephen H. Westin, James R. Arvo, and Kenneth E. Torrance, *Predicting reflectance functions from complex surfaces*, SIGGRAPH Comput. Graph. **26** (1992), no. 2, 255–264. [35](#)
- [WLL⁺08] Tim Weyrich, Jason Lawrence, Hendrik Lensch, Szymon Rusinkiewicz, and Todd Zickler, *Principles of appearance acquisition and representation*, SIGGRAPH '08: ACM SIGGRAPH 2008 classes (New York, NY, USA), ACM, 2008, pp. 1–119. [33](#)
- [WMP⁺06] Tim Weyrich, Wojciech Matusik, Hanspeter Pfister, Bernd Bickel, Craig Donner, Chien Tu, Janet McAndless, Jinho Lee, Addy Ngan, Henrik Wann Jensen, and Markus Gross, *Analysis of human faces using a measurement-based skin reflectance model*, ACM SIGGRAPH, vol. 25, 2006, pp. 1013–1024. [39](#)
- [WPGC95] G. Winter, J. Priaux, M. Galn, and P. Cuesta, *Genetic algorithms in engineering and computer science*, John Wiley and sons, 1995. [40](#)
- [WS02] Y. Wang and D. Samaras, *Estimation of multiple illuminants from a single image of arbitrary known geometry*, European Conference on Computer Vision (ECCV), Springer, 2002, pp. 272–288. [50](#)
- [WSTS08] Tai-Pang Wu, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum, *Interactive normal reconstruction from a single image*, ACM Transactions on Graphics (SIGGRAPH Asia) **27** (2008), no. 5, 1–9. [50](#)
- [WT06] T-P Wu and C-K Tang, *Separating subsurface scattering from photometric image*, International Conference on Pattern Recognition (ICPR), IEEE Computer Society, 2006, pp. 207–210. [39](#)
- [WZT⁺08] J. Wang, S. Zhao, X. Tong, S. Lin, Z. Lin, Y. Dong, B. Guo, and H-Y. Shum, *Modeling and rendering of heterogeneous translucent materials using the diffusion equation*, ACM Transactions on Graphics (TOG) **27** (2008), no. 1, 1–18. [39](#), [45](#), [46](#)
- [XGL⁺07] K. Xu, Y. Gao, Y. Li, T. Ju, and S-M. Hu, *Real-time homogenous translucent material editing*, Computer Graphics Forum **26** (2007), no. 3, 545–552. [45](#), [48](#)
- [ZY01] Y. Zhang and Y-H. Yang, *Multiple illuminant direction detection with application to image synthesis*, IEEE Transactions on Pattern Analysis and Machine Intelligence **23** (2001), no. 8, 915–920. [50](#)

Chapter 4

Simulation

4.1 Basics and previous work

4.1.1 Monte Carlo integration

Most of the equations that define the outgoing radiance given input lighting and material or media properties (such as equations 2.29, 2.31, 2.33 or 2.34) present integrals that most of the times cannot be solved analytically. As a consequence, the result of these integrals must be somehow estimated. In computer graphics, a widely used estimator is obtained by applying Monte Carlo integration:

$$\int_V f(x)dx \approx \frac{V}{N} \sum_{i=1}^N f(\hat{x}_i) \quad (4.1)$$

where V refers to the integration volume, x is the integration variable (that can be n-dimensional) and $\hat{x}_i \in V$ are the N randomly chosen samples from which the integral is estimated. This estimator is unbiased and consistent, although its variance is $N^{-\frac{1}{2}}$, which means that for reducing the error by half, the number of samples must be increased four times.

There are several sampling strategies for Monte Carlo integration, such as stratified or importance sampling, which have been used for different rendering algorithms (DBB02). More detailed explanations about different sampling techniques can be found on several books (KW08). All the contributions of this PhD are based on Monte Carlo integration.

4. SIMULATION

4.1.2 Ray tracing

Monte Carlo integration for computer graphics is mainly applied to ray tracing techniques (Gla89). The main idea is to discretize the continuum of light paths into discrete segments, called rays. The goal of these rays is to approximate the behavior of light. Once a scene has been modeled, including camera, light sources and the mathematical definition of the surfaces (triangles, most of the time), rays are cast to the camera and intersected with the geometry. These intersections are computed by creating a system of equations, that includes the ray (the equation of a straight line) and geometry (with its specific equation). The solution among the intersections with all the objects of the scene that is closest to the beginning of the ray is considered as the intersection. The first ray cast is often called *primary ray*.

Then, color is estimated by applying Monte Carlo integration to equation 2.31 for the corresponding BRDF (or the function that defines the properties of the surface). Light sources are directly queried by casting a *shadow ray* from the intersection point to the position of the light source, that checks if there is any other geometry blocking the radiance that comes directly from the light source. This shadow ray is estimating direct illumination.

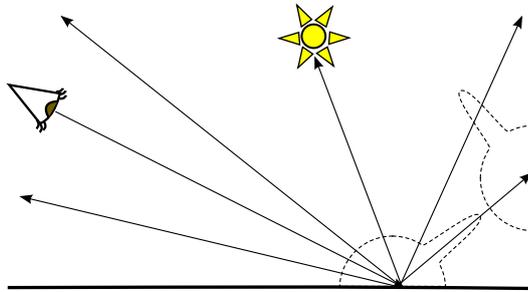


Figure 4.1: Basic ray tracing

Indirect illumination (radiance coming from previous light interactions) is estimated by choosing N random samples (new rays) on the hemisphere (according to the BRDF), and shooting them towards the geometry again. The incoming radiance is recursively obtained at the new intersection points (up to a certain level in order to prevent endless recursion). These rays are called *secondary rays*. Figure 4.1 shows the ray-tracing process.

Usually, as the number of geometry objects increases, the ray tracing process becomes slower (as finding the closest intersection needs more and more computations). There are plenty of acceleration structures (GS87; RSH00; WK06) that optimize the search for the closest intersection, and are used quite often. Also, there are much research based on ray tracing based algorithms, that aim to give better and better estimates in the shortest time possible (PH04).

4.1.3 Photon mapping

Photon mapping (Jen01) is a technique that has become a milestone in global illumination simulation. Recursive ray tracing presents two important shortcomings:

- Being Monte Carlo based, reducing noise to the half requires four times the number of samples. Furthermore, this noise is high frequency and thus very noticeable.

- There are some light paths that correspond to relevant phenomena (such as caustics) that are hardly found.

Although there are several Monte Carlo techniques that deal with these issues, the most relevant breakthrough is the photon mapping technique. Its main advantage is that it decouples the paths coming from the light sources and the paths coming from the camera. This enables an easier simulation of several phenomena such as caustics and color bleeding, and the noise of the result is low-frequency.

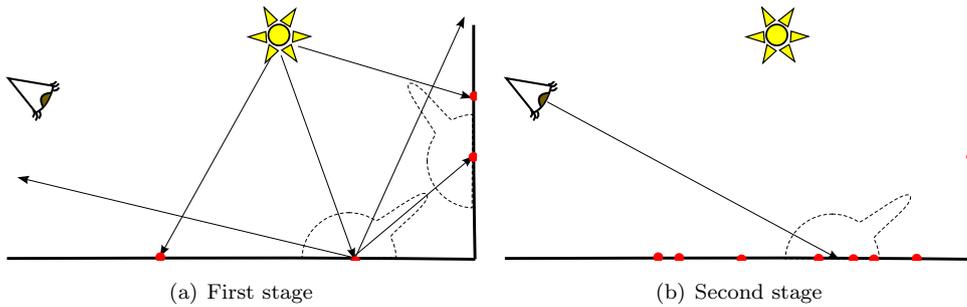


Figure 4.2: Photon mapping

It consists on two stages:

- On the first stage, particles that contain illumination information are cast from the light sources, recursively interacting with the geometry in the same way than rays in recursive ray tracing. When a photon interacts with a diffuse surface, its lighting information is stored at the point of interaction into a data structure (the photon map).
- On the second stage, rays are traced from the camera in the same way than ray tracing. However, at each interaction, incident radiance is estimated from the closest photons that have been stored in the photon map. This stage introduces some bias in the solution while reducing the frequency of the noise.

Figure 4.2 represents the two stages of the photon mapping algorithm. There are hybrid algorithms (between recursive ray tracing and photon mapping) that take the best from the two approaches. Details can be found on a specialized book (Jen01).

4.1.4 Participating media

The simulation of light transport in participating media usually relies on Monte-Carlo techniques for ray tracing (CPP+05; GNJJ08; GJJ09). The basic Monte-Carlo technique is *ray marching*, and is the natural extension of ray tracing for participating media, but applied to the Radiative Transfer Equation (equation 2.29).

A ray that traverses a participating medium is split into different steps and incident radiance is queried at every of them. Shadow rays are cast towards the light sources, computing single scattering by adding all the incident radiance weighted by the phase function.

However, part of the incoming irradiance can come from a previous interaction (what is called *multiple scattering*). It can be computed by integrating along the phase function by using Monte Carlo

4. SIMULATION

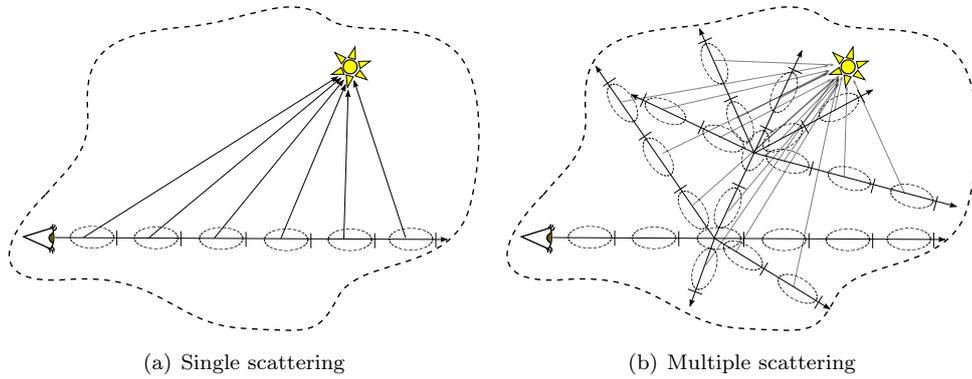


Figure 4.3: Ray marching. Notice that for clarity multiple scattering is shown just in a single step, while in reality is computed at each of them.

techniques, sampling the phase function along the whole sphere and recursively shooting secondary rays (see figure 4.3).

Notice that the full ray marching approach is computationally very expensive. Depending on the ray marching step, high frequency noise will appear (if it is too big) or many steps will be computed (which is expensive). This increases exponentially with multiple scattering, as for every step several new rays need to be computed. Nevertheless, it is possible to simulate participating media (or at least multiple scattering) using photon mapping. This is less computationally expensive and the resulting noise presents low frequency.

It is again a two-pass algorithm. In the first pass, photons are cast from the light sources and stored in the volume, and on the second step ray marching is applied and radiance is estimated from nearest photons at each ray marching step (see figure 4.4). Most of the times, single scattering is computed using standard ray marching, and multiple scattering is computed from the photon map. An detailed description can be found in the original photon mapping book ([Jen01](#)).

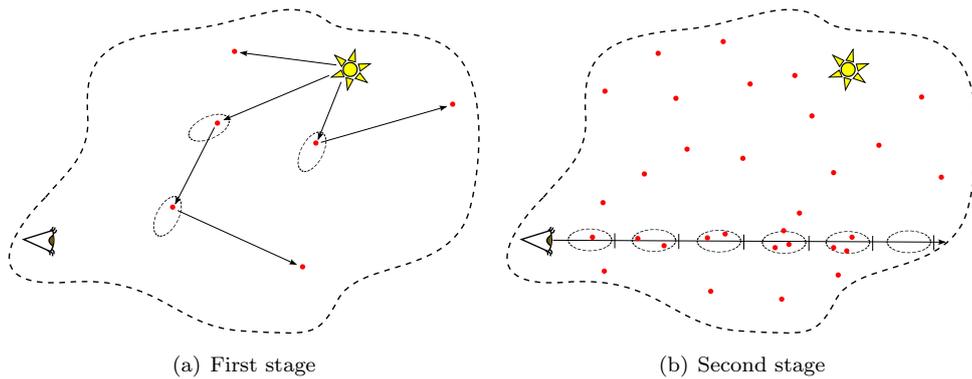


Figure 4.4: Photon mapping for participating media

However, being such a complex and high dimensional scattering problem, there are many Monte Carlo techniques that are able to simulate participating media ([CPP+05](#); [GNJJ08](#)). There have been also attempts to solve in different ways the Radiative Transfer Equation (RTE, see section 2.4.1), such

as the method proposed by Kaneda et al. (KYNN91). Nishita et al. (NSTN93) display water from outer space modifying this method, but both works only take into account single scattering. In the work of Premoze and Ashikhmin (PA01), no radiance due to scattering is calculated at all, using empirical equations based on experimental data instead. Mobley (Mob94) developed a method to solve the RTE analytically, but it cannot be extended to take into account inelastic scattering. Part of the research of this PhD extends the ray marching / photon mapping algorithm to simulate inelastic scattering for participating medium in a global illumination context (see section 4.2).

Translucent materials can be treated as a participating medium behind a surface, so they can be simulated using photon mapping. However, there are plenty of techniques which are specific for translucent materials (JMLH01; JB02; DJ05; dLE07).

4.1.5 Inelastic scattering

There is some previous work that deals with the simulation of inelastic scattering for surfaces (Gla94; WTP01; WWLP06). Inelastic scattering for participating media has also been a research target: Cerezo and Seron (CS04) can simulate underwater imagery, and even include a bio-optical model (although simpler than the one introduced in section 3.5). Furthermore, they use a discrete ordinate method, which requires an angular and spatial discretization of the volume (high memory requirements), plus they cannot provide a full solution to the light transport problem. Specular-to-diffuse paths defining caustics, for instance, cannot be obtained. Gutierrez et al. (GMAS05) present a method that takes into account the simulation of some inelastic scattering events, although their model is very simplified. Both of these works are limited in the number of inelastic phenomena that are able to simulate. For instance, none of them considers anti-Stokes interactions, thus missing important aspects of the radiance transport such as Raman scattering (whose influence in the radiance fields picked up by remote sensing should not be neglected (MVB03)). The simulation method presented in section 4.2 is able to render both Stokes and anti-Stokes inelastic scattering, while using the data from the complete bio-optical ocean water model introduced in section 3.5.

4.1.6 Non-linear media

There is not a lot of previous works that take into account non-linear media for rendering. Some of them are specific for atmospheric phenomena, in which the index of refraction varies at different heights. Berger et al. (BTL90) and Lintu et al. (LHM05) divide the atmosphere into several layers (perpendicular to height) and compute paths for each of them. Musgrave (Mus90) proposes a model based just in total reflection. These models are limited to certain scenes and lose physical accuracy.

Gröller (Grö95) proposes a more general approach for non-linear ray tracing, although its work focuses on mathematical and physical systems instead on the effect of the index of refraction on the path of light. Stam and Languenou (SL96) solves this problem by using geometrical optics but the distribution of the index of refraction is still limited.

Gutierrez et al. (GSMA06) present a model that takes into account the Eikonal equation, including a model of the Earth's atmosphere by describing its varying index of refraction.

When the medium traversed by light is non-homogeneous, with the index of refraction varying continuously from point to point, the trajectory of light is affected at each differential step. The final result of this differential change is a curved path that distorts the normal view of a real scene, but that path cannot be efficiently calculated by using traditional ray tracing. Fermat's principle (see

4. SIMULATION

section 2.2.1) can be applied to obtain the curved trajectory of the light rays. Doing the appropriate deduction (GSAM04; Hal89) Fermat’s principle gives the following equation:

$$\frac{d}{dl} \left(n_\lambda \frac{d\vec{r}}{dl} \right) - \nabla n_\lambda = 0 \leftrightarrow \frac{d}{dl} \left(n_\lambda \frac{dx_j}{dl} \right) - \frac{\partial n_\lambda}{\partial x_j} = 0 \quad (4.2)$$

where l is the length of the arc, n is the index of refraction at each differential point of the medium and ($j = 1, 2, 3$) are the three coordinates of a point.

This equation can be identified as an Initial Value Problem (GSMA06) and solved using numerical methods such as the Euler’s (BF88). Part of the research of this PhD deals with the study and improvement of the efficiency if this approach (see section 4.3), so non-linear media can be simulated in a practical time frame without losing physical accuracy.

4.1.7 Wave phenomena

Given its complexity, light polarization (see section 2.1.1.1) has very often been neglected in computer graphics. For polarization simulation, Stokes polarization notation has often been used (WTP01).

Even less effort has been done for simulating light wave interference (see section 2.1.1.2) for computer graphics. There is one exception which deals with interference on surfaces (GMN94). Interference on participating media (mostly rainbows) are almost always simulated by the use of the Lorenz-Mie theory (Lor90; Mie08) (see section 4.4 for details). Furthermore, there is a very interesting on-line resource that considers such phenomena (Lavb), including a software that is capable of simulating phenomena owed to Mie theory, named MiePlot (Lava).

This PhD deals with the simulation of the phase function of rainbows, by considering both polarization and interference (see section 4.4). The resulting tabulated phase function can be used as input for any participating media rendering algorithm in order to get an image of a rainbow¹.

4.2 Simulation of inelastic scattering on participating media

Having so far developed a bio-optical model as in section 3.5, we can now formalize it into a set of parameters and equations to fully simulate the in-water light field. To summarize, the four constituents of the model (for which the coefficients and phase functions have been defined in the bio-optical mode of section 3.5) and their interactions with light are given in Table 4.1. Table 4.2 shows how the main functions that define the model are derived from IOP and related functions at constituent level. This research has already been published (GSMA08).

The model allows for easy adjusting of its parameters to simulate different types of water and thus obtain different in-water light fields. As well as minerals and detritus, other particulate components of water can be added from oceanographic studies (although minerals and detritus have the greatest influence in the final appearance of water). Mie theory can again be used to model the scattering by these new particles, and the phase function can be approximated by using a two term Henyey-Greenstein phase function (3.28). An overview of the most significant parameters of the model, the equations in which they can be found and the corresponding values used for the simulations in this

¹While the simulation of a phase function can be considered as *modeling* the phase function, we have preferred to include the algorithm in the simulation chapter (chapter 4) instead of in the modeling one (chapter 3)

4.2 Simulation of inelastic scattering on participating media

Constituent	Absorption	Elastic Scat.	Inelastic Scat.
Pure water	Yes	Yes	Raman Scattering
Minerals, detritus	Yes	Yes	No
Phytoplankton	Yes	Yes	Fluorescence
CDOM	Yes	No	Fluorescence

Table 4.1: Water constituents and interactions

Equations	
$\sigma_a(\lambda)$	$= \sigma_{a,d}(\lambda) + \sigma_{a,p}(\lambda) + \sigma_{a,w}(\lambda) + \sigma_{a,y}(\lambda)$
$\sigma_s(\lambda)$	$= \sigma_{s,w}(\lambda) + \sigma_{s,d}(\lambda) + \sigma_{s,p}(\lambda)$
$p(\lambda, \theta)$	$= \frac{\sigma_{s,w}(\lambda)p_w(\lambda, \theta) + \sigma_{s,d}(\lambda)p_d(\lambda, \theta) + \sigma_{s,p}(\lambda)p_p(\lambda, \theta)}{\sigma(\lambda)}$
$\sigma_t(\lambda)$	$= \sigma_a(\lambda) + \sigma_s(\lambda)$
$\sigma_i(\lambda)$	$= \sigma_{i,p}(\lambda') + \sigma_{i,w}(\lambda') + \sigma_{i,y}(\lambda')$
$p_i(\lambda_i, \lambda_o, \theta)$	$= \frac{\sigma_{i,p}(\lambda_i)p_p(\lambda_i, \lambda_o, \theta) + \sigma_{i,w}(\lambda_i)p_w(\lambda_i, \lambda_o, \theta) + \sigma_{i,y}(\lambda_i)p_y(\lambda_i, \lambda_o, \theta)}{\sigma_i(\lambda_i)}$
$f_\lambda(\lambda_i, \lambda_o)$	$= \frac{\sigma_{i,p}(\lambda_i)f_{\lambda,p}(\lambda_i, \lambda_o) + \sigma_{i,w}(\lambda_i)f_{\lambda,w}(\lambda_i, \lambda_o) + \sigma_{i,y}(\lambda_i)f_{\lambda,y}(\lambda_i, \lambda_o)}{\sigma_i(\lambda_i)}$

Table 4.2: The main functions of the model. Coefficients for water (w), detritus (d), phytoplankton (p) and yellow mater (CDOM, y) have been defined on the bio-optical model presented in section 3.5)

paper can be found in Table 4.3. Note that for simplicity we have not included the values that are already specified throughout the text during the explanation of the bio-optical model (more specifically, those included in tables 3.4 and 3.5). The first three correspond to the parameters specifically analyzed in Fig. 5.8.

Once we have formalized the model into a set of equations, we rely on radiative transfer theory to obtain a solution for the in-water light field. We solve the Full Radiative Transfer Equation (2.34) by extending the traditional photon mapping algorithm in a way similar to Gutierrez et al. (GMAS05), but using our bio-optical model and simulating a wider range of effects (including anti Stokes inelastic scattering or different types of absorption and scattering events). The main contribution of the algorithm is applied during the photon tracing stage: in order to account for inelastic scattering events, we include the possibility of an absorbed photon being re-emitted at a different wavelength (with Stokes or anti-Stokes behavior). Photons interact with the medium according to its optical distance, which is a function of the extinction coefficient. At each interaction, we statistically decide which kind of interaction occurs (refer to Table 4.1) with a Russian Roulette algorithm. Whilst Jensen’s algorithm (Jen01) contemplates only an absorption or elastic scattering event as possibilities, we now take into account all ten different events specified in Table 4.1.

For each of the absorption interactions that can potentially generate an inelastic event j (pure water, phytoplankton or CDOM) we define its *inelastic probability* (χ_j), the probability that an absorption event generates an inelastic scattering event:

$$\chi_j(\lambda_i) = \int_{\lambda_a}^{\lambda_b} f_\lambda(\lambda_i, \lambda_o) d\lambda_i \approx \sum_{k=1}^{N_\lambda} f_\lambda(\lambda_k, \lambda_o) \quad (4.3)$$

where λ_a and λ_b are the lower and upper limits of the simulated wavelengths, and $i \in [1..N_\lambda]$ refer to samples in wavelength domain.

4. SIMULATION

Table 4.3: Parameters of the model

Parameter	Equations	Simulated values	Units
C	(3.23) (3.27)	[0..1.0]	$\frac{mg}{m^3}$
$\alpha_d(400)$	(3.24)	[0..0.1]	m^{-1}
$\alpha_y(440)$	(3.22)	[0..0.1]	m^{-1}
S_y	(3.22)	0.014	nm^{-1}
S_d	(3.24)	0.011	nm^{-1}
A_0	(3.33)	$\frac{150}{700}$	nm^{-1}
A_1	(3.33)	4	-
A_2	(3.33)	4	-
B_0	(3.33)	$\frac{1}{450 \cdot 10^{-7}}$	nm^{-1}
B_1	(3.33)	$\frac{1}{650 \cdot 10^{-7}}$	nm^{-1}
Γ_p	(3.31)	0.1	-
Γ_y	(3.33)	-	-
$A_i, i = 1..4$	(3.35)	0.41, 0.39, 0.10, 0.10	-
$\tilde{\nu}_i, i = 1..4$	(3.35)	3250, 3425, 3530, 3625	-
$\Delta\tilde{\nu}_i, i = 1..4$	(3.35)	210, 175, 140, 140	-

In order to apply the Russian Roulette algorithm, we will define an albedo $\Lambda_j(\lambda)$ for each interaction j as follows:

- If the interaction j represents an elastic scattering event, then $\Lambda_j(\lambda) = \frac{\sigma_{s,j}(\lambda)}{\sigma_t(\lambda)}$
- If j represents an absorption interaction that does not show inelastic scattering (detritus and minerals, basically), then $\Lambda_j(\lambda) = \frac{\sigma_{a,j}(\lambda)}{\sigma_t(\lambda)}$
- For each absorption interaction that could generate inelastic scattering:
 - If the interaction j represents the effective inelastic scattering event within the absorption interaction: $\Lambda_j(\lambda) = \frac{\sigma_{i,j}(\lambda)}{\sigma_t(\lambda)} \chi_j(\lambda)$
 - If the interaction j represents the pure absorption event (no inelastic scattering happening at all): $\Lambda_j(\lambda) = \frac{\sigma_{i,j}(\lambda)}{\sigma_t(\lambda)} (1 - \chi_j(\lambda))$

Thus, at each interaction a random number ξ between 0 and 1 is generated so that:

- $\xi \in [0, \xi_1) \rightarrow$ absorption by pure water.
- $\xi \in [\xi_1, \xi_2) \rightarrow$ Raman scattering (inelastic scattering by pure water).
- $\xi \in [\xi_2, \xi_3) \rightarrow$ absorption by minerals and detritus.
- $\xi \in [\xi_3, \xi_4) \rightarrow$ absorption by phytoplankton.
- $\xi \in [\xi_4, \xi_5) \rightarrow$ inelastic scattering by phytoplankton.

4.2 Simulation of inelastic scattering on participating media

- $\xi \in [\xi_5, \xi_6) \rightarrow$ absorption by CDOM.
- $\xi \in [\xi_6, \xi_7) \rightarrow$ inelastic scattering by CDOM.
- $\xi \in [\xi_7, \xi_8) \rightarrow$ elastic scattering by pure water.
- $\xi \in [\xi_8, \xi_9) \rightarrow$ elastic scattering by minerals and detritus.
- $\xi \in [\xi_9, 1] \rightarrow$ elastic scattering by phytoplankton.

where $\xi_i(\lambda)$ is given by:

$$\xi_i(\lambda) = \sum_{j=1}^i \Lambda_j(\lambda) \quad (4.4)$$

To compute the new re-emitted wavelength after a inelastic scattering event j , the normalized wavelength redistribution function $\frac{f_\lambda(\lambda_i, \lambda_o)}{\chi_j(\lambda_i)}$ is treated as a probability distribution function (PDF) given the excitation wavelength λ_i . To sample it efficiently we first build its normalized cumulative distribution function (CDF) and then inverse importance sample this CDF. Greater values of the PDF for a given wavelength will translate to steeper areas of the CDF, thus increasing the probability of a re-emission at such wavelength. Note that the definition of $f_\lambda(\lambda_i, \lambda_o)$ is not limited to the visible spectrum, which might result in re-emissions happening at wavelengths beyond the visible spectrum. However, as $\chi_j(\lambda_i)$ is limited to the simulated (visible) spectrum, only inelastic interactions within this spectrum are considered. It could happen that a photon inelastically scattered at such wavelengths suffers a second inelastic scattering event that brings it back to the visible light range. Given the low probability of this chain of events and our computer graphics approach, we assume that a photon beyond the visible spectrum is definitely absorbed. Fig. 4.5 shows a global overview of the algorithm during the photon tracing stage.

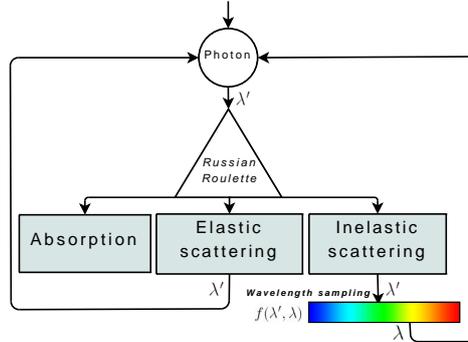


Figure 4.5: Photon tracing algorithm. Inelastic scattering events generate a photon with a different associated wavelength according to the wavelength redistribution function.

To estimate radiance we adopt a trade-off between speed and memory requirements similar to the proposed by Jensen (Jen01): we only store photons in the photon map if they have been reflected or transmitted from surfaces, or if they have already been scattered at least once. Thus, we can compute single scattering more efficiently by ray marching through the medium and sampling the light sources. Taking into account the wavelength redistribution function for inelastic scattering, a new term will be added at each step of the ray marching process:

4. SIMULATION

$$+ \sum_{l=1}^N \sum_{i=1}^{N_\lambda} \left\{ L_l(\lambda'_i, \vec{w}_l) p_I(\lambda'_i, \lambda, \vec{w}_l, \vec{w}_o) \alpha_I(\lambda'_i) f_I(\lambda'_i, \lambda) \Delta x \right\} \quad (4.5)$$

where $i \in [1..N_\lambda]$ and $l \in [1..N]$ refer to samples in the wavelength and light source domain respectively, \vec{w}_l is the direction to the light with an incoming radiance L_l and Δx represent the ray marching steps.

Multiple scattering will be computed from the photon map. To account for multiple *inelastic* scattering we modify the radiance estimate expression by including a new term:

$$+ \sum_{k=1}^n \left\{ p_I(\lambda'_k, \lambda, \vec{w}_k, \vec{w}_o) f_I(\lambda'_k, \lambda, x) \frac{\Delta \Phi_k}{\frac{4}{3} \pi r^2} \right\} \quad (4.6)$$

where r is the radius of the sphere that contains the n closest photons, and k represents each of the stored photons.

Images resulting from the application of this algorithm to the ocean water model presented in section 3.5 can be found on section 5.3.

4.3 Optimized rendering of non-linear media

Several optimization techniques plus a study of the effect of the chosen numerical methods on computation time are presented here. This work has already been published (MGS06; MGS07).

4.3.1 General considerations for optimization techniques

The algorithm in which this research is based (GSMA06), although capable of providing a complete solution for light transport in non-linear media, takes several minutes per frame. Assuming that the medium traversed by light or the scene has certain properties, several optimizations techniques (that will be presented below) can be employed. This results in a significant reduction of the computation time. Also, different numerical methods seem to be better suited on a per-scene basis, depending on the gradients of the index of refraction.

For simplification, the coordinates used in the explanations will be camera coordinates: x and y axis will determine horizontal and vertical in the projection plane, while z axis will determine the perpendicular to the projection plane (see figure 4.6).

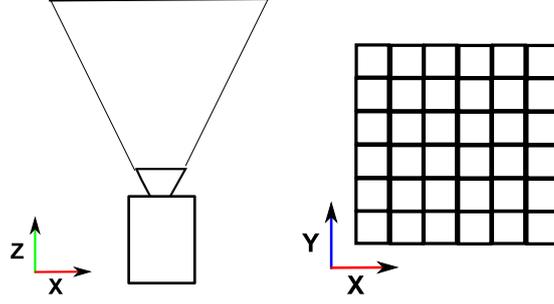


Figure 4.6: Camera coordinates. Left: view from top. Right: view of the projection plane of the camera.

4.3.1.1 Factors of computation time

There are several factors that influence the computation times of the curved ray tracing algorithm. Some of these are related to the standard ray tracing and photon mapping algorithms, while others are specific of the curved ray tracing algorithm (GSMA06).

The factors coming from traditional direct ray tracing, as well as from photon mapping, are:

Number of rays (n_{rays}). The number of rays that will be traced.

Time for intersection ($t_{intersection}$). Average time of intersecting a straight segment with the geometry. Depends mainly on the number of polygons of the geometry and on the structure and the way they are stored and intersected (kd-tree, octree, BSP tree, grid of voxels or instant ray tracing (WK06)).

Number of photons ($n_{photons}$). The number of photons that will be traced from the light sources to the scene.

On the other hand, the factors that are inherent to the introduced curved ray tracing algorithm are:

Number of steps (n_{steps}). The average number of steps required for the numerical method until the intersection with the geometry, depending on the geometry itself, on the chosen numerical method and on the traversed medium.

Time for step advance (t_{step}). Average time required by the numerical method for computing a new step (calculating net point x_{i+1} from net point x_i), that depends only on the chosen numerical method.

Following these factors and this notation, the time required for the curved ray tracing algorithm without any optimization is:

$$t_{total} = n_{rays} \cdot n_{steps} \cdot (t_{intersection} + t_{step}) \quad (4.7)$$

while for photon shooting:

$$t_{total} = n_{photons} \cdot n_{steps} \cdot (t_{intersection} + t_{step}) \quad (4.8)$$

For the different optimization techniques that will be presented in the next sections, 3D transformations and interpolations are required. For completeness we introduce the corresponding symbols for both operations here; the influence of these times will be explained in the corresponding sections:

4. SIMULATION

Linear transformation time ($t_{transform}$). The time required for transforming a 3D point using a linear transform, basically multiplying vector and matrix.

Interpolation time ($t_{interpolation}$). The time required for interpolating (linear interpolation) a 3D point from a set of 3D points.

Specific data structures for geometry (kd-tree, octree, BVH...) or polygon number reduction techniques are techniques that also improve efficiency, as they do on standard ray tracing. However, being already well documented, their effect is out of the scope of this PhD.

4.3.2 Optimization techniques for curved ray tracing

4.3.2.1 Parallelization

Although the presented algorithm is not parallel in (GSMA06), being based on ray tracing an adequate parallelization is straight-forward. The computation of each single curved ray is independent of the computation of the rest of rays, and therefore can take place in a different processing unit. As a consequence, the overhead of rendering on multiple processing units is negligible compared to the improvement on performance. Our system includes parallel computation on a single machine (shared memory).

4.3.2.2 Symmetries

Most atmospheric effects (like mirages) can be seen even when the index of refraction of the medium depends only on the Y axis (vertical), and as a consequence, they present a symmetry in YZ plane. If the index of refraction presents this symmetry then also the path of different light rays will follow this symmetry (trivial deduction from equation 4.2). As a consequence, we can use a linear transform (symmetry) to calculate the different net points $x_0, x_1...x_n$ of the right half of the curved rays from the points of the left half.

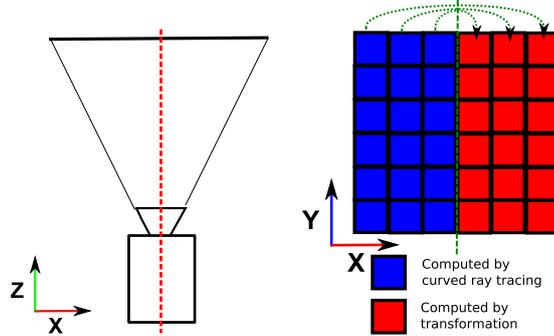


Figure 4.7: Symmetries optimization. Left: Symmetry on the camera YZ plane. Right: pixels whose path can be transformed.

Figure 4.7 (right) shows which paths would be computed and which of them transformed from other paths. Also, XZ symmetry could be considered the same way.

The computing time using this technique is:

$$t_{total} = n_{rays} \cdot n_{steps} \cdot \left(t_{intersection} + \frac{t_{step} + t_{transform}}{2} \right) \quad (4.9)$$

4.3.2.3 Undersampling

Assuming that the gradient of the index of refraction in the medium is locally moderate, the technique of undersampling can be used. Some paths are computed normally, while other paths have their net points $x_0, x_1 \dots x_n$ interpolated from the net points of nearby paths (traversing nearby pixels). As we have assumed moderate local gradients, there are not too different gradients from one path to nearby paths, and as the gradient is what defines the path (see equation 4.2), the interpolated path and the path which would have been computed with the algorithm will not present visible errors.

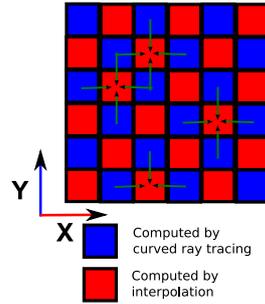


Figure 4.8: Undersampling: the path of some pixels is interpolated from nearby pixels.

Figure 4.8 shows which paths could be interpolated from other paths.

Let α be the rate of interpolated paths compared to the total paths, then the computing time of a frame is:

$$t_{total} = n_{rays} n_{steps} (t_{intersection} + (1 - \alpha) t_{step} + \alpha t_{interpolation}) \quad (4.10)$$

4.3.2.4 Path caching

If the camera is static, whereas lights and geometry are dynamic, the path of each of the primary rays do not change from frame to frame. Taking advantage of this circumstance, it is possible to keep in several linked lists the net points of the paths calculated in one frame and re-use them in following frames, so from frame to frame the only calculations that have to be done are:

- The intersection of each of the segments of the path with the geometry.
- The extra segments that were not calculated in the previous frame because the path intersected on a segment and stopped the calculations.

The structure to save the different paths of light can be as simple as one linked list (or array) per ray with the 3D points $x_0, x_1 \dots x_n$ that are calculated by the numerical method. From one frame to the next one, this structure will remain as is. It is recommended generally using linked lists, as there is no knowledge during compilation time of the number of steps of a single ray. If using also the Smart Step Selection technique, anyway, the structure might be different (see section 4.3.3).

As a consequence, the time required in the first frame remains the same as if no optimization had been done (equation 4.7), and in the best circumstances (no extra segments required) the computing time of the rest of the frames is:

$$t_{total} = n_{rays} \cdot n_{steps} \cdot t_{intersection} \quad (4.11)$$

Notice that in this way we have removed the t_{step} time completely, as the calculations of the step are not required anymore. This technique becomes really useful with complex numerical methods that give an accurate solution in just a few steps, with the trade-off of higher memory requirements.

4. SIMULATION

4.3.3 Smart step selection

After any optimization is performed, the expression given for time is given by the following equation:

$$t_{total} = n_{rays} \cdot n_{steps} \cdot t_{intersection} + C \quad (4.12)$$

where C is a constant that represents the overall time related to the numerical method, that is affected directly by the previous optimizations. If we were able to predict exactly at which step the segment finds its first collision, then we will require just one intersection per ray, so consequently the resulting time will be:

$$t_{total} = n_{rays} \cdot t_{intersection} + C \quad (4.13)$$

that, compared to equation 4.12, it is n_{steps} faster.

In order to make an intelligent prediction of the chosen step, we take into account the following:

spatial vicinity If one of the rays hits a surface in the step n , there is a high probability that the rays corresponding to nearby pixels / solid angles will hit the same surface in the step $n - 1$, n or $n + 1$.

temporal vicinity If one of the rays hits a surface in a frame in the step n , there is a high probability that in the following frame, the ray corresponding to the same pixel / solid angle will hit the same surface in the step $n - 1$, n or $n + 1$.

This way, the prediction can be done by looking at the segment at which either the same ray in the previous frame or the nearby rays have collided. Therefore, if the original ray collided at step n , we first check for intersections in the step n of the new ray, then on the step $n - 1$ and then on the step $n + 1$. If none of them detect a collision, then the intersections are checked as in the original algorithm (starting from the first step). In the best case (step n being the right prediction) the resulting computing time remains as stated in equation 4.13 since each ray requires the computation of a single intersection.

4.3.4 Optimization techniques for curved particle tracing

A physically correct global illumination in non-homogeneous media can be simulated by the curved photon mapping algorithm (GMAS05), thus avoiding tracing shadow rays. The previously presented optimization techniques can also be optimized. In the next subsections we show how to adapt curved ray tracing optimization techniques so they can be applied for curved photon mapping.

4.3.4.1 Parallelization

Photon shooting can be as easily parallelized as ray tracing is. However, it is important to remark that while building the photon map, if two or more threads are inserting photons in the map, it could happen that one thread writes the photon at the last position and before updating the photon counter, so another thread can write a photon in the same position, erasing the information of the previous photon. This means that the structure of the photon map is not thread-safe (on writing). As a consequence, writing on the photon map (adding photons) should be done inside a mutual exclusion section. This means a slight reduction on performance in parallelization. However, querying the photon map for radiance estimation is thread safe, as it does not require writing on it.

4.3.4.2 Symmetry

Outgoing directions from light sources are sampled (either uniformly or using importance sampling) randomly. However, the fact that this sampling is random is not compatible with the symmetry technique (section 4.3.2.2), since it only works if the directions of the primary rays are known.

The symmetry technique can be somewhat adapted by sampling just one of the symmetric hemispheres instead of the whole sphere of directions of the light. For each sampled direction on one hemisphere, two particles are cast: one towards the sampled direction and the other one towards the symmetric direction. One of the two paths is computed using the standard algorithm while the other path is computed as the symmetric of the first one. The application of this technique reduces the time of the first interaction of the photons with the scene to:

$$t_{total} = n_{photons} \cdot n_{steps} \cdot \left(t_{intersection} + \frac{t_{step} + t_{transform}}{2} \right) \quad (4.14)$$

4.3.4.3 Undersampling

Undersampling (section 4.3.2.3) has the same problem with randomness than symmetry (section 4.3.4.2). However, it can be also adapted to particle tracing. First of all, a threshold t parameter is set. This threshold is an angle and is measured in degrees. The second parameter to be set is n , a number which sets the number of paths that are taken into account to interpolate a new path.

When a new outgoing direction (\vec{d}) is randomly sampled, the undersampling algorithm acts in two different ways depending on the new sampled direction:

- If there exist a number of previously stored paths $n_p \geq n$ with direction $\vec{d}_i (i = 1..n_p)$ that fulfill $\arccos(\vec{d}_i \cdot \vec{d}) \leq t$ (closer to \vec{d} than the angle t) then the path of this sample is computed by interpolating it from the n closest stored paths, and is not stored for future interpolations.
- If the previous condition is not met, then the path followed by the photon is computed using the algorithm presented in section 4.1.6. Every point $x_0, x_1...x_n$ computed by the algorithm will be stored, associated to the direction \vec{d} . As a consequence, this path can be used for interpolations afterwards.

Our tests show that $t = 2$ and $n = 3$ are adequate parameters that reduce time while keeping negligible errors. $t > 2$ and $n < 3$ will lead to higher errors and inaccuracies, while $t < 2$ and $n > 3$ will lead to more accurate but much more time consuming simulations.

Let α be the rate of directions that are interpolated. Considering that directions are sampled uniformly:

$$\alpha = \left(\frac{2\pi (1 - \cos \frac{t}{2})}{4\pi} \right)^n = \left(\frac{1 - \cos \frac{t}{2}}{2} \right)^n \quad (4.15)$$

whilst the time required for computing the paths coming from the light sources will be:

$$t_{total} = n_{photons} n_{steps} (t_{intersection} + (1 - \alpha) t_{step} + \alpha t_{interpolation}) \quad (4.16)$$

4. SIMULATION

4.3.4.4 Path caching

Adapting path caching (section 4.3.2.4) to photon shooting is straight forward: instead of storing the paths followed by the rays, the paths followed by the photons that are shot from the light sources are stored. If a fairly big number of photons is required, more paths have to be cached and thus more memory will be used. On the other hand, as it was shown in section 4.3.2.4, this leads to a huge increase of performance on the photon shooting stage.

If light sources are static but the rest of the elements of the geometry are dynamic, this technique becomes very useful. Paths are stored in the first frame and are not calculated again. This technique can be easily combined with temporal coherence photon techniques (TMD+04) which will further increase performance. The time required for computing the paths coming from the light sources is given by:

$$t_{total} = n_{photons} \cdot n_{steps} \cdot t_{intersection} \quad (4.17)$$

It is important to highlight that only the paths of the photons that are shot from the light source are required. The path of secondary photons (after interaction with any surface) might be stored too, but this would obviously require more memory, and if the previous interactions change position in image space, the stored paths should be discarded.

4.3.4.5 Smart step selection

The smart step selection technique (section 4.3.3) can also be applied to particle tracing from the light source. A threshold angle t is set as parameter of the method (depending on the scene, from 2 to 4 degrees will yield optimal results). For each sampled direction, the ordinal number of step in which it has collided is stored. For each new sampled direction \vec{d} , if there exists a previously stored direction \vec{d}_i so that $\arccos(\vec{d}_i \cdot \vec{d}) \leq t$ (closer to \vec{d} than the threshold t), then the step associated to \vec{d}_i is first checked for collisions in the new sample \vec{d} . This way spacial vicinity is considered. In order to consider also temporal vicinity, the same rule is applied, but directions \vec{d}_i from the previous frame are also taken into account.

If there are more than one \vec{d}_i that can be considered for a new sampled direction \vec{d} , then each of them has a priority. First, the ones from spacial vicinity are checked, and then, the ones from temporal vicinity (previous frame). Inside each of the lists (spacial and temporal), first the directions with less ordinal number of step are checked.

In the best case (always guessing the right step) the time needed for computing the paths coming from the light sources is:

$$t_{total} = n_{photons} \cdot t_{intersection} + C \quad (4.18)$$

where C is a constant that depends on the rest of the optimization techniques (see section 4.3.3).

4.3.5 Numerical methods

The chosen numerical method greatly influences computation time and efficiency, depending on the gradient of the index of refraction in the medium. For our simulations we have tested the following methods (BF88):

- Euler method. It is the most simple. Its main advantage is its low computation time, and its main disadvantage is its inaccuracy. It requires a very small step size to be usable.
- Order 2 Runge-Kutta method. More accurate than Euler method.
- Order 4 Runge-Kutta method. More accurate than the order 2 version.
- Dormand Prince method (DP80). Similar to an order 4 Runge-Kutta method, but it uses an adaptative variable step size. Instead of setting the step size as parameter (as the previous methods), it requires a *tolerance*, that is the maximum error that can be committed in a specific step. The step size varies in order to fit this tolerance.

In section 5.4 (results chapter) there is an analysis on the different numerical methods, together with the study of the performance impact by each of the optimization techniques presented before.

4.4 Polarization and interference: The rainbow

The research presented in this section comes from the cooperation with Iman Sadeghi, Philip Laven, Wojciech Jarosz and Henrik Wann Jensen. We first introduce a small insight into the optics of water drops from the simulation point of view, and then we present our algorithm that takes into account polarization and interference for simulating phase functions of water drops in order to get rainbows.

4.4.1 Optics of water drops

4.4.1.1 Spherical water drops

The formation of rainbows can be understood by considering geometric ray paths within the circular cross-section of a spherical drop of water. Figure 4.9 shows a light ray entering the drop of water, suffering two refractions and one internal reflection. This light ray is deviated by angle θ :

$$\theta = 180^\circ + 2i - 4r = 180^\circ + 2i - 4 \arcsin\left(\frac{\sin(i)}{n}\right), \quad (4.19)$$

where n is the relative refractive index defined by $n = n_i/n_t$ where n_i is the refractive index of the sphere (water) and n_t is the refractive index of the surrounding medium (air). As the refractive index of air is about 1.0003, it is convenient to assume that $n_t = 1$, thus making $n = n_i$. Using Equation 4.19, it is simple to calculate how θ varies as a function of impact parameter b for a specific value of n . Figure 4.10 shows such a graph for $n = 1.33257$ in which θ reaches a minimum value at $\theta = 137.86^\circ$, which is known as the geometric rainbow angle $\theta_{rainbow}$. The shape of the curve in

4. SIMULATION

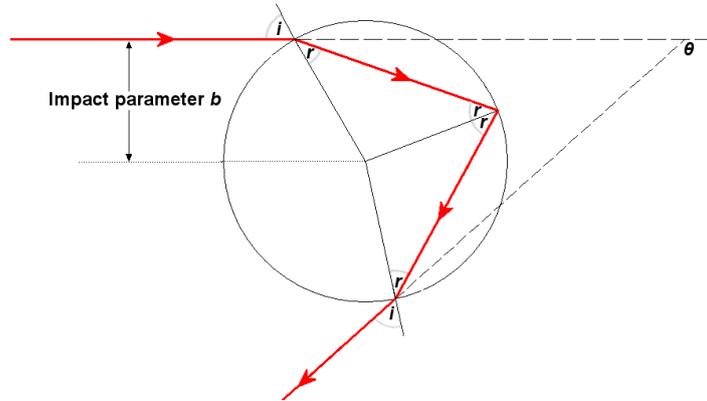


Figure 4.9: Geometric ray paths responsible for the formation of the primary rainbow

Figure 4.10 means that much of the incident light is scattered in directions close to $\theta_{rainbow}$. It is this concentration of scattered light that makes the rainbow visible. Note that two geometric rays contribute to the scattering when $\theta > \theta_{rainbow}$, but no geometric rays can contribute to the scattering when $\theta < \theta_{rainbow}$. Hence, $\theta_{rainbow}$ marks the boundary between 2-ray scattering and no scattering. $\theta_{rainbow}$ can be defined solely as a function of n , as shown in Equation 4.20.

$$\theta_{rainbow} = 2 \arcsin \left(\frac{1}{n^2} \left(\frac{4 - n^2}{3} \right)^{\frac{3}{2}} \right) \quad (4.20)$$

The refractive index of water varies between 1.34451 at $\lambda = 400 \text{ nm}$ (corresponding to violet light) and 1.33141 at $\lambda = 700 \text{ nm}$ (red light). Applying these values of n to Equation 4.20 shows that $\theta_{rainbow}$ varies with wavelength from 139.57° at $\lambda = 400 \text{ nm}$ to 137.69° at $\lambda = 700 \text{ nm}$. This dependence of $\theta_{rainbow}$ on wavelength is responsible for the colors of the rainbow (changing from red on the inside of the arc through to violet on the outside of the arc) - as well as determining that the rainbow has a width of about 2° .

Light rays that undergo one internal reflection in the water drop produce the primary rainbow. Rainbows can also be generated by light rays that have suffered two or more internal reflections: in the case of two internal reflections, $\theta_{rainbow}$ for the resulting secondary rainbow for varies between 129.5° for red light and 126.1° for violet light. Note the colors of the secondary rainbow (red on the inside of the arc and violet on the outside) are reversed compared with the primary rainbow.

Although geometrical optics can provide a good explanation of the formation of the primary and secondary rainbows, real rainbows exhibit some features that are not easily understood. For example, additional arcs (known as supernumerary arcs) occasionally appear on the inside of the primary rainbow: such arcs are typically violet or blue. Supernumerary arcs caused great consternation because they were not predicted by geometrical optics. However, in the 1830s, scientists such as Young and Airy realized that they were a consequence of the wave theory of light (Air38). As mentioned above, two geometrical rays contribute to scattering when $\theta > \theta_{rainbow}$. As the two rays have different path lengths, they have different phases - which result in an interference pattern consisting of a series of maxima and minima as a function of θ . In practice, the phase difference between the two rays is determined not just by the difference in physical path length, but also by phase changes due to reflection and due to the crossing of focal lines (or caustics).

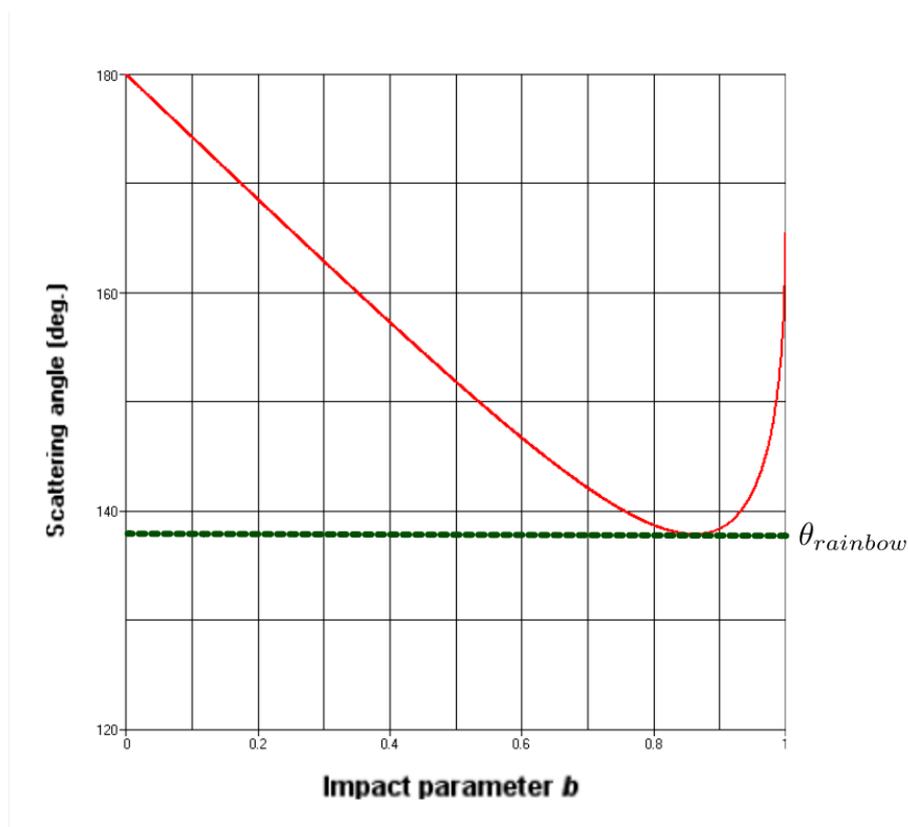


Figure 4.10: Variation of θ as a function of the impact parameter b for $n = 1.33257$ for rays that undergo one internal reflection (as shown in Figure 4.9)

The method of Airy is a mathematical solution for these phenomena, although it is not completely accurate. On the other hand, Young's work fails to consider the phase difference owed to the crossing of focal lines. Fortunately, Mie (Mie08) developed a rigorous solution by deduction from Maxwell's equations (see section 2.1) to the problem of scattering of light from homogeneous spheres, even considering phenomena such as surface waves or diffraction (Cow08). There has been plenty of research around this work (Lavb), and there is even some free software that is able to simulate phase functions based on Mie theory and other methods, named MiePlot (Lava).

4.4.1.2 Non-spherical water drops

However, raindrops are not perfect spheres, and a large amount of research on the exact shape of raindrops exists, mainly in the field of atmospheric sciences (BC87; BKI91; BCX91). As they fall, the underside of the drops tends to become flattened, and surface tension is not sufficient to maintain the spherical shape. This effect is more pronounced the larger the raindrop: the axial ratio falls from about 0.98 at diameters $d = 1\text{mm}$ to about 0.7 at $d = 5\text{mm}$ (BCX91).

In our model, we follow the approach of Beard and Chuang (BC87), who take into account surface tension, hydrostatic pressure and aerodynamic pressure. The influence of hydrodynamic pressure

4. SIMULATION

	1	2	3	4	5	6
a	0.00050	0.001025	0.001555	0.002135	0.00275	0.0034
b	0.00048	0.00093	0.001308	0.00165	0.00194	0.00220
k	0.01	0.15	0.20	0.27	0.29	0.32

Table 4.4: Parameters a, b, k used in Equation 4.21 for equivalent volume spheres of diameters $d = 1$ through $d = 6$ mm.

from internal circulation is argued to be very small and thus left out of their model. They propose a cosine series fit to the model, with the shape of the raindrop profile given by the polar curve $r = d/2[1 + \sum c_n \cos(n\theta)]$, where d is the diameter of the equivalent volume sphere, while the coefficients c_n are listed as tabled values (BC87).

For the purpose of ray tracing, we approximate such shape by deforming an ellipsoid (defined by (a, b)) according to the following implicit equation:

$$\frac{x^2}{a^2} \left(1 - \frac{k}{b} y\right) + \frac{y^2}{b^2} + \frac{z^2}{a^2} \left(1 - \frac{k}{b} y\right) = 0 \quad (4.21)$$

where $k \in [0..1]$ measures the degree of deformation (with $k = 0$ indicating a perfect ellipsoid). Figure 4.11 shows the resulting shapes for different diameters of the equivalent volume sphere, along with a direct comparison with the shapes obtained by Beard and Chuang (BC87). The use of an implicit equation enables efficient ray tracing, although a triangle mesh could potentially be used. Visual inspection shows that there is a very close match between the profiles. The values used for each shape are listed in Table 4.4.

Notice that, while Mie's solution works perfectly for spherical water drops, different water drop shapes cannot be simulated. As we have seen, water drops with diameter less than 1 mm are practically spherical, and as a consequence Mie theory gives adequate results. On the other hand, the ray-tracing method presented in section 4.4.2 can simulate interference or even focal lines for any particle shape, although it cannot simulate surface waves nor diffraction. However, the effect of surface waves and diffraction is imperceptible for water drops bigger than 1 mm diameter, and as a consequence our ray-tracing method is a right option for bigger water drop sizes. We have performed several validation tests comparing the results of our phase function simulation method against the results of other methods implemented in MiePlot (Lava). They can be found on section 5.5.1.

4.4.2 Light transport

In order to raytrace rainbows we need to take into account polarization and interference, for which the traditional representation of light needs to be extended. As light is an electromagnetic wave (see section 2.1.1) we can characterize light in terms of an electric field \vec{E} , perpendicular to the direction of the ray, whose polarization is given by two orthogonal components. If we define a coordinate system in which the propagation direction of the light ray travels along the z axis, we get the following equations:

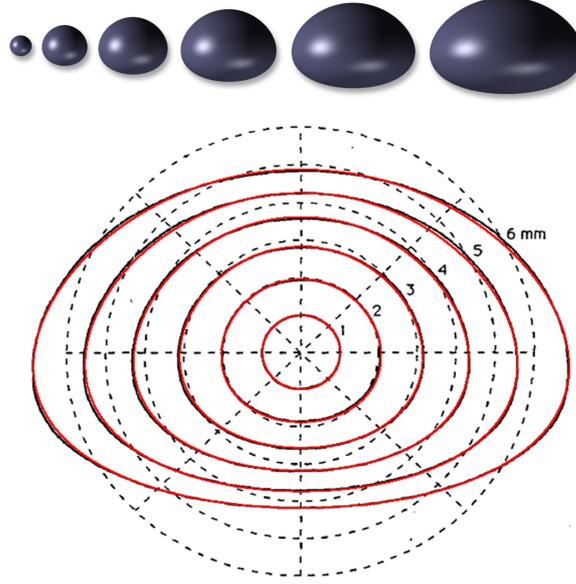


Figure 4.11: Top: non-spherical raindrop shape with increasing radius. Bottom: comparison with the shapes proposed by Beard and Chuang (BC87). The red profiles are the result of our method described by Equation 4.21, while the dotted lines show the equivalent volume spheres.

$$\begin{aligned}
 E_x &= A_x \cos\left(\frac{2\pi}{\lambda}z - \omega t + \delta_x\right) \\
 E_y &= A_y \cos\left(\frac{2\pi}{\lambda}z - \omega t + \delta_y\right)
 \end{aligned}
 \tag{4.22}$$

where A_x and A_y are the amplitudes, λ is the wavelength, z refers to the traversed path, ω is the angular frequency, t is time and δ_x and δ_y represent phase offsets. As ω is constant for each wavelength and as we are considering an stationary simulation of the phase function of the rainbow, the the factor ωt is a constant that will be mathematically simplified. For the sake of clarity, this addend will be ignored in the rest of the text. Notice that by ignoring it both E_x and E_y can be expressed in phasor notation (Gia89).

This representation presents one degree of freedom: given the fact that the z axis is fixed to the light propagation direction there are infinite coordinate systems (rotating along that axis) in which light can be represented. The definition of E_x and E_y depends on the chosen coordinate system. However, changing the coordinate system from (x, y) to another one (x', y') is a trivial operation:

$$\begin{aligned}
 E_{x'} &= \cos\phi E_x + \sin\phi E_y \\
 E_{y'} &= -\sin\phi E_x + \cos\phi E_y
 \end{aligned}
 \tag{4.23}$$

where ϕ is the angle of rotation between the two coordinate systems, as it can be seen in figure 4.12.

As a consequence, we can have an arbitrary coordinate system for the different rays, and rotate it as needed.

4. SIMULATION

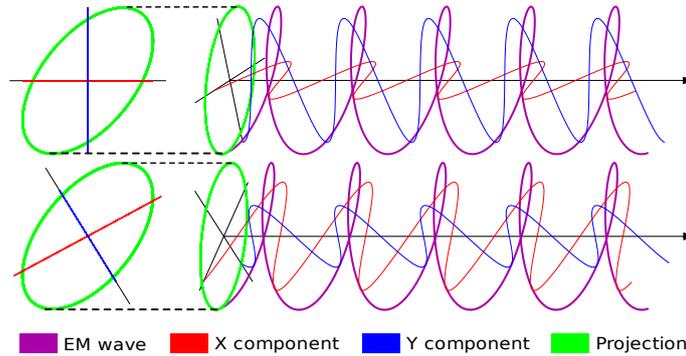


Figure 4.12: Rotating electromagnetic wave coordinate system. Up: before rotation. Down: after rotation. Notice that, even the components change, the wave and its corresponding projection remains the same.

4.4.2.1 Ray-particle interactions

When light interacts with a dielectric-dielectric interface (such as air-water interface), it gets both reflected and refracted. For the simulation of this phenomena, two new rays are generated: one in the reflection direction (incident direction symmetric to the normal of the surface) and one in the refraction direction (according to Snell's law). The amplitudes of the light ray are split between the transmitted and reflected ray in terms of the parallel and perpendicular components with respect to the plane of incidence (see figure 4.13).

When a ray is traversing space, its wave components orientation is defined biunivocally by a base coordinate system. Any defined orientation function works, assumed that each propagation direction results always into the same base orientation coordinate system. The light components coordinate system of the incident ray must be rotated as in 4.4.3.2 to the coordinate system defined by the vector perpendicular to the plane of incidence and the vector contained in the plane of incidence. The respective amplitudes in then multiplied by the Fresnel coefficients $t_{\parallel}, t_{\perp}, r_{\parallel}$ and r_{\perp} , which can be found in most optics books (LLT95). The outgoing rays are then rotated back to the base rotation coordinate system.

The phase of the different components can change at the interactions under two circumstances:

- Multiplying a phasor and a negative Fresnel coefficient is equivalent to a phase shift of π (van57).
- In the case of a total internal reflection, there exist some phase shifts δ_{\parallel} and δ_{\perp} for the reflected fields parallel and perpendicular to the plane of incidence (GMN94) (see section 2.2.3).

Note that there is no phase change after refraction.

The interactions between the ray and the particle define the polarization state of the light. However, these Fresnel interactions cannot be computed using Stokes parameters (typically used for polarization computations), because the Stokes vector only considers the phase difference between the two components of the light. As we need to compute interference between the light rays that exit the particle, we need absolute phase computations in both components. Apart from Fresnel interactions, the phase of the rays is affected the traversed optical path and the traversed focal lines.

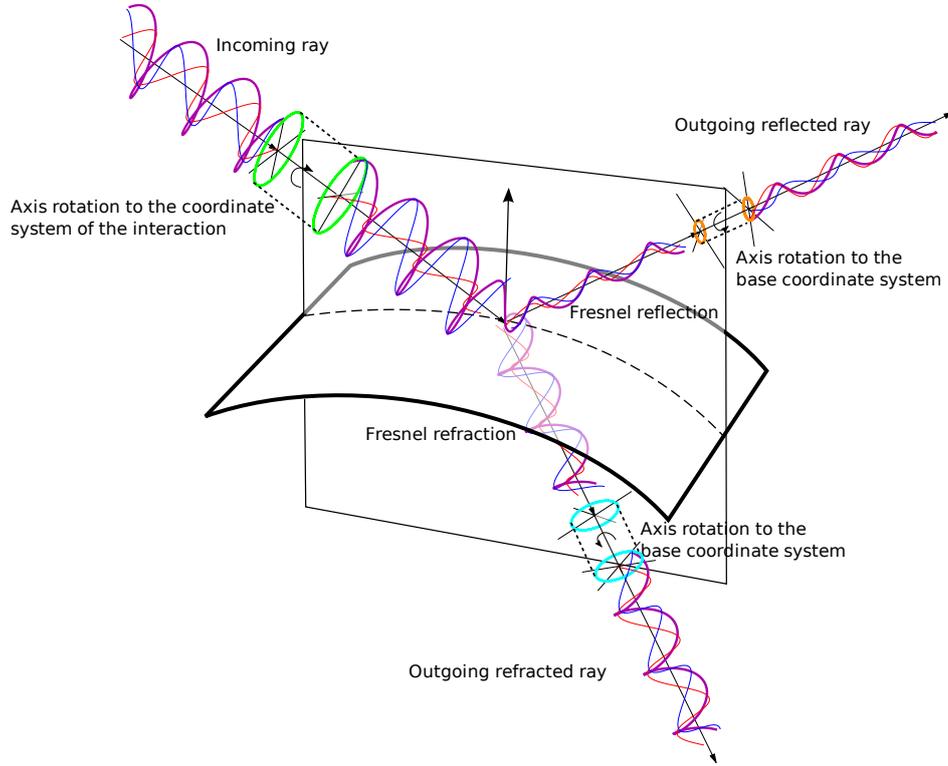


Figure 4.13: Ray-particle interaction. First, the components of the incoming wave are rotated so they become parallel and perpendicular to the interaction plane. Then both reflection and refraction are computed according to Fresnel equations (including phase shifts). After, both for the reflected and the refracted wave, another rotation is done in order to get back to the base rotation coordinate system.

4.4.2.2 Optical path

The optical path l that is traversed by the ray is defined as follows:

$$l = \int_P n ds \quad (4.24)$$

where n is the index of refraction at a specific point of the path P , and s refers to the differential traversed path.

For the simulation just two different indexes of refraction are considered: the one of the medium n_i (air in our case) and the one of the particle n_t (water in our case). This means that the total optical path traversed by a ray is

$$l = n_i s_i + n_t s_t \quad (4.25)$$

where s_t is the total distance that the ray travels inside the particle (easy to compute using ray tracing) and s_i is the distance traveled by light outside the particle. However, the distance from the sun related to the size of the particle, plus the fact that what happens to the ray when it exits the particle is unknown makes s_i difficult to compute. Fortunately, we need just relative distances

4. SIMULATION

between different rays, so we set up a common reference plane for all the rays that are cast towards the particle that is perpendicular to their direction. Also, the same is done for outgoing rays: we set a plane perpendicular to each outgoing ray that is at a certain distance from the center of the particle (see figure 4.14).

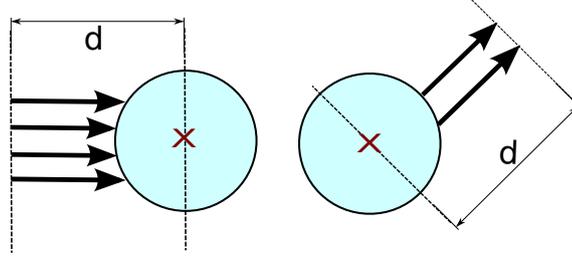


Figure 4.14: Setting up planes for incoming (left) and outgoing rays (right) in order at the same distance d to compute the optical path outside the particle s_i

Including the optical path into Equation 4.22 we get that:

$$E_i = A_i \cos\left(\frac{2\pi}{\lambda}(z+l) + \delta_i\right) = A_i \cos\left(\frac{2\pi}{\lambda}z + \left(\frac{2\pi}{\lambda}l + \delta_i\right)\right) \quad (4.26)$$

where i refers to both channels x and y and l is the traversed optical distance. Thus, the phase shift due to optical path length is $2\pi\lambda^{-1}l$.

4.4.2.3 Focal lines

Also, the wavefronts that enter the particle are not completely plain, and therefore they concentrate into focal lines (see figure 4.15). Each passage through a focal line along the path yields a phase advance of $\pi/2$ (van57). However, detecting three dimensional focal lines under a generic geometry is a very daunting task, plus computationally very expensive. Even more, detecting which rays are traversing a focal line (that is a generic three dimensional curve) in order to account for the phase shift is even more complex.

Fortunately, for the simulation of correct rainbows, there is no need to consider all the focal lines affecting the whole phase function, but just the ones that affect the area close to the rainbow. The only computation needed for the right simulation of the rainbow is to detect the phase difference between the two wavefronts that affect the rainbow, so therefore just the difference on the number of focal lines is required. For this, the sign of the derivative of the outgoing angle (θ) with respect to the impact parameter (b) is analyzed (see figure 4.9). When the derivative is positive, one focal line is added. As a consequence, the phase difference from focal lines in the rainbow area is achieved (Lavb).

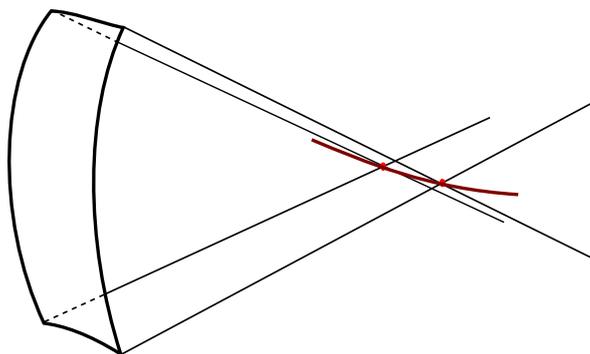


Figure 4.15: A focal line is the curve that contains all the differential focal points given by a wavefront interacting with a surface (either reflection or refraction)

4.4.3 Ray casting

Once defined how to simulate the interactions between light and the particle itself we are able to compute the phase function for a water drop. How to do this? Intuitively, we could use a Monte-Carlo based approach: cast rays by importance sampling the light source, compute the interactions with the particle and store the outgoing rays with their corresponding directions in a sphere.

However, this simple approach would fail. As all the interactions with the particle are perfectly specular or exact refractions, there is no way to sample any of the ray tracing steps in such a way that the outgoing directions are controlled (no shadow rays and nothing such as a diffuse BRDF). This problem is very similar to the problem of rendering caustics in a 3D scene. Nevertheless, a density estimate technique could be used as in photon mapping. The outgoing rays could be stored in some kind of structure such as a kd-tree and then in a second stage the outgoing radiance values from each direction for the phase function could be estimated from the closest stored outgoing rays. Interference between different wavefronts could be computed also in this stage.

Notice that at this stage interference between different wavefronts is very relevant, and it should be simulated. Two different wavefronts represented by rays that have followed different paths results into interference light patterns such as the supernumerary arcs of the rainbow, that are owed to the constructive or destructive interference at each differential solid angle around the particle (see figure 4.16). The density estimate solution does not work for simulating interference (it was implemented and tested). There are several problems with density estimate:

- From the closest outgoing rays, interference must be computed. However the following question arises: which rays should be interfered with which rays? We cannot compute interference from all of them, because it might happen that more than one ray are representing the same wavefront, yielding wrong results when they interference is computed between them. Furthermore, it is not possible to identify the wavefront that is represented by an outgoing ray.
- For a specific direction, a wavefront that is represented by a low density set of outgoing rays is probably yielding some error in the phase. This is increased by the fact that the variation of phase due to optical path is very high, specially as the particle gets bigger.

4. SIMULATION

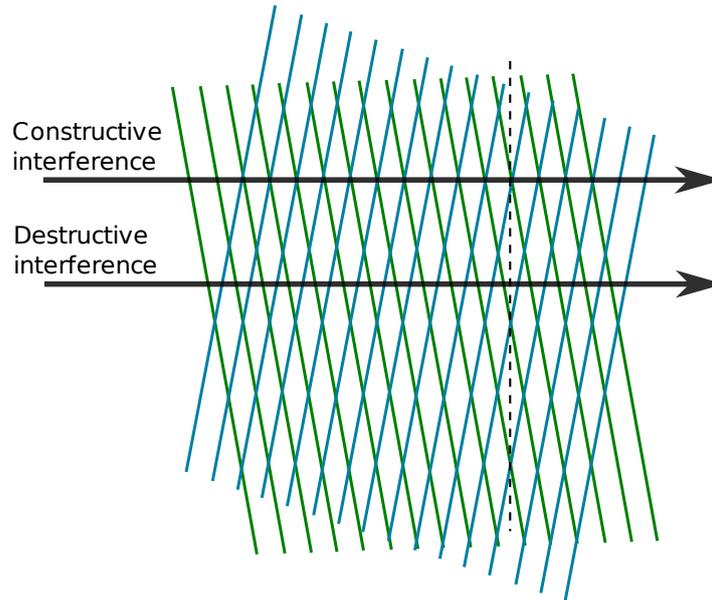


Figure 4.16: Interference between two wavefronts that are orientated differently (blue and green). The blue and green lines are marking the area of the wavefront in which the phase is 0. When they are in phase (intersection of the lines, equal phases, it can be checked at the dotted line), there is constructive interference, and when their phases have a π phase difference, there is destructive interference (as it would be perceived from the left of the figure)

The conclusion is that density estimate techniques fail to give an adequate solution due to the impossibility to know which ray is representing which wavefront, thus deducing an accurate phase from an unordered set of rays is not possible. Our solution overcomes the limitation of density estimation by getting the corresponding ray at a specific direction by interpolating from the "neighbor" rays. This means that it is needed to know which neighbors do each outgoing ray have. This is the reason why when casting rays we also generate a graph in which knowing the neighbor of a ray is a straight forward operation. Furthermore, as it can be seen below, this neighbor knowledge enables to identify whether two close rays belong to the same wavefront, successfully overcoming the limitations of the density estimate issue.

4.4.3.1 Casting rays

Rays are cast from a collimated light source towards the particle. In order to be able to interpolate the needed data from contiguous stored outgoing rays, we need them to be structured in such a way that:

- Rays that are representing the same wavefront are identified.
- At each wavefront, the outgoing stored rays from which the data is interpolated are identified.

4.4 Polarization and interference: The rainbow

A grid seems to be a sensible way to cast the rays, as there are obvious adjacency connections in a grid. All the starting points of the rays are situated in the same plane (the projection plane, perpendicular to the direction of the collimated light source) forming a grid, while the propagation direction of those rays is the direction of the collimated light source. However, our preliminary tests show that a uniform grid presents an important shortcoming: the simulations require more density of samples in areas of the projection of the particle that are close to the edge while not so many in areas close to its center. We therefore distribute the samples according to the following formulation:

$$u' = \alpha u + (1 - \alpha) \sin\left(\frac{\pi}{2} u\right) \quad (4.27)$$

$$v' = \alpha v + (1 - \alpha) \sin\left(\frac{\pi}{2} v\right) \quad (4.28)$$

where $u \in [-1, 1]$ and $v \in [-1, 1]$ are the parameters to be sampled uniformly in order to get the rays, $u' \in [-1, 1]$ and $v' \in [-1, 1]$ are the coordinates of the point of the projection plane (with its center being at $(0, 0)$ and α is a parameter that defines how uniform the grid is. $\alpha = 1$ means that the grid is completely uniform while $\alpha = 0$ means that the grid is distributed in such a way in which it prioritizes the edges of the projection plane. Our tests show that values around $\alpha = 0.3$ give an efficient distribution of rays.

Even using the previous non-uniform grid distribution, there is lack of information in the projected areas close to the projected edge of the particle. In order to cover all the projected area of the particle, rays must be traced towards the edge of the particle. This is done by finding the projected edge of the particle using dicotomic search and casting the corresponding ray towards it. All these can be seen in figure 4.17.

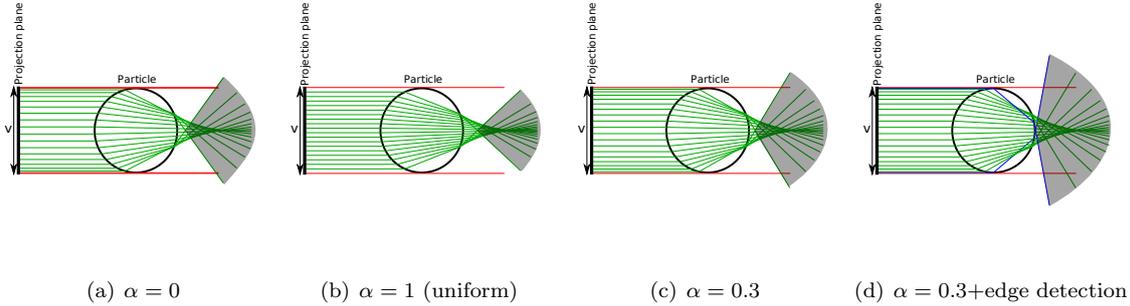


Figure 4.17: Different ray casting distributions. Paths shown follow two refractions on the particle. Green lines represent the rays that hit the particle, red lines represent the rays that miss the particle and blue lines represent the rays that are cast towards the projected edge of the particle. The gray area marks the angle covered by the outgoing rays (by shooting infinite rays it would become 180°). Notice that the outgoing rays cover more angle (and therefore more information) on the rightmost figure.

The typical resolution of this grid for an average quality result is around 1000×1000 rays (per wavelength).

Instead of casting directly beams of light under a specific polarization state, we carry the information of two different polarization states of light per ray: one polarized vertically and one polarized horizontally. When evaluating the outgoing radiance information, any polarization state (or even unpolarized light) can be computed by combining these two. Each ray will carry the following information:

4. SIMULATION

- Two phasors that represent the two components of the horizontal polarization state along the interactions (amplitudes are $A_x = 1$, $A_y = 0$, phases are 0).
- Two phasors that represent the two components of the vertical polarization state along the interactions (amplitudes are $A_x = 0$, $A_y = 1$, phases are 0).
- The traversed optical path (a float number, that is separated from the phases of the phasors so it can be interpolated independently).

The information regarding the number of light bounces p is also used, meaning $p = 0$ one reflection, $p = 1$ two refractions, $p = 2$ two refractions with an internal reflection and so on (see figure 4.18). We limit the simulation up to bounce $p = 3$, given the fact that $p = 2$ gives the primary rainbow and $p = 3$ gives the secondary rainbow. This is a good compromise between accuracy and computation time.

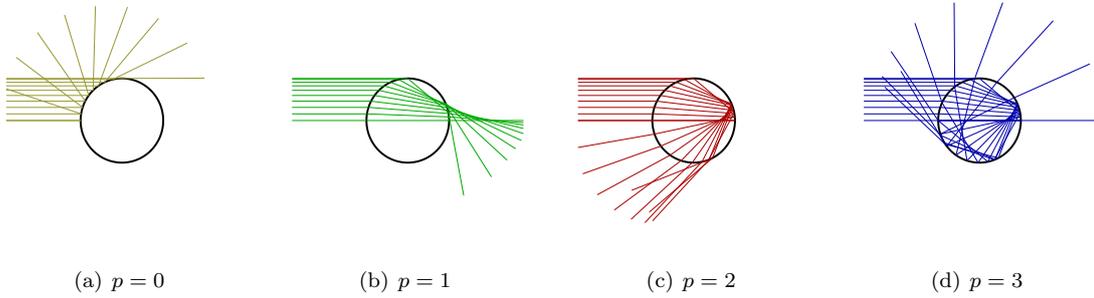


Figure 4.18: Explanation of the different bounces p . $p = 2$ generates the primary rainbow and $p = 3$ generates the secondary rainbow. Just the half of the rays have been shown for the sake of clarity.

Also, as the grid is not uniform, the cast phasor intensities must be normalized according to the area in the projection plane that is covered by the ray. As the intensity of the phasor is $I = A_x^2 + A_y^2$, the amplitudes are normalized according to the square root of this area. The area covered by the ray is computed by averaging the areas of the four (or less) regions that contain the ray. Furthermore, in order to correctly estimate the outgoing radiance we need to divide the intensity of each ray by the corresponding solid angle represented by it. For computing the solid angle covered by the ray we average the solid angles of the patches that contain that ray (analogously to the area of the incident rays, see figure 4.19). Formally, this results into the following equation

$$A'_x = A_x \frac{\sqrt{\frac{\sum_{i=1}^n a_i}{n}}}{\sum_{j=1}^m a_j} \frac{1}{\sqrt{\frac{\sum_{i=1}^n s_i}{n}}} \quad (4.29)$$

where n is the number of cells that contain the ray (usually four, or less if the ray is cast towards the edge of the particle), A'_x is the normalized amplitude for that phasor, a_i are the corresponding areas for each of the area regions contained by the ray, s_i are the corresponding solid angles of the patches contained by the ray, m is the total number of cells and a_j are the areas for all the area regions of the simulation. The analogous exists for the y axis. This has to be applied both for the horizontal and vertical polarization phasors. This equation leads to the correct normalization of the irradiance and the outgoing radiance.

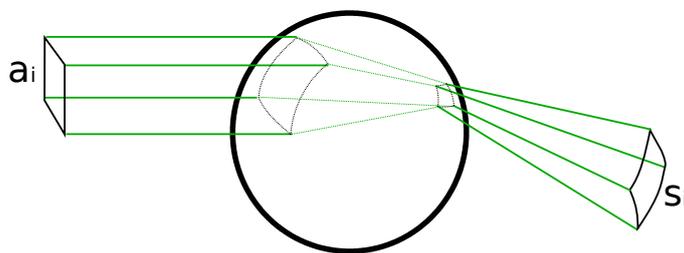


Figure 4.19: Four adjacent rays define a region of area a_i in the projection plane (when they are cast) and a patch of solid angle s_i (when they exit the particle).

4.4.3.2 Collecting sphere

When the ray exits the particle, a final axis rotation is performed (as in Equation) to the two axis that follow the variations of the two angles (θ and ϕ) that parametrize the outgoing directions. Although this would not be required for a plain simulation of the rainbow (just the intensity needs to be considered), global illumination algorithms that consider polarization and use the resulting phase function require standardized outgoing axis coordinates.

After casting the rays and letting the rays interact, outgoing rays are stored with their corresponding adjacentness relationships (as explained in previous section, we are casting in a grid distribution). All these compose a big set of four corner patches on a sphere (as in figure 4.19), in which each of the corners is an outgoing ray.

When querying the outgoing radiance (the two phasors for the two components of the electromagnetic wave) at a specific direction that lies inside the solid angle contained by one of the patches, it is computed by interpolating (using bilinear interpolation) the data of the rays at the four corners of the patch. Then, interference is computed by combining the phasor components that have been interpolated at all the patches that contain the corresponding direction (see figure 4.20).

However, finding all the patches that contain the specific direction that we are querying for computing the phase function is a very time consuming task. An average simulation may result into up to several millions of patches, so checking each of them would be unpractical and inefficient. However, if we interpret the patches as a virtual geometry along the sphere, we can check that an outgoing direction is inside the solid angle covered by a patch by tracing a ray from $(0,0,0)$ towards that corresponding direction. Intersected patches, therefore, contain this outgoing direction. This enables the use of a ray-tracing acceleration structure for storing the patches and make this search much more efficient. In our work, we use a bounding volume hierarchy (GS87) (using axis aligned bounding boxes and constructing it using a surface area heuristic) for this search, with a modification so instead of looking for the closest intersection, it looks for all the intersections (because we actually need to check all the patches that contain the corresponding direction).

As each direction query is independent from the others, the code is easily parallelized. We use multiple threads for computing all the queries needed for tabulating the phase function. Even parallelizing and applying the acceleration technique, searching for all the patches that contain each of the directions in which the phase function is sampled is still the bottleneck of the algorithm (30 minutes per wavelength at 1000x1000 casting grid resolution).

4. SIMULATION

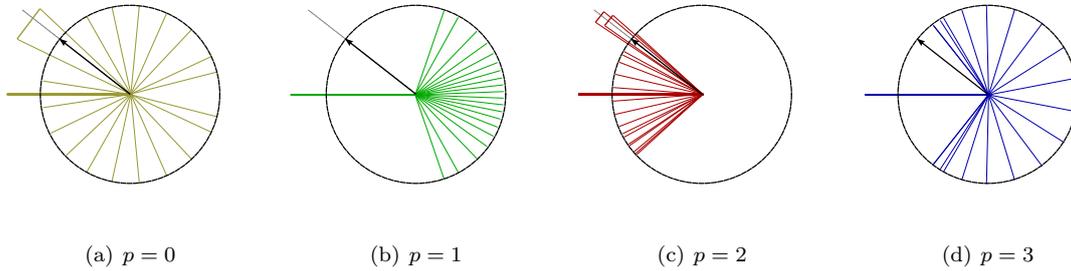


Figure 4.20: Querying the collecting spheres for the patches that contain a specific direction. Notice that, while none of the outgoing patches from $p = 1$ and $p = 3$ contains the specific direction, one patch from $p = 0$ and two patches of $p = 2$ contain it. This is the reason why an adjacentness relationship between the different casted rays is used: otherwise, we would never identify the two patches at $p = 2$ that should interfere. The different bounces have been separated for clarity purposes. In the algorithm this separation is unnecessary as all the patches that contain the ray are searched, independently of their corresponding p .

4.4.3.3 Interpolation and interference

The radiance value at a specific direction is obtained by combining and evaluating the result from interpolating the outgoing rays at the corners of each traversed patch (see figure 4.21). At each of the patches bilinear interpolation is used. At the specific outgoing direction, the interpolated values are the total traversed optical path, the outgoing phasor components for vertical polarization and the outgoing phasor components for horizontal polarization (in total, a real value and four phasors).

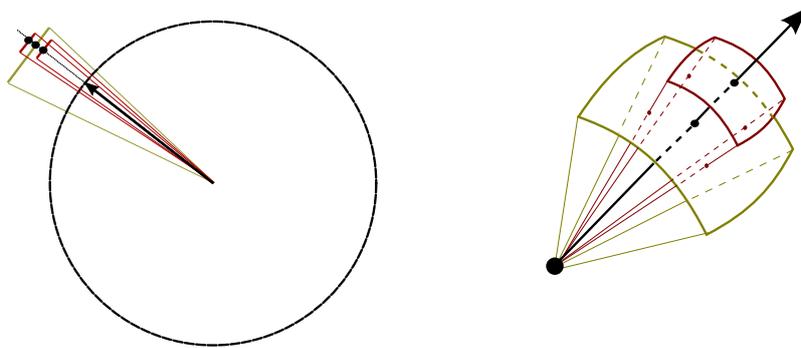


Figure 4.21: At each of the intersected patches (see figure 4.20), bilinear interpolation gives the specific data at that direction (black dots). Then, interference is applied between the interpolated value from each of the patches. Left: 2D representation, right: Simplified 3D representation.

Although bilinearly interpolating a real value is straight forward, a specific representation has to

be chosen for interpolating phasors. Phasors have the following form:

$$A \cos \left(\frac{2\pi}{\lambda} t + \delta \right) \quad (4.30)$$

where A is the amplitude and δ is the phase. This phasor can therefore be represented by a two coordinate vector (A, δ) so each of the coordinates can be interpolated independently. This is called *polar representation* of the phasor. However, this representation shows a singularity when $A = 0$. Also, when interpolating the phase δ the sharp phase shifts of π that happen at specific angles in Fresnel reflections (which are very relevant for the resulting phase function) are smoothed. If the two phasors to be interpolated have $\delta = 0$ and $\delta = \pi$, respectively, the interpolated values go from 0 to π , and the sharp phase shift that should happen at a specific angle disappears.

However, cartesian representation can also be used for phasors. The two vector components in cartesian representation are (x, y) where $x = A \cos(\delta)$ and $y = A \sin(\delta)$. These two components can also be interpolated independently. Phasor arithmetic (addition and multiplication) can be done by considering the phasor to be a complex number $x + iy$. Furthermore, a phase shift of ϕ can be computed by multiplying the complex number by $\cos(\phi) + i \sin(\phi)$ (we note this phasor from now on as $S(\phi)$). Apart from simplifying the implementation and better representing phase shifts, this arithmetic also speeds up computation. Therefore, cartesian representation is used.

The interpolation results into four phasors representing the two components for the vertical and horizontal polarizations, plus an optical path computed at the exact direction being queried. We note the four resulting phasors as $E_{h_x k}$, $E_{h_y k}$, $E_{v_x k}$ and $E_{v_y k}$ where the h and v subindices refer respectively to the horizontal and vertical polarizations, x and y refer to the two components of the electromagnetic radiation and k represents the patch at which the phasors have been computed. Optical path is noted as l_i . Once these are computed for all the patches, interference is computed by adding the corresponding phasors and considering with their respective optical paths:

$$\begin{aligned} E_{h_x} &= \sum_{kP} E_{h_x k} S \left(\frac{2\pi}{\lambda} l_k \right) S \left(F_k \frac{\pi}{2} \right) \\ E_{h_y} &= \sum_{kP} E_{h_y k} S \left(\frac{2\pi}{\lambda} l_k \right) S \left(F_k \frac{\pi}{2} \right) \\ E_{v_x} &= \sum_{kP} E_{v_x k} S \left(\frac{2\pi}{\lambda} l_k \right) S \left(F_k \frac{\pi}{2} \right) \\ E_{v_y} &= \sum_{kP} E_{v_y k} S \left(\frac{2\pi}{\lambda} l_k \right) S \left(F_k \frac{\pi}{2} \right) \end{aligned} \quad (4.31)$$

where $E_{h_x}, E_{h_y}, E_{v_x}$ and E_{v_y} represent the phasors that result from the interference and the inclusion, $S \left(\frac{2\pi}{\lambda} l_i \right)$ represents the phase shift due to the optical path as in Equation 4.26 and $S \left(F_i \frac{\pi}{2} \right)$ is the phase shift due to focal lines as explained in section 4.4.2.3.

The advantage of having these four phasors for each sampled direction is that now it is possible to consider any polarization for incoming light. All these data can be stored and used afterwards in order to do simulations under different incoming light polarizations. Given I_x and I_y the components of the incoming light (phasors) with the x following the horizontal axis and y following the vertical axis we can compute the final outgoing radiance as:

$$L = |L_x|^2 + |L_y|^2 \quad (4.32)$$

$$L_x = I_x E_{h_x} + I_y E_{v_x} \quad (4.33)$$

$$L_y = I_x E_{h_y} + I_y E_{v_y} \quad (4.34)$$

Results and validation of this algorithm can be found in section 5.5.

4. SIMULATION

4.5 Summary

This chapter has introduced the basics of simulation of light transport, along with several contributions of this PhD.

The first of the presented contributions is related to the bio-optical model introduced in section 3.5. An algorithm that is able to simulate this bio-optical model, including the complexity of simulating inelastic scattering (both single and multiple scattering), is presented in section 4.2. Results are shown in section 5.3. The result of this research has been published in the Computer Graphics Forum ([GSMA08](#)).

Next contribution is an efficient algorithm that is able to simulate the light transport in non-linear media, in which the index of refraction varies at every differential point in the volume. This contribution presents a set of optimization techniques and an study of the effect of the chosen numerical method on the simulation time. It is introduced in section 4.3 and the corresponding results are in section 5.4. This has appeared in two different publications in different development stages ([MGS06](#); [MGS07](#)).

The last contribution presented in this chapter is related to the simulation of rainbows from the very basic fundamental phenomena owed to electromagnetic radiation (such as polarization and interference). It is presented in section 4.4 with results in section 5.5.

References

- [Air38] G.B. Airy, *On the intensity of light in the neighbourhood of a caustic*, Trans. Cambridge Philos. Soc. (1838), no. 6, 379–403. [82](#)
- [BC87] Kenneth V. Beard and Catherine Chuang, *A new model for the equilibrium shape of raindrops*, Journal of the Atmospheric Sciences **44** (1987), no. 11, 1509–1524. [83](#), [84](#), [85](#)
- [BCX91] V.N. Bringi, V. Chandrasekar, and R. Xiao, *Raindrop axis ratios and size distributions in florida rainshafts: an assessment of multiparameter radar algorithms*, IEEE Transactions on Geoscience and Remote Sensing **36** (1991), no. 3, 703–715. [83](#)
- [BF88] Richard L. Burden and J. Douglas Faires, *Numerical analysis*, fourth ed., PWS-Kent Publishing Co., Boston, 1988. [70](#), [81](#)
- [BKI91] Kenneth V. Beard, Rodney J. Kubesh, and Harry T. Ochs III, *Laboratory measurements of small raindrop distortion. part i: Axis ratios and fall behavior*, Journal of the Atmospheric Sciences **48** (1991), no. 5, 698–710. [83](#)
- [BTL90] Marc Berger, Terry Trout, and Nancy Levit, *Ray tracing mirages*, IEEE Comput. Graph. Appl. **10** (1990), no. 3, 36–41. [69](#)
- [Cow08] John M. Cowley, *Diffraction physics*, Elsevier Science Ltd, 2008. [83](#)
- [CPP⁺05] Eva Cerezo, Frederic Pérez, Xavier Pueyo, Francisco J. Seron, and Francois X. Sillion, *A survey on participating media rendering techniques*, vol. 21, 2005. [67](#), [68](#)
- [CS04] E. Cerezo and F. J. Seron, *Rendering natural waters taking fluorescence into account*, Computer Animation and Virtual Worlds **15** (2004), 1–14. [69](#)
- [DBB02] Philip Dutre, Philippe Bekaert, and Kavita Bala, *Advanced global illumination*, AK Peters, Ltd., July 2002. [65](#)
- [DJ05] C. Donner and H. W. Jensen, *Light diffusion in multi-layered translucent materials*, SIGGRAPH (Los Angeles, California), ACM, 2005, pp. 1032–1039. [69](#)
- [dLE07] E. d’Eon, D. Luebke, and E. Enderton, *Efficient rendering of human skin*, Eurographics Symposium on Rendering (Grenoble, France) (J. Kautz and S. Pattanaik, eds.), Eurographics Association, 2007, pp. 147–157. [69](#)
- [DP80] J.R. Dormand and P.J. Prince, *A family of embeded runge-kutta formulae*, Journal of Computational and Applied Mathematics **6(1)** (1980), 19–26. [81](#)
- [Gia89] Douglas C. Giancoli, *Physics for scientists and engineers*, Prentice Hall, 1989. [85](#)

REFERENCES

- [GJJD09] Diego Gutierrez, Henrik Wann Jensen, Wojciech Jarosz, and Craig Donner, *Scattering*, SIGGRAPH ASIA '09: ACM SIGGRAPH ASIA 2009 Courses (New York, NY, USA), ACM, 2009, pp. 1–620. [67](#)
- [Gla89] Andrew S. Glassner (ed.), *An introduction to ray tracing*, Academic Press Ltd., London, UK, UK, 1989. [66](#)
- [Gla94] A. S. Glassner, *A model for fluorescence and phosphorescence*, Fifth Eurographics Workshop on Rendering, Springer-Verlag, 1994, pp. 60–70. [69](#)
- [GMAS05] Diego Gutierrez, Adolfo Munoz, Oscar Anson, and Francisco J. Seron, *Non-linear volume photon mapping.*, Rendering Techniques, 2005, pp. 291–300. [69](#), [71](#), [78](#)
- [GMN94] Jay S. Gondek, Gary W. Meyer, and Jonathan G. Newman, *Wavelength dependent reflectance functions*, SIGGRAPH '94 (New York, NY, USA), ACM, 1994, pp. 213–220. [70](#), [86](#)
- [GNJJ08] Diego Gutierrez, Srinivasa G. Narasimhan, Henrik Wann Jensen, and Wojciech Jarosz, *Scattering*, SIGGRAPH Asia '08: ACM SIGGRAPH ASIA 2008 courses (New York, NY, USA), ACM, 2008, pp. 1–12. [67](#), [68](#)
- [Grö95] Meister Eduard Gröller, *Nonlinear raytracing: visualizing strange worlds*, Visual Computer **11** (1995), no. 5, 263–274. [69](#)
- [GS87] Jeffrey Goldsmith and John Salmon, *Automatic creation of object hierarchies for ray tracing*, IEEE Comput. Graph. Appl. **7** (1987), no. 5, 14–20. [66](#), [93](#)
- [GSAM04] D. Gutierrez, F. J. Seron, O. Anson, and A. Munoz, *Chasing the green flash: a global illumination solution for inhomogeneous media*, SCCG'04: Proc. of the 20th Spring Conference on Computer graphics (New York, NY, USA), ACM Press, 2004, pp. 97–105. [70](#)
- [GSMA06] Diego Gutierrez, Francisco J. Serón, Adolfo Muñoz, and Oscar Anson, *Simulation of atmospheric phenomena.*, Computers & Graphics **30** (2006), no. 6, 994–1010. [69](#), [70](#), [74](#), [75](#), [76](#)
- [GSMA08] Diego Gutierrez, Francisco J. Serón, Adolfo Muñoz, and Oscar Anson, *Visualizing underwater ocean optics*, Computer Graphics Forum **27** (2008), no. 2, 547–556.
- [Hal89] Roy Hall, *Illumination and color in computer generated imagery*, Springer-Verlag New York, Inc., New York, NY, USA, 1989. [70](#)
- [JB02] H. W. Jensen and J. Buhler, *A rapid hierarchical rendering technique for translucent materials*, SIGGRAPH (San Antonio, Texas), ACM, 2002, pp. 576–581. [69](#)
- [Jen01] H.W. Jensen, *Realistic image synthesis using photon mapping*, A.K. Peters, Natick, Massachusetts, 2001. [66](#), [67](#), [68](#), [71](#), [73](#)
- [JMLH01] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan, *A practical model for sub-surface light transport*, SIGGRAPH (Los Angeles, California), ACM, 2001, pp. 511–518. [69](#)
- [KW08] Malvin H. Kalos and Paula A. Whitlock, *Monte carlo methods*, Wiley-VCH, 2008. [65](#)
- [KYNN91] Kazufumi Kaneda, G. Yuan, Eihachiro Nakamae, and Tomoyuki Nishita, *Realistic visual simulation of water surfaces taking into account radiative transfer*, Proc. of CAD/Graphics 91, 1991, pp. 25–30. [69](#)
- [Lava] Philip Laven, *Mieplot* - <http://www.philiplaven.com/mieplot.htm>. [70](#), [83](#), [84](#)

-
- [Lavb] Philip Laven, <http://www.philiplaven.com/>. 70, 83, 88
- [LHM05] Andrei Lințu, Jörg Haber, and Marcus Magnor, *Realistic solar disc rendering*, WSCG'2005 Full Papers Conference Proceedings (Vaclav Skala, ed.), 2005, pp. 79–86. 69
- [LLT95] S.G. Lipson, H. Lipson, and D.S. Tanhauser, *Optical physics*, third edition ed., Cambridge, 1995. 86
- [Lor90] L. Lorenz, *Lysbevælgelser i og uden for en af plane lysblger belyst kugle. det kongelig danske videnskabernes selskabs skrifter*, Rkke, Naturvidenskabelig og Matematisk Afdeling **6** (1890), no. 1, 2–62. 70
- [MGS06] Adolfo Muñoz, Diego Gutierrez, and Francisco J. Serón, *Efficient physically-based simulation of non-linear media*, GRAPHITE (Y. T. Lee, Siti Mariyam Hj. Shamsuddin, Diego Gutierrez, and Norhaida Mohd Suaib, eds.), ACM, 2006, pp. 97–105. 74, 96
- [MGS07] Adolfo Munoz, Diego Gutierrez, and Francisco J. Seron, *Optimization techniques for curved path computing*, Vis. Comput. **23** (2007), no. 7, 493–502. 74, 96
- [Mie08] G. Mie, *Beitr age zur optik tr uber medien, speziell kolloidaler metall osungen*, Annalen der Physik **25** (1908), no. 3, 377–445. 70, 83
- [Mob94] Curtis D. Mobley, *Light and water: Radiative transfer in natural waters*, Academic Press, Inc., San Diego, 1994. 69
- [Mus90] F. Kenton Musgrave, *A note on ray tracing mirages*, IEEE Comput. Graph. Appl. **10** (1990), no. 6, 10–12. 69
- [MVB03] F. Wittrock M. Vountas, A. Richter and J. P. Burrows, *Inelastic scattering in ocean water and its impact on trace gas retrievals from satellite data*, Atmos. Chem. Phys. **3** (2003), 1365–1375. 69
- [NSTN93] Tomoyuki Nishita, T. Shirai, K. Tadamura, and Eihachiro Nakamae, *Display of the earth taking into account atmosphere scattering*, Computer Graphics (SIGGRAPH '93 Proceedings), vol. 24, 1993, pp. 175–182. 69
- [PA01] Simon Premoze and Michael Ashikhmin, *Rendering natural waters*, Comput. Graph. Forum **20** (2001), no. 4, 189–199. 69
- [PH04] Matt Pharr and Greg Humphreys, *Physically based rendering: From theory to implementation*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004. 66
- [RSH00] Erik Reinhard, Brian E. Smits, and Chuck Hansen, *Dynamic acceleration structures for interactive ray tracing*, Proceedings of the Eurographics Workshop on Rendering Techniques 2000 (London, UK), Springer-Verlag, 2000, pp. 299–306. 66
- [SL96] Jos Stam and Eric Languenou, *Ray tracing in non-constant media*, Proceedings of the eurographics workshop on Rendering techniques '96 (London, UK), Springer-Verlag, 1996, pp. 225–ff. 69
- [TMD⁺04] T. Tawara, K. Myszkowski, K. Dmitriev, V. Havran, C. Domez, and H.-P. Seidel, *Exploiting temporal coherence in global illumination*, SCCG '04: Proceedings of the 20th spring conference on Computer graphics (New York, NY, USA), ACM Press, 2004, pp. 23–33. 80
- [van57] H. C. van de Hulst, *Light scattering by small particles*, Dover Publications Inc., 1957. 86, 88

REFERENCES

- [WK06] Carsten Wächter and Alexander Keller, *Instant ray tracing: The bounding interval hierarchy*, In *Rendering Techniques 2006 Proceedings of the 17th Eurographics Symposium on Rendering*, 2006, pp. 139–149. [66](#), [75](#)
- [WTP01] Alexander Wilkie, Robert F. Tobler, and Werner Purgathofer, *Combined rendering of polarization and fluorescence effects*, *Proceedings of the 12th Eurographics Workshop on Rendering Techniques (London, UK)*, Springer-Verlag, 2001, pp. 197–204. [69](#), [70](#)
- [WWLP06] Alexander Wilkie, Andrea Weidlich, Caroline Larboulette, and Werner Purgathofer, *A reflectance model for diffuse fluorescent surfaces*, 11 2006, pp. 321–328. [69](#)

Chapter 5

Results

5.1 Material acquisition from inverse rendering

We have presented a method based on genetic algorithms suitable for capturing the appearance of opaque and translucent materials depicted in single images (see section 3.2). The algorithm converges to an approximate solution with little user interaction. Figure 5.2 shows how the algorithm works, as the image rendered with the estimated parameters evolves through iterations converging to the input image. As our aim was to study the performance of the algorithm and the influence of the configuration parameters our termination condition was set to a maximum number of generations, i.e. 50, and not to a maximum error between the final and input images.

The actual ground truth values of the reflectance parameters and the evolution of the estimated values can be seen in table 5.1. Even though the number of generations is not very large almost all errors are below 10%, and it can be seen that those parameters which have very large errors do not have a significant relevance perceptually in the final result.

	Ground truth	Initial value	#10	#20	#30	#40	#50	Error
η	1.5	1.3	1.18195	1.29587	1.36760	1.35480	1.35480	9.7%
g	0.5	0	0.43301	0.49859	0.50151	0.50151	0.50151	0.3%
b	20	10	15.8393	17.6512	18.4832	18.4832	18.4832	7.6%
$\sigma_{s,R}$	2.19	1	1.31844	2.26811	2.26811	2.07799	2.07799	5.1%
$\sigma_{s,G}$	2.62	1	1.80584	2.76615	2.76615	2.54309	2.54309	2.9%
$\sigma_{s,B}$	3	1	2.24059	2.42196	2.42196	2.91114	2.91114	3.0%
$\sigma_{a,R}$	0.0021	1	0.04108	0.01706	0.01527	0.01527	0.00685	226.2%
$\sigma_{a,G}$	0.0041	1	0.01207	0.03432	0.02742	0.02742	0.00916	123.4%
$\sigma_{a,B}$	0.0071	1	0.00650	0.00477	0.00948	0.00948	0.00725	2.1%

Table 5.1: Evolution of the values of the estimated reflectance parameters as the number of generations increases

Additional results are shown in Figure 5.1. The probabilities of replacement, crossover and mutation were all fixed to 0.8, the number of generations was 50 and the number of individuals in each

5. RESULTS

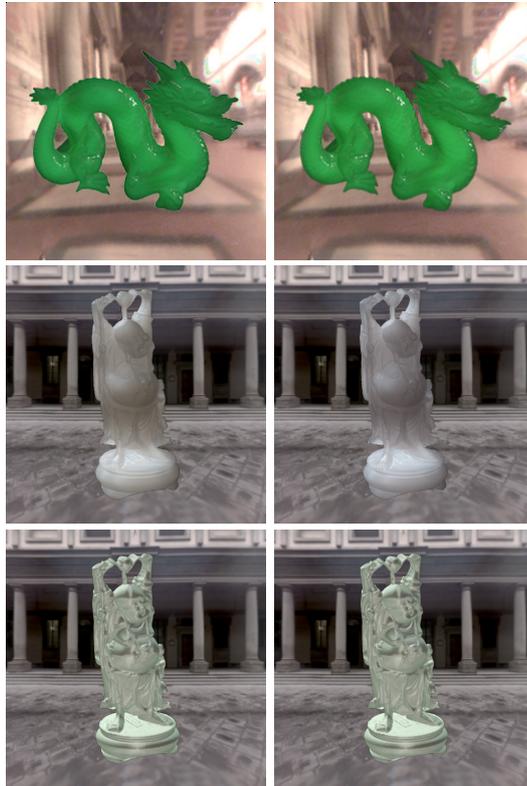


Figure 5.1: *Left column:* Original image. *Right column:* Image rendered with our algorithm. In the first two rows the material is translucent, and thus modeled with the *dipole* model. The last row shows an opaque material, modeled with the *Phong* model.



Figure 5.2: Evolution of the result of the genetic algorithm as the number of generations increases. Left: original image from which the Buddha's reflectance properties are meant to be captured. Rest (from left to right): partial results every ten generations, showing convergence to the solution.

5.2 BSSRDF approximation from single HDR photographs

generation was set to 40. All images in this paper have been rendered on a AMD Opteron Quad-core machine @3GHz and 4Gb of RAM, and took between 15 and 20 minutes. For the diffusion approximation, we have used a fast hierarchical rendering technique (JB02).

One of the main lines for future research is exploring the possibilities that mutation techniques can offer with the objective of accelerating convergence to the solution and of overcoming local minima. Besides, the operation with the highest cost is rendering the scene with each set of parameters for evaluation by comparison with the original image, so creating the chromosomes of possible solutions intelligently instead of relying on brute force is vital, and more sophisticated mutation functions could also help in this direction. To reduce the parameter space, we have assumed that information of light sources, geometry and camera position was known, and only the reflectance characteristics of an object in the image were unknown. It would be interesting to stress our approach further and see how genetic algorithms perform as the problem becomes even more ill-posed. Still, the potential use of this technique for acquiring materials from photographs is still questionable, and as a consequence other options have been considered (as it can be seen in section 3.3).

5.2 BSSRDF approximation from single HDR photographs

In section 3.3 we have introduced a method to approximate a BSSRDF from an optically thick object in an image, which greatly simplifies current capture techniques from previous works. We have demonstrated our model in controlled environments, where both the light direction and the object's shape are known: we fit our data to the dipole model and compare our results with physically measured materials (Jen01), obtaining a very reasonable approximation of the extinction and albedo coefficients of the materials tested. We then generalize our method to off-the-shelf images, where neither the lighting conditions nor the shape of the object are known, and rely on simple existing light detection and depth extraction algorithms. We note that perfect solutions do not exist for neither problem, so we choose simple algorithms that work sufficiently well for our purposes. The light detection algorithm works best if there is a dominant directional light in the input image, whereas depth is better approximated for simple objects.

We solve Equation 3.15 in 30-40 seconds on a Dual Opteron @2.2 GHz. with 4 GB. of RAM, using between 20 and 30 basis functions for our representation. The recovered BSSRDF for the different materials can be directly used for rendering with no restrictions: for different geometries (see Figure 5.5) and under different illumination conditions (see Figure 5.4). Figure 5.3 show several results for a wide range of translucent materials, including wax, soap, milk, ketchup, orange juice, detergent, grape and human skin. Our method works well even for extremely complex materials like skin, although it obviously cannot reproduce the subtleties of light transport in its multi-layered structure.

Real objects present varying degrees of translucency depending on their size. However, it is not possible to accurately determine this parameter from a single image. To simulate this change in appearance, the captured BSSRDF can be modified as: $R'_d(r) = R_d(\frac{r}{s})$ where s is the relative scale factor between the source and the target objects (see Figure 5.6, left). Additionally, similar to (NGD⁺06), our method allows for simple material edits, like linearly combining different diffusion profiles (see Figure 5.6, right). More advanced or artistic effects can be achieved by specific material editing tools such as SubEdit (STPP09), which in turn can benefit from the ease of use of our image-based acquisition method.

There are several limitations in our method. We are currently bounded by the current state of the art in depth and light approximation algorithms from single images, which in practice means that the algorithm works better with images showing simple, convex shapes lit from one direction. Violating our initial assumptions usually leads to poor results. However, we believe that we have shown a

5. RESULTS



Figure 5.3: Results of our algorithm. The small insets show the original images where the material properties are acquired from (please refer to the supplementary material for the complete data). In reading order, grape, orange soap, wax blue soap, whole milk, purple soap, ketchup, orange juice, whitish soap, liquid detergent, skin and greenish soap.

5.2 BSSRDF approximation from single HDR photographs

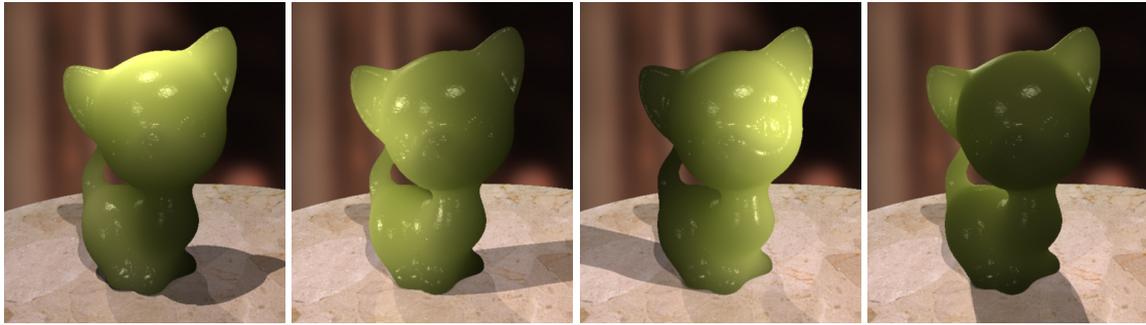


Figure 5.4: The estimated BSSRDF for the grape material used to render cats under different lighting conditions.



Figure 5.5: Several estimated BSSRDFs (grape, orange soap and wax) used to render different geometries lighting conditions.

5. RESULTS



Figure 5.6: Left: Different apparent sizes for the cats, with scaling factors s of 0.25 and 1. Notice how scattering becomes more visible as the object's size decreases. Right: Two new soap materials obtained by blending different diffusion profiles.



Figure 5.7: Example of our technique as an image-editing tool. From left to right: original photograph, transfer of the wax material from the candle to the owl, and transfer from the purple wax in Figure 5.3 to the owl.

varied enough range of objects and images to still consider the method practical. Additionally, we have found that using more than 40 basis functions tends to increase the condition number of the matrices and yield noisy solutions. We thus capture a coarser representation of the diffusion profiles, which nevertheless produces good results.

Our method could also be potentially used in an image-editing context, by transferring the captured profile in an image object to another. By applying the same depth estimation technique both to the source and target objects, a new depiction of the latter can be created (see Figure 5.7). The main drawback of the technique is the double depth estimation process, which tends to accumulate larger errors in the final result. We believe this is an interesting avenue for future work.

5.3 Underwater imagery

Chapter 2 has introduced the concept of participating media and how it is usually defined the corresponding absorption and scattering coefficients plus a phase function. Chapter 3 shows how can these information can be obtained in general situations. Furthermore, section 2.3.3 has introduced the concept of inelastic scattering and how can it be defined.

On the other hand, chapter 4 has briefly introduced algorithms for simulating generic participating media, and section 4.2 has introduced a new algorithm (contribution of this PhD) that is able to simulate participating media with inelastic scattering.

Also, section 3.5 introduces a bio-optical model of ocean water, obtained from oceanography and photography literature, filling our participating media model with data for absorption and both elastic and inelastic scattering. As the reader most probably has realized, this enables the possibility of simulating underwater imagery using the bio-optical model.

We have used the values from Table 4.3 for our simulations. In the images produced we only vary the chlorophyll concentration C , minerals and detritus turbidity $\alpha_d(400)$ and CDOM turbidity $\alpha_y(440)$. The choice of those three parameters to reduce the dimensionality of the model was based on their greater overall influence on the resulting light field. The photon map contains 400000 photons, with 250 used in the estimation of radiance. Ray-marching depth is set at 200 steps. Each of the images has been rendered in a Dual Xeon Pentium 4 at 2.8GHz with 2GB RAM at 512×384 resolution, casting one ray per pixel.

We have performed per-scene energy-balance tests and found that almost 99% of the energy emitted by the light sources is absorbed by the medium after only a few interactions of the photons with the medium. Fig. 5.9 shows the result for a scene with $C = 0.1$, $\alpha_d(400) = 0.1$ and $\alpha_y(440) = 0.1$. It can be seen how after the fourth interaction there is very incremental variation on the absorbed energy (expressed as flux), and after the fifth bounce the contribution is negligible. This relatively fast convergence is due to the strong absorption in water. We have therefore limited the number of interactions per photon to five, in order to speed up the simulations.

Variations of the parameters C , $\alpha_d(400)$ and $\alpha_y(440)$ yield different probabilities for absorption, elastic and inelastic scattering events, which in turn affect the in-water light field. The results can be seen in Fig. 5.8, with each of the varying parameters influencing the final light field as follows:

Chlorophyll concentration (C) affects mainly both elastic and inelastic scattering. The effects of inelastic scattering are mostly masked by the more predominant elastic scattering and absorption, which increases slowly. The third column in Fig. 5.8 shows brighter images than the previous two due

5. RESULTS

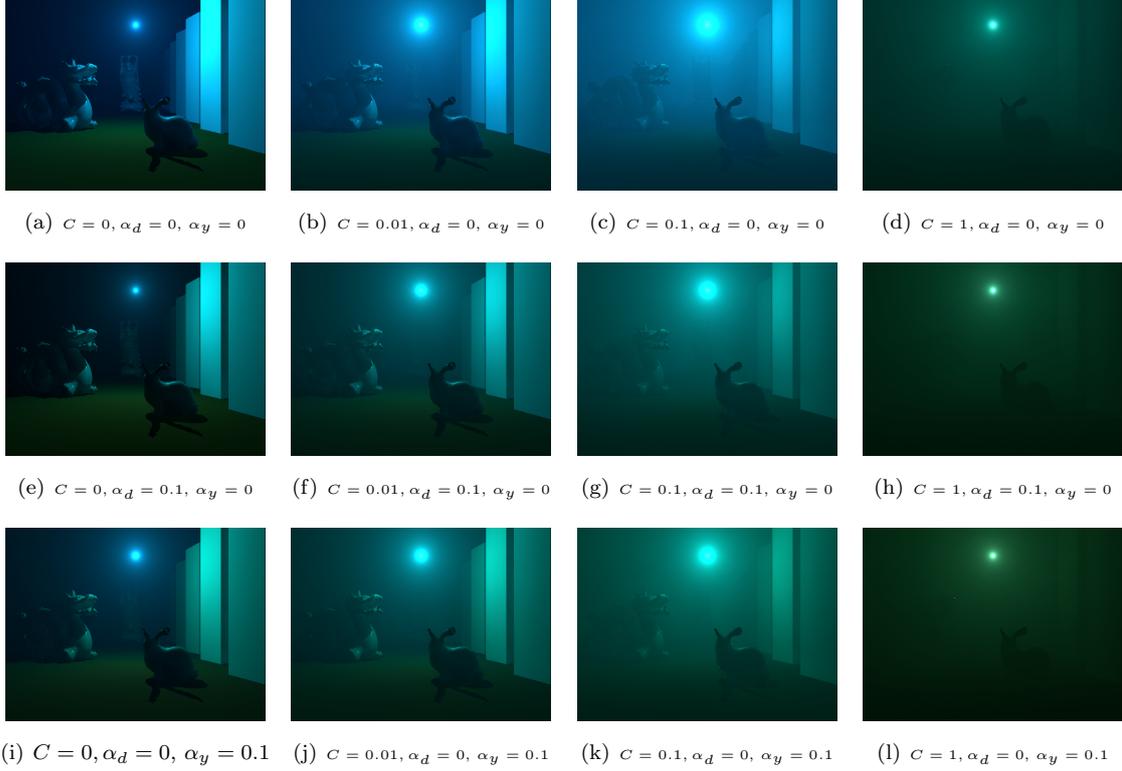


Figure 5.8: Resulting pictures varying the chlorophyll concentration C , the minerals and detritus turbidity α_d at 400nm and the CDOM turbidity α_y at 440nm.

to in-scattering. For higher values (fourth column), out-scattering prevails and the images become darker.

Minerals and detritus turbidity ($\alpha_d(400)$) increases absorption at lower wavelengths, thus reducing the brightness of the scene and the overall blue hue. Scattering is also increased, making the images appear murkier. Fig. 5.8 shows variations of the minerals and detritus turbidity between the first and second rows for direct comparison.

CDOM turbidity ($\alpha_y(440)$) increases absorption slightly (darker images) and introduces inelastic scattering (change in hue). This can be seen by comparing the first and third rows in Fig. 5.8.

We have undergone a visual validation of our model by rendering different natural waters. Fig. 5.10 shows the resulting underwater images for Atlantic, Mediterranean, Baltic, North Sea and shallow coastal waters rich in CDOM respectively. All the images have been simulated at the same depth and are illuminated by the same isotropic point light source. The changes in color are clearly noticeable, from a darker blue in the case of Atlantic water, to the greener hue in the image of the North Sea. The smaller patches below the first four images correspond to the simulations by Frisvad et al. (FCJ07) for the same types of water, and are shown for comparison purposes. Our simulations based on radiative transfer approximately match their simulations based on Lorenz-Mie theory. The differences are mainly owed to two factors: on the one hand, the overall darker tone in our images is due to in-water absorption, whereas (FCJ07) renders the *surface* of the water body; on the other hand, the absence of inelastic scattering effects in (FCJ07) can have a visible influence the final appearance of water, as

5.4 Efficiency of non-linear media simulation

shown in Fig. 5.11 for the Baltic case. The properties of the water have been adjusted according to measurements found in (BSF⁺03)(Mob94) for our bio-optical model and (BSF⁺03) in the model by Frisvad et al. In both cases, it is only the changes in the constituents of the waters which yield the different colors. We have additionally performed a numerical analysis of the in-water radiance field, to quantify the influence of each constituent. The results can be seen in Fig. 5.12.

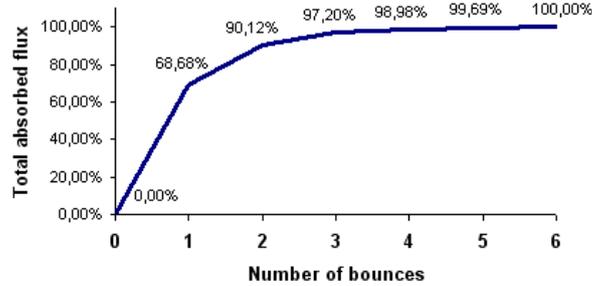


Figure 5.9: Convergence of the algorithm using our bio-optical model.



Figure 5.10: Rendered images of different waters. From left to right: Atlantic, Mediterranean, Baltic, North Sea and shallow coastal waters rich in CDOM. Smaller patches below for comparison purposes by Frisvad et al. (FCJ07) (used with permission).

5.4 Efficiency of non-linear media simulation

One of the goals of this PhD is to simulate phenomena owed to non-linear media in a much more efficient way by using several techniques and specific numerical methods. In subsection 5.4.1 a study about the effect of each numerical method to the simulation of a curved path can be found. Subsection 5.4.2, on the other hand, presents a set of tests that proves the improvement on rendering time owed to each optimization technique presented on section 4.3.

5.4.1 Study of numerical methods

As stated in section 4.3.5, the chosen numerical method greatly influences computation time and efficiency. We compare the following four methods (BF88):

5. RESULTS

- Euler method.
- Order 2 Runge-Kutta method.
- Order 4 Runge-Kutta method.
- Dormand Prince method (DP80).

The errors of the figures have been computed by choosing a very simple medium in which the index of refraction varies just in the vertical axis. The variation follows the equation of a straight line:

$$n = mx + b \quad (5.1)$$

where n is the index of refraction, x is the vertical axis and m and b are the parameters that define the slope and the value at zero of the equation of the line.

In (GSMA06) it is shown that this specific configuration of the index of refraction can be solved both analytically and numerically, so we compare the analytical solution with the numerical solution given by each of the numerical methods. The error of each numerical method is the distance between the ending point given by the method and the ending point given analytically. The time includes tracing a single ray through the medium, in seconds.

Figure 5.4.1 includes all the results of the several tests about the performance of the different numerical methods. Figures 5.13(a) and 5.13(b) show the performance of the methods considering variation in the corresponding parameter (step or tolerance). The distance of the ray is fixed to 1000 and the gradient (slope of the straight line) is 0.1. As expected, time increases when the step becomes smaller. Euler is more efficient than Runge-Kutta, and order 2 more efficient than order 4. Dormand-Prince takes more time as tolerance is reduced, and is not comparable to other methods because it uses different parameters.

Regarding the committed error, Dormand-Prince (figure 5.13(d)) is more stable, because as tolerance decreases, error decreases. This is due to the fact that this method uses an adaptive step. The methods which have a fixed step (figure 5.13(c)) present a different behavior. Error is big both when the step is either too big or too small. If the step is too big, the error committed at each of the steps is too big; if the step is too small, the error is small at each of the steps but a huge amount of steps is required, so as a consequence the accumulated error increases.

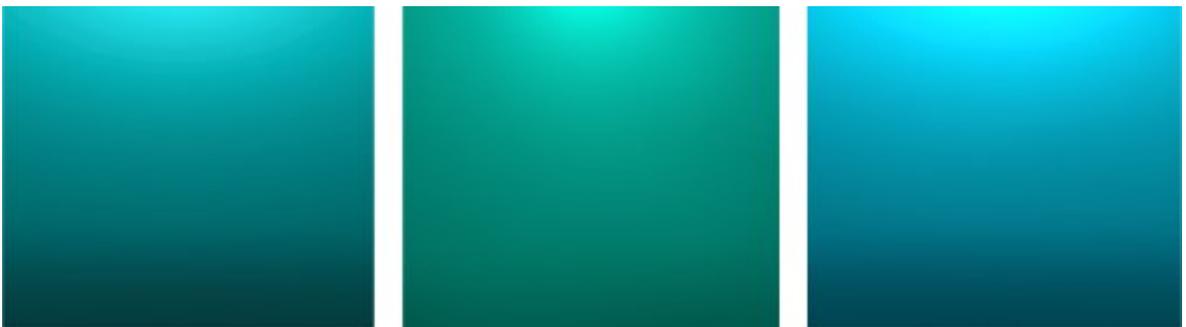


Figure 5.11: The influence of inelastic scattering in the apparent optical properties of water (Baltic sea): left, inelastic scattering events included in the simulation. Middle, no inelastic scattering. Right, no anti-Stokes interactions included.

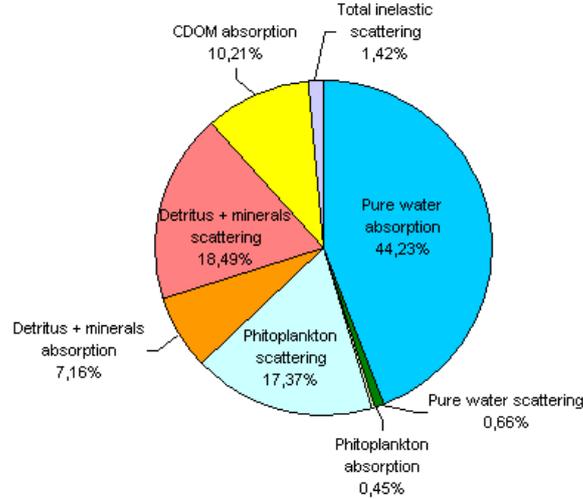


Figure 5.12: Influence of each of the different possible interactions of our model on the resulting in-water light field (Baltic sea).

The influence of the distance traveled by the ray has also been studied, with step size $5 \cdot 10^{-3}$ for the methods with fixed step and tolerance 10^{-6} for Dormand-Prince. The slope is still 0.1. Figures 5.13(e) and 5.13(f) show the result. Surprisingly, in short distances, the fixed methods perform slightly better both in time and error. This is due to a right choice of the step size. In long distances, Dormand Prince performs by far much better.

Figures 5.13(g) and 5.13(h) show the influence of the gradient (slope of the line) on the performance of the methods. The parameters are the same than in the study of the effect of distance, and the distance is fixed to 1000. Time is not affected by the gradient for fixed-step methods, as the number of steps that they compute is exactly the same. However committed varies depending on the gradient (being too high at low gradients). On the other hand, Dormand-Prince, having an adaptive step size, has a varying time depending on the gradient (being high at low gradients). However the committed error is quite low under all gradients, as the step adapts in order to reduce it.

As a conclusion, for specific circumstances (short distances and specific gradients) fixed-step methods seem to perform better, if choosing the right step, with Euler being more efficient and Runge-Kutta (order 4) more accurate. However, it is not always possible to find the exact right step for a certain simulation, plus under general circumstances, Dormand Prince performs better. As a consequence, Dormand Prince will be used in all the cases, unless the scene is specially simple and small (circumstances under which fixed-step methods perform better).

5.4.2 Efficiency due to optimization techniques

5.4.2.1 Test scenes

In order to check rendering times and committed error, four very different test scenes have been taken into account. Figure 5.14(a) presents an inferior mirage. The most interesting characteristic of this scene for our simulation methods is its simplicity. The medium has just one gradient on its

5. RESULTS

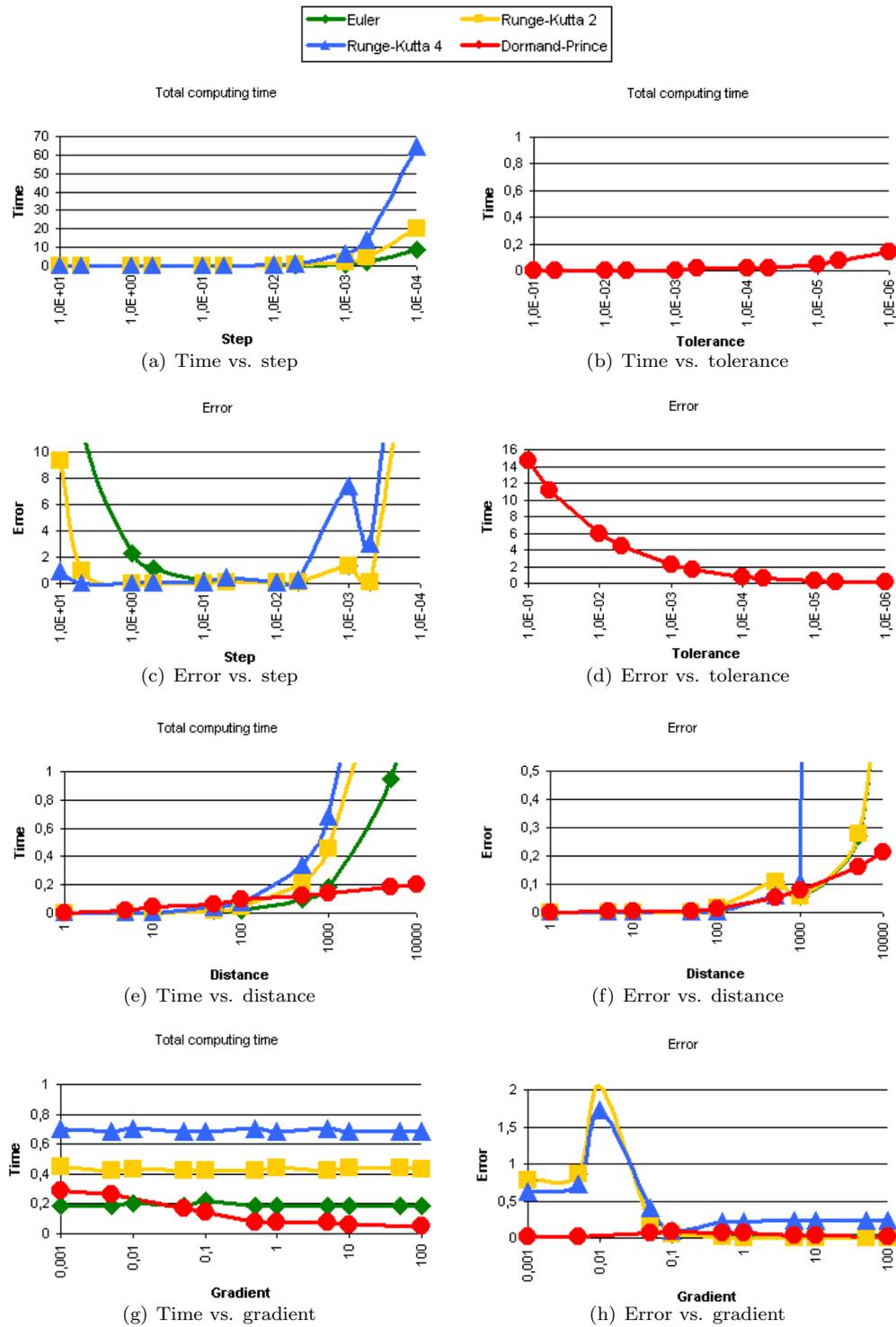


Figure 5.13: Performance of numerical methods

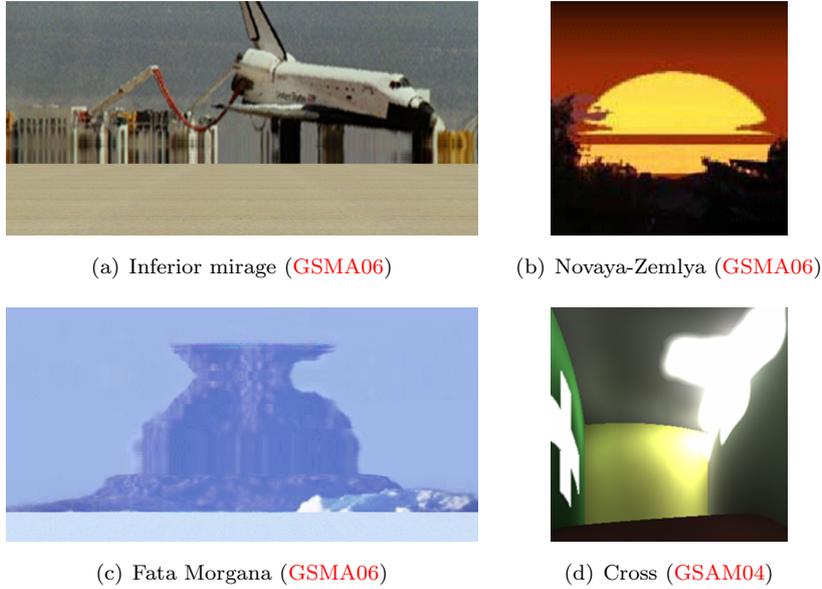


Figure 5.14: Test scenes

index of refraction (near the floor). The shuttle is located at 2 kilometers from the camera. Figure 5.14(b) presents a Novaya-Zemlya effect. The medium is as simple as the inferior mirage. However, the geometry (the sun) is much further away ($150 \cdot 10^6$ kilometers from the camera), and, as a consequence, numerical methods require by far many more steps. Figure 5.14(c) presents the Fata Morgana effect. The geometry close (12 kilometers) as in the inferior mirage, but the medium is more complex, with two strong gradients of the index of refraction at different heights. Figure 5.14(d) presents a gradient of a medium in a scene in which light enters to a room through a cross-shaped window. The gradient has been exaggerated for demonstration purposes. The dimensions of the room are $5 \times 5 \times 5$ meters. In this test scene, the effect of the optimization techniques with particle tracing is considered.

The index of refraction of the inferior mirage, Fata Morgana and Novaya-Zemlya are based on the model described in (GSMA06). In the Cross scene, the index of refraction follows the equation $n = 0.1y + 1$ where n is the index of refraction and y is the vertical axis.

5.4.2.2 Rendering times study

In table 5.2, the computation times (in format *minutes : seconds*) and the committed errors (considering kilometers as unit of the scene, errors compared to the ground truth) in a single frame of the test scenes can be seen. These kilometers are considered as the average error of each collision point (of rays in the three previous scenes and of photons in the cross scene, sampling the same shooting directions).

The undersampling technique for ray tracing has parameter $\alpha = \frac{1}{2}$, photon mapping undersampling has parameters $t = 2$, $n = 3$ and smart step selection in photon mapping has parameter $t = 3$.

Ground truth is referring to the scene simulated using the algorithm introduced in section 4.1.6, which comes from previous work (GSMA06), using the Dormand Prince numerical method, with a

5. RESULTS

TECHNIQUE	INFERIOR MIRAGE		NOVAYA-ZEMLYA		FATA MORGANA		CROSS	
	TIME	ERROR	TIME	ERROR	TIME	ERROR	TIME	ERROR
GROUND TRUTH	8:15	0.000	10:12	0.000	9:42	0.000	14:10	0.000
PARALLELIZATION	4:32	0.000	5:58	0.000	5:11	0.000	11:32	0.000
SYMMETRIES	6:04	0.000	8:14	0.000	7:32	0.000	12:26	0.000
UNDERSAMPLING	5:54	0.005	8:01	0.230	6:58	0.011	11:51	0.004
PATH CACHING	4:12	0.000	4:42	0.000	4:22	0.000	11:05	0.000
SMART STEP SEL.	3:57	0.000	4:21	0.000	3:56	0.000	11:10	0.000
COMBINATION	1:55	0.019	2:51	0.032	2:02	0.042	9:31	0.002

Table 5.2: Results of the tests.

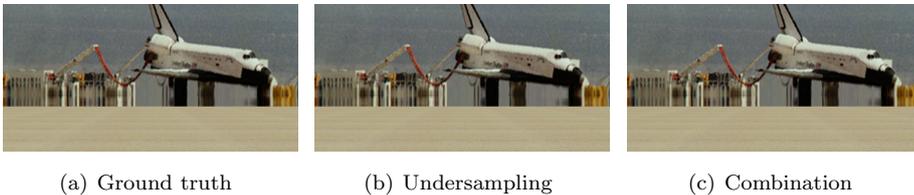


Figure 5.15: Sample scene that has been simulated using the different techniques.

tolerance of 10^{-6} km. Inferior mirage and Fata Morgana scenes have been rendered at 800×400 pixel resolution, while Novaya-Zemlya and cross scenes have been simulated at 400×400 pixel resolution. 500000 photons were shot on the cross scene, and the radiance estimation was done using 300 photons. All the scenes have been rendered using one ray per pixel, on a Pentium IV at 2.8Ghz (with hyper-threading technology).

As it has been stated before, the technique of *path caching* is related to frame to frame coherency and therefore to animations. However, in the time data of table 5.2, we are considering a single frame. In order to achieve meaningful single frame data we computed before the previous frame (the one that generates the initial rays) and then the measured frame. Regarding the *combination* technique, it refers to a single frame, computed using parallelization, undersampling and the smart step selection techniques.

The effect of the optimizations is not as impressive on the cross scene. That happens because the bottleneck of the simulation is not the curved paths but the radiance estimation of the photon mapping technique. On the other hand, the effect of these techniques happens to be more perceivable on the Fata Morgana scene, in which the traverse medium is more complex than in the rest of the scenery.

Although most of the techniques yield no error because of the underlying theory behind them, the undersampling technique does. However, its errors are negligible compared to the dimensions of the geometry of the scene. Furthermore, the resulting images are visually indistinguishable from the original one. For instance, doing a visual test with the inferior mirage scene, we get the results of figure 5.15. The two simulations that have some error are 5.15(b) and 5.15(c), and they show that the errors are not perceptible.

5.5 Rainbow rendering using simulated phase functions

Section 4.4 has presented a new method for simulating phase functions for rainbows, including phenomena such as polarization (see section 2.1.1.1) and interference (see section 2.1.1.2). This algorithm is able to create simulations from arbitrary geometry. The results presented in this section come from the cooperation with Iman Sadeghi, Philip Laven, Wojciech Jarosz and Henrik Wann Jensen. First we validate our algorithm against different simulation methods in Mieplot (Lav), and then we show our results.

5.5.1 Validation

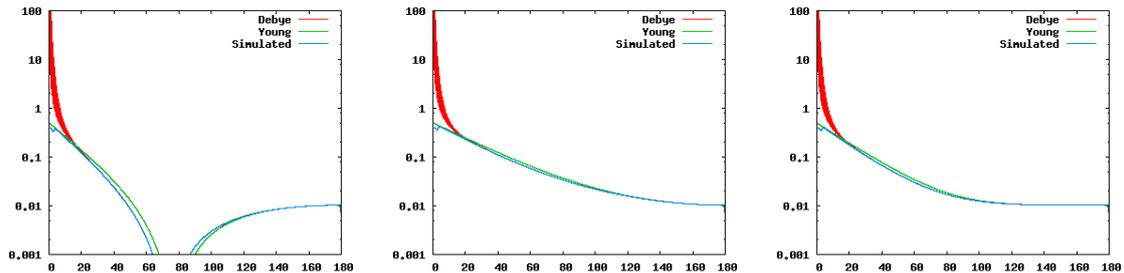


Figure 5.16: Comparison of our method against Young method and Debye series in Mieplot, for bounce $p = 0$. Left: parallel polarization. Middle: perpendicular polarization. Right: unpolarized light. Our simulation is more accurate than Young’s method. The difference on the region $[0^\circ, 20^\circ]$ is owed to diffraction, that is taken into account only by Debye series.

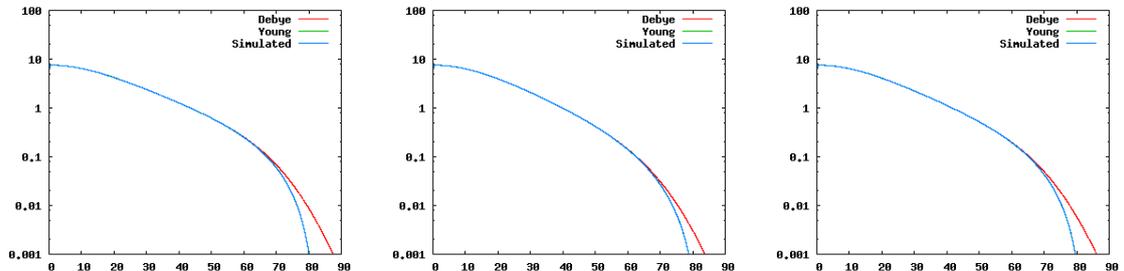


Figure 5.17: Comparison of our method against Young method and Debye series in Mieplot, for bounce $p = 1$. Left: parallel polarization. Middle: perpendicular polarization. Right: unpolarized light. Our simulation matches Young’s method perfectly. The difference on the region in $[70^\circ - 90^\circ]$ is owed to surface waves, that are taken into account only by Debye series.

In order to validate the algorithm, we have simulated several phase functions for spherical particles and compared them with previous methods available in MiePlot (Lav). We have compared our results at each bounce p (until $p = 3$, secondary rainbow) against Debye series (that consider the same phenomena than Mie’s phase function for separate bounces) and Young method (that considers geometrical optics and interference). Figures 5.16, 5.17, 5.18 and 5.19 shows this comparisons for each bounce, respectively. The simulations consider a $0.1mm$ sphere for validation purposes, although our method is able to simulate particles of arbitrary geometry. Taking Debye series as ground truth, our

5. RESULTS

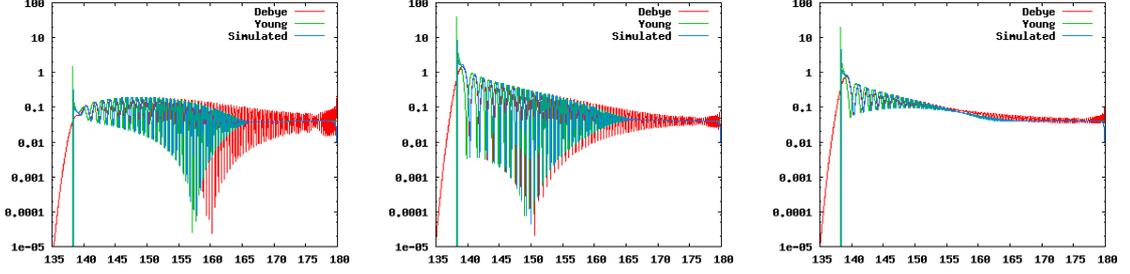


Figure 5.18: Comparison of our method against Young method and Debye series in Mieplot, for bounce $p = 2$ (primary rainbow). Left: parallel polarization. Middle: perpendicular polarization. Right: unpolarized light. Our simulation matches Debye series better than Young method because it takes into account focal lines. There are still some differences, however, that can be explained by diffraction and surface waves.

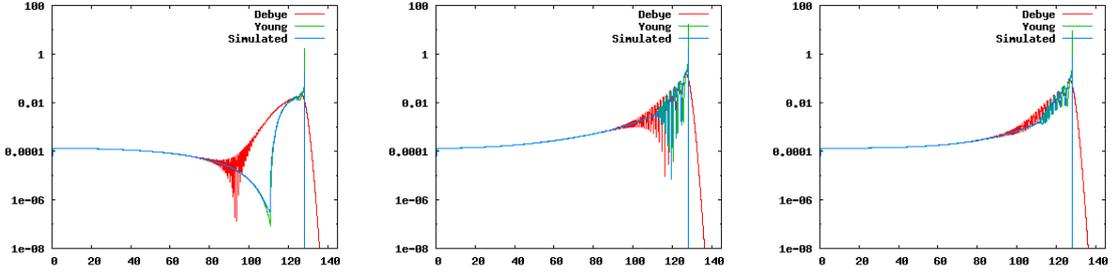


Figure 5.19: Comparison of our method against Young method and Debye series in Mieplot, for bounce $p = 3$ (secondary rainbow). Left: parallel polarization. Middle: perpendicular polarization. Right: unpolarized light. Again, even we still match Debye series better, there are some differences owed to diffraction and surface waves.

method is more accurate than Young method because our method considers focal lines. Still, there are some differences that are owed to the fact that our method does not consider surface waves nor diffraction.

Figure 5.20 shows phase functions simulated with three different methods of Mieplot: Mie theory, Young method (which has been modified in order to include focal lines) and our ray tracing method presented in section 4.4. These validations include bounces 0, 1, 2 and 3 for our simulations, and consider 33 wavelengths between $380nm$ and $700nm$. The comparisons are done for different spherical particle sizes. By interpreting these results we can conclude the following:

- The result of our algorithm is equivalent to the results of the modified Young's method (although our method can be used for arbitrary geometry of the particle). This means that phenomena such as polarization and interference are simulated correctly by our method.
- Mie looks similar to both solutions for bigger particles. This means that the phenomena that are not being considered neither by Young's method nor by ours (such as diffraction or surface waves) are relevant only for small phase functions. Small particles (with radius less than $500\mu m$) are spherical, while bigger particles have different shapes according to speed, surface tension and air resistance (BC87; BKI91; BCX91). As a consequence, while smaller particles will be simulated using Mie's phase function, bigger particles (that are not spheres anymore) will be simulated with our method.

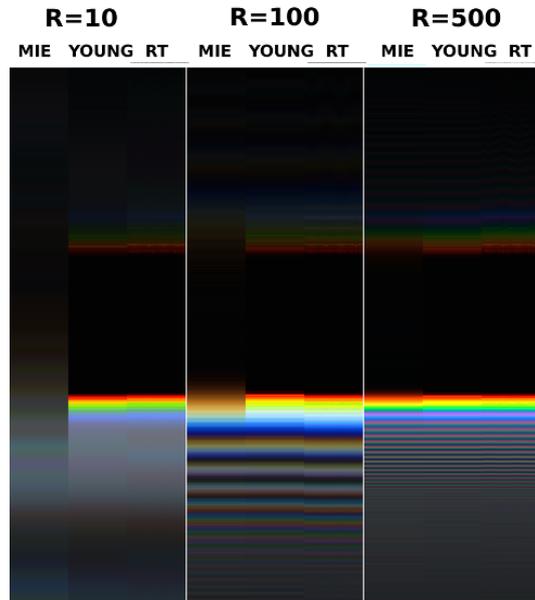


Figure 5.20: Comparison between different simulated phase functions using different methods: Mie, Young (modified to include focal lines) and our method (RT), for different spherical particles of radii $10\mu m$, $100\mu m$ and $500\mu m$. Not all the phase function is included here, but just the range of angles related to the primary and secondary rainbows.

5.5.2 Result



(a) Single rainbow

(b) Primary/Secondary rainbows and Alexander's dark band

Figure 5.21: Different rainbow simulations

Once proved that our algorithm works correctly for particles bigger than $500\mu m$, it is time to see some rendered rainbows. We attempt to match different rainbow photographs with our rendered rainbows using different parameters and different distributions of water drops. The renders have been done using standard ray marching (see section 4.1.4) and considering just single scattering. In our

5. RESULTS

preliminary renders, multiple scattering do not give any relevant visual result compared to plain single scattering, so for efficiency reasons it is ignored in the simulations.

A simulation of the standard rainbow can be seen on figure 5.25. The used particle for this simulation has been a $500\mu m$ sphere. Notice that no secondary rainbow nor other phenomena are perceived, as they are hidden by the bright background. However, with darker backgrounds it is possible to perceive both the primary and secondary rainbow, such as in figure 5.21(b).

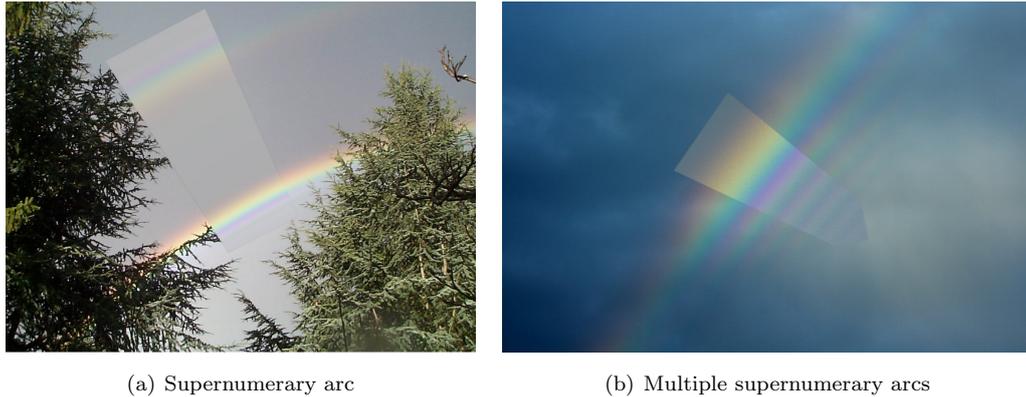


Figure 5.22: Different supernumerary arc simulations

When the right illumination conditions are given, it is possible to perceive supernumerary arcs. These supernumerary arcs are owed to interference, and cannot simulated by a simple geometrical interpretation of the rainbow. Figure 5.22(a) shows the supernumerary arc of a $500\mu m$ sphere.

It is possible also to find multiple supernumerary arcs for the same rainbow, that are also owed to interference. Figure 5.22(b) shows these multiple supernumerary arcs. The separation between the supernumerary arcs and the diminished color of this rainbow, however, are owed to smaller phase functions, so Mie theory was used to simulate a $200\mu m$ sphere.



Figure 5.23: Special kinds of rainbow

It is possible to get also different colored rainbows by changing the spectrum that defines the illumination coming from the sun. In figure 5.23(a), the color of the sun has been modified in order to be red, and therefore other colors of the rainbow not perceived. In the photograph, however, it is possible that this red color was not owed to a sudden "change" of the color of the sun, but by a

5.5 Rainbow rendering using simulated phase functions

modification of the incident irradiance due to Rayleigh scattering on the atmosphere traversed by the light rays. We have preferred to avoid simulating the whole atmosphere of the Earth and simulate this phenomena by modifying the sun spectrum.

A very specific kind of rainbow is called *cloudbow*. It is owed to very small water drops (with radius less than $100\mu\text{m}$) and as a consequence it cannot be simulated used our algorithm. Figure 5.23(b) shows a cloudbow simulated using Mie theory for a $50\mu\text{m}$ sphere.



Figure 5.24: Twinned bows

There is also a phenomena called *twinned rainbow* that is sometimes seen in nature. This twinned rainbow happens when there are two different water drop showers with different sizes. If one (or both) kinds of water drops is big enough so it is not anymore a sphere, then the angular position of the rainbow is altered. Figure 5.24 shows a twinned rainbow, simulated using a $500\mu\text{m}$ sphere and a 1mm radius water drop with a non-spherical shape obtained from a relevant study about water drop shapes (BC87). Notice that this simulation is not possible with Mie's theory.

Finally, figure 5.25 shows a full set of our simulated results.



Figure 5.25: Our rendering results for different types of rainbows: (a) Rainbow derived from Lorenz-Mie theory. (b) Single primary rainbow with considering the water drop size distribution and the angular view of the sun. (c) Double rainbow with a flipped secondary rainbow. (d) Twinned rainbow resulted from mixture of non-spherical water drops and spherical ones. (e) Multiple supernumerary rainbows caused by small water drops with uniform sizes.

5. RESULTS

References

- [BC87] Kenneth V. Beard and Catherine Chuang, *A new model for the equilibrium shape of raindrops*, Journal of the Atmospheric Sciences **44** (1987), no. 11, 1509–1524. [116](#), [119](#)
- [BCX91] V.N. Bringi, V. Chandrasekar, and R. Xiao, *Raindrop axis ratios and size distributions in florida rainshafts:an assessment of multiparameter radar algorithms*, IEEE Transactions on Geoscience and Remote Sensing **36** (1991), no. 3, 703–715. [116](#)
- [BF88] Richard L. Burden and J. Douglas Faires, *Numerical analysis*, fourth ed., PWS-Kent Publishing Co., Boston, 1988. [109](#)
- [BKI91] Kenneth V. Beard, Rodney J. Kubesh, and Harry T. Ochs III, *Laboratory measurements of small raindrop distortion. part i: Axis ratios and fall behavior*, Journal of the Atmospheric Sciences **48** (1991), no. 5, 698–710. [116](#)
- [BSF⁺03] M. Babin, D. Stramski, G. M. Ferrari, H. Claustre, A. Bricaud, G. Obolensky, and N. Hoepffner, *Variations in the light absorption coefficients of phytoplankton, nonalgal particles, and dissolved organic matter in coastal waters around europe*, J. Geophys. Res. **108(C7)** (2003), no. 3211. [109](#)
- [DP80] J.R. Dormand and P.J. Prince, *A family of embeded runge-kutta formulae*, Journal of Computational and Applied Mathematics **6(1)** (1980), 19–26. [110](#)
- [FCJ07] Jeppe Revall Frisvad, Niels Jørgen Christensen, and Henrik Wann Jensen, *Computing the scattering properties of participating media using lorenz-mie theory*, SIGGRAPH '07: ACM SIGGRAPH 2007 papers (New York, NY, USA), ACM, 2007, p. 60. [108](#), [109](#)
- [GSAM04] D. Gutierrez, F. J. Seron, O. Anson, and A. Munoz, *Chasing the green flash: a global illumination solution for inhomogeneous media*, SCCG'04: Proc. of the 20th Spring Conference on Computer graphics (New York, NY, USA), ACM Press, 2004, pp. 97–105. [113](#)
- [GSMA06] Diego Gutierrez, Francisco J. Serón, Adolfo Muñoz, and Oscar Anson, *Simulation of atmospheric phenomena.*, Computers & Graphics **30** (2006), no. 6, 994–1010. [110](#), [113](#)
- [JB02] H. W. Jensen and J. Buhler, *A rapid hierarchical rendering technique for translucent materials*, SIGGRAPH (San Antonio, Texas), ACM, 2002, pp. 576–581. [103](#)
- [Jen01] H.W. Jensen, *Realistic image synthesis using photon mapping*, A.K. Peters, Natick, Massachusetts, 2001. [103](#)
- [Lav] Philip Laven, *Mieplot* - <http://www.philiplaven.com/mieplot.htm>. [115](#)
- [Mob94] Curtis D. Mobley, *Light and water: Radiative transfer in natural waters*, Academic Press, Inc., San Diego, 1994. [109](#)

REFERENCES

- [NGD⁺06] S. G. Narasimhan, M. Gupta, C. Donner, R. Ramamoorthi, S. K. Nayar, and H. W. Jensen, *Acquiring scattering properties of participating media by dilution*, SIGGRAPH (Boston, Massachusetts), ACM, 2006, pp. 1003–1012. [103](#)
- [STPP09] Y. Song, X. Tong, F. Pellacini, and P. Peers, *Subedit: A representation for editing measured heterogeneous subsurface scattering*, ACM Transactions on Graphics (SIGGRAPH) **28** (2009), no. 3, 1–10. [103](#)

Chapter 6

Conclusions

Along this PhD, several extensions and contributions to the modeling and simulation light transport in participating media have been developed and tested. Three approaches for specific participating media models have been defined as an example of the different techniques for modeling participating media:

- Modeling translucent materials from photographs: a technique to estimate BSSRDFs from single HDR photographs with no previous knowledge about geometry nor lighting nor camera calibration. On a first step, optimization with genetic algorithms was tested (section 3.2) although a more polished and practical solution based on linear system resolution is shown to work more efficiently (section 3.3).
- Modeling ocean water from measured data: a full model of ocean water (that includes inelastic scattering) has been gathered, based on oceanography papers, optical data and standardized models (section 3.5).
- Modeling rainbow phase functions from simulation¹: a technique that is able to simulate phase functions for a particle of arbitrary geometry including phenomena such as polarization and wave interference has been developed and validated (section 4.4).

Also, simulation algorithms that extend the traditional definition and simulation algorithms for radiative transfer in participating media haven been developed:

- Simulation of inelastic scattering in participating media: a technique that extends ray-marching and photon mapping to be able to consider inelastic scattering phenomena (such as fluorescence and Raman scattering) in participating media has been developed (section 4.2). Although it works for any generic medium that defines the corresponding properties, it has been applied to the bio-optical model presented in section 3.5.
- Efficient techniques for simulating non-linear media: several techniques that increase the efficiency of the simulation of non-linear media (and as a consequence, of phenomena such as mirages) have been developed.

¹Notice that this technique has been included into the simulation chapter (chapter 4) instead of the modeling one

6. CONCLUSIONS

- Algorithm for simulating polarization and wave interference for generic geometry: applied to simulating the phase function of a rainbow, an algorithm that is able to consider both polarization and wave interference has been developed (section 4.4).

All the results for the different techniques are shown in chapter 5 and validate the correctness of the presented techniques. This work satisfies the goals for this PhD presented in the introduction (see section 1.3). Notice also the measurable results (publications, projects...) of this PhD can be found in section 1.4.

6.1 Future work

Lots of work is still to be done. While many phenomena that are owed to the very specific wave-particle nature of light have already been included in this PhD, still some are missing. Would it be possible to simulate diffraction for computer graphics? Or maybe other quantum phenomena, such as surface waves? How much closer must we get to Maxwell's equations in order to model and simulate every single perceivable phenomenon in nature? There are several phenomena still to simulate that might lead to very interesting possibilities in the field of physically-based rendering.

Furthermore, there is still a lot of work to do in terms of efficiency. Could any of the presented techniques use modern hardware in order to work at interactive rates or even on real time? Non-linear media could easily be adapted to modern graphics hardware (as each ray is completely independent from the others) maybe reaching interactive rates. Translucent material approximation from photographs can also be adapted to GPUs, both the computation of the matrices and the linear solving algorithm. This technique would greatly benefit from working at interactive rates as a potential extension to image-editing tools. However, other presented techniques are not so easily adapted and might an algorithm redesign for working on current graphics hardware.

Also, it would be interesting to extend the application of some of these algorithms:

- The algorithm for rendering participating media including inelastic scattering can be potentially used for a generic participating media model. Instead of ocean water, it could be applied to any model if the corresponding data is available.
- The algorithm for simulating the phase function of rainbows is trivially extended to different phase functions, given its capabilities of considering polarization and wave interference. It could be applied to different particle shapes, such as ice, in order to simulate new and interesting phenomena. Also, how would this algorithm be adapted to full scene rendering?
- A database of materials could be created by gathering the estimations obtained from a big set of photographs using the algorithm presented in section 3.3.