BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

RENDERING OF UNDERWATER SCENES

VYKRESLOVÁNÍ PODMOŘSKÝCH SCÉN

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR **Bc. MARTIN SMUTNÝ**
AUTOR PRÁCE

SUPERVISOR **Ing. TOMÁŠ MILET, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2023

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Master's Thesis Assignment

146878

Institut:	Department of Computer Graphics and Multimedia (UPGM)
Student:	**Smutný Martin, Bc.**
Programme:	Information Technology and Artificial Intelligence
Specialization:	Computer Graphics and Interaction
Title:	**Rendering of Underwater Scenes**
Category:	Computer Graphics
Academic year:	2022/23

Assignment:

1. Familiarize yourself with rendering of fluids and underwater scenes. Study algorithms for visualization of water surfaces, computation of light underwater and rendering of lighting effects. Study procedural generation.
2. Design an application that visualizes underwater scenery in high quality.
3. Implement selected graphic techniques and the application.
4. Evaluate the implemented application and create a demonstration video.

Literature:

- Darles Emmanuelle, et. al.: A Survey of Ocean Simulation and Rendering Techniques in Computer Graphics. Computer Graphics Forum - CGF (2011).
- A. Sedlazeck, R. Koch.: Simulating Deep Sea Underwater Images Using Physical Models for Light Attenuation, Scattering, and Refraction. Vision, Modeling, and Visualization (2011)
- Philippe Decaudin and Fabrice Neyret.: Volumetric Billboards. Compute Graphics forum 2009.

Requirements for the semestral defence:
Items 1, 2 and the core of the application.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor:	**Milet Tomáš, Ing., Ph.D.**
Head of Department:	Černocký Jan, prof. Dr. Ing.
Beginning of work:	1.11.2022
Submission deadline:	24.5.2023
Approval date:	31.10.2022

Abstract

This thesis focuses on rendering realistic underwater environments using real-time techniques. Focus is on rendering ocean waters because of their complex and highly variable optical properties. Rendering such media is a challenging field in computer graphics that involves simulation of light scattering inside the water body. Methods in the field of real-time computer graphics for rendering and simulation of oceans are investigated. Single scattering and multi scattering approximation techniques are designed to render a water volume using ray marching. The optical properties of the volume are described by a physically-based bio-optical model. Furthermore, surface and volumetric caustics are rendered as they refract through a simulated ocean surface using Fourier transform. The designed technique to render underwater environments was not implemented in its entirety. Only a certain subset of techniques was experimented with.

Abstrakt

Tato práce se zabývá vykreslováním realistických podvodních scén pomocí technik v reálném čase. Práce se zaměřuje na vykreslování oceanických vod, které mají komplexní a vysoce proměnlivé optické vlastnosti. Vykreslování takových prostředí je náročnou oblastí v počítačové grafice, která zahrnuje simulaci rozptylu světla uvnitř vodního objemu. Jsou zkoumány metody počítačové grafiky pracující v reálném čase pro vykreslování a simulaci oceánů. Jsou navrhnuty techniky aproximující rozptyl první řádu a vícenásobný rozptyl pro vykreslení vodního objemu za pomocí krokování paprsku. Optické vlastnosti objemu jsou popsány pomocí fyzikálně založeného bio-optického modelu. Navíc, jsou vykresleny povrchové a volumetrické kaustiky odpovídající lomu světla skrze simulovaný vodních povrch pomocí Fourierovy transformace. Navržená technika pro vykreslování scén nebyla implementována v plné podobě. Byly implementovány pouze některé navržené techniky a s nimi byly provedeny experimenty zvlášť.

Keywords

real-time rendering, participating media, light scattering, ocean rendering, underwater rendering, water surface, bio-optical model, ray marching, caustics, godrays, single scattering, multiple scattering, implicit surfaces.

Klíčová slova

real-time vykreslování, opticky aktivní prostředí, rozptyl světla, vykreslování oceánu, vykreslování pod vodou, vodní povrch, bio-optický model, krokování paprsku, kaustiky, světelné paprsky, rozptyl prvního řádu, vícenásobný rozptyl, implicitní plochy.

Reference

SMUTNÝ, Martin. *Rendering of Underwater Scenes.* Brno, 2023. Master's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Tomáš Milet, Ph.D.

Rozšířený abstrakt

Odvětví počítačové grafika, do které tato práce patří, se snaží zreprodukovat vizuální element přírodních fenoménů do digitální podoby. Jedním z těchto fenoménů jsou oceány a podmořské prostředí, které jsou známé svými modrozelenými barvami, a blyštivými kaustikami na dně a blízko vodního povrchu. Tyto vizuální fenomény jsou způsobeny interakcí světla s povrchem vody a následně se samotným objemem vody. Obzvláště oceanické vody obsahují plno částic a organizmů, které ovlivňuji jakým způsobem se světlo šíří a rozptyluje ve vodě.

K simulaci těchto efektů vychází počítačová grafika z oborů zabývající se šířením radiace v prostoru (angl. radiative transfer). Fotorealistická počítačová grafika simulující interakce světla s objemy vychází konkrétně z tzv. rovnice přenosu radiace (angl. radiative transfer equation). Tato rovnice je převedena do podoby tzv. vykreslovací rovnice pro objemy (angl. volume rendering equation), která umožňuje vypočítat množství světla podél paprsku světla v každém bodě objemu. Zatímco, fotorealistické metody využívají numerické metody pro výpočet této rovince, metody reálného času se typicky omezují na hrubé aproximace této rovince a věrnější vykreslování objemů je často velice obtížné. Zejména simulátory, a hry přestavují případy, kdy věrnější a zároveň interaktivní zobrazení je vítané až chtěné.

Tato práce se tedy zabývá studiem, návrhem a implementací technik pro účely zobrazování objemů vody ve vysoké kvalitě a navíc se omezuje na techniky reálného času. Vodní objemy, zejména ty oceanické, vykazují složité a velice proměnlivé optické vlastnosti. Tímto představují výzvu pro techniky reálného času.

Nejdříve jsou zavedené radiometrické veličiny, které jsou používány v rámci celé práce. Poté je interakce světla s objemy, pro tyto účely se zavadí abstrakce tzv. opticky aktivního prostředí (angl. participating media). Je zmíněna rovnice přenosu radiace a jak souvisí s rovnicí pro vykreslování objemů. Důraz je kladen na pochopení rovnice pro vykreslování objemů, protože tato rovnice představuje základ, který je aproximován pro dosáhnutí reálného času. Následně je zmíněno jakým způsobem interaguje světlo s povrchy, které ohraničují vodní objem. Jedná se tedy převážně o studium rovnic pro výpočet odrazu a lomu od vodní plochy. Navíc je zmíněna odrazivost materiálů vodního dna.

Důležitá část práce je věnována studiu optických vlastností přírodních vod z pohledu optiky zabývající se vlastnostmi oceánů. Je studováno složení vodního objemu oceanických vod, a jakým způsobem jednotlivé složky ovlivňují šíření světla v objemu. Zejména jsou studovány fyzické procesy ovliňující množství světla v objemu, a to: absorpce a rozptyl. Nakonec jsou oceány rozděleny do kategorií dle optických vlastností.

Dvě kapitoly jsou věnovány studiem a porovnání technik reálného času pro vykreslování a simulace oceánů a efektů uvnitř vodního objemu, tzv. kaustik. Jsou zmíněny následující druhy technik: techniky pro simulaci vodního povrchu, fyzikálně založené modely popisující optické vlastnosti oceanikých vod, techniky pro vykreslování opticky aktivních prostředí, a nakonec techniky procedurální generace a implicitní reprezentace modelů. Druhá kapitola se zabývá studiem technik vykreslující kaustiky v reálném čase. Uvedeny a porovnány jsou zejména techniky pro povrchové kaustiky a pro volumetrické kaustiky.

Na základě nastudovaných technik byla navrhnuta metoda pro vykreslování vodních objemů a povrchů založená na simulaci šíření světla v objemu vody. Výpočet množství světla v každém bodě objemu je použito aproximace rozptylu prvního řádu a vícenásobného rozptylu. Navržená scéna se skládá z atmosferického modelu, vodního povrchu, vodního objemu, a podvodního dna. Byl také navržen fyzikální model popisující optické vlastnosti vodního objemu, který uvažuje vertikální rozložení hodnot optických parametrů, tedy optické parametry se mění se vzdáleností od vodního povrchu. Vícenásobný rozptyl využívá

právě optických vlastností vody pro aproximaci tohoto efektu. Následně byly navrhnuty techniky pro vykreslování povrchových a volumetrických kaustik.

Navržená technika nebyla implementována v celé podobě. Jen podmnožina navržených technik byla implementována a byly s nimi provedeny experimenty. Přednost byla dána rozsáhlému studiu existujících metod v této oblasti.

Rendering of Underwater Scenes

Declaration

I hereby declare that this Master's thesis was prepared as an original work by the author under the supervision of Ing. Tomáš Milet, Ph.D. The supplementary information was provided by Ing. Tomáš Milet, Ph.D. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....
Martin Smutný
May 24, 2023

Acknowledgements

I would like to thank my supervisor Ing. Tomáš Milet, Ph.D. for his valuable advice, helpful suggestions, and for all the extensive consultations. I am also grateful to my family and friends for their support during my studies.

Contents

Chapter 1

Introduction

One of the goals of computer graphics is to reproduce natural phenomena in the digital world. Among the natural phenomena is undeniably are oceanic environments. Specifically, the typical green and blue hues of oceanic waters that can be viewed from above and below the waters, as shown in the photos of Figure 1.1. These typical colors are caused by the complex and highly-variable optical properties in these waters. Sunlight that penetrates into through the ocean surface into the waters causes well-known phenomenon, called *caustics*.

Figure 1.1: The left image is a photo of a tropic island, by Flo Dahm, pexels.com. The right image is a photo of a coral reef in Philippines, by Zir YU, pexels.com.

Reproducing the visual complexity of underwater environments as close to the perceived reality as possible poses a challenge in computer graphics. Offline methods in computer graphics deal with simulation of light as it propagates through such volumes in a straightforward way. On the other hand, real-time methods must look for alternative approaches, that mostly consisted of making simplifications and crude approximations.

The goal of this thesis is then to, first, investigate methods and techniques in the field of real-time rendering, that try to simulate and render oceanic environments in high quality. Generally, these approaches try to simulate the ocean surface and the interactions of light in the water volume. And second, to design and implement an application that is able to render an underwater scenery in high quality.

A physically-based bio-optical model is designed to describe optical parameters of a water volume of oceanic type with non-uniform vertical distribution. A single scattering and multiple scattering approximation technique is proposed to render the volume through the use of ray marching. Finally, the ocean surface is simulated to incorporate physically-based underwater effects, such as caustics.

The rest of thesis is organized as follows. In Chapter 2, general concepts and principles of light scattering by volumes and surface that bound the water body are explained. Chapter 3 studies scattering in natural waters from the point of view of ocean optics. In Chapter 4 the related works for ocean simulation and rendering are summarized, these consist of existing solutions for: ocean surface simulation, modeling of bio-optical parameters, participating media rendering and procedural generation. Lighting effects, such as caustics are subject to Chapter 5. Finally, Chapters 6 and 7 describe the designed and subsequently implemented solution.

Chapter 2

Fundamentals of Light Scattering Theory

This chapter studies the theoretical fundamentals of light scattering in media and from surfaces. The propagation of radiation due to the light being scattered is studied by the field of radiative transfer. It is the radiative transfer equation (RTE) that describes the propagation of light in media. In order to render such media, which is the goal of this thesis, the volume rendering equation (VRE) must be studied. The VRE expresses the RTE in a form that is suitable for rendering media.

The chapter is then organized as follows: Section 2.1 introduces the radiometric quantities, which form the basis of the derivation of the rendering algorithms, that will be used throughout this thesis. Section 2.2 introduces the concept of participating media and studies the radiative transfer in such media to arrive at the VRE. Section 2.3 develops the equations that describe how light is scattered from surfaces that bound the water body, specifically, the planar water surface interface and an opaque sea floor.

2.1 Radiometric and Photometric Quantities

This section provides an overview of radiometric quantities used in rendering based on the work of Pharr et al. [74].

Radiometry is the study of measuring electromagnetic radiation. This radiation propagates as waves in an environment. Electromagnetic waves with different *wavelengths* (λ) have different properties. There is a wide range of wavelengths from gamma waves with less than 10 pm in length to extreme low frequency (ELF) radio waves that are tens of thousands of kilometers long. For rendering, only a small range of these wavelengths is of particular interest, and that is the range that is visible to humans: from approximately 400 nm to 700 nm (see Figure 2.1).

Closely related field that is based on radiometric principles is *radiative transfer*. It is the study of the transfer of radiant energy based on empirical observations. Radiative transfer operates at the level of *geometric optics* (also known as *ray optics*), where macroscopic properties of light are sufficient to describe how light interacts with objects much larger than its wavelength. Incorporating phenomena from wave optics models of light is also not uncommon.

Radiometry was originally built on an abstraction of light based on particles flowing through space. These particles, called photons, are emitted from sources of illumination.

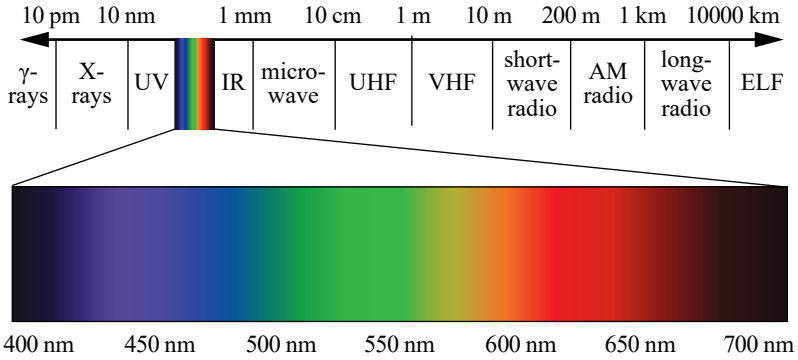


Figure 2.1: The visible spectrum of wavelengths in the context of the full electromagnetic spectrum. (*Image adapted from Real-Time Rendering, 4th edition [5].*)

Each photon at a particular wavelength carries a particular amount of *energy* Q measured in joules [J]. There are four radiometric quantities that are central to rendering. These fundamental radiometric quantities effectively provide different ways of measuring energy of photons. All of them are generally wavelength-dependent, though that dependence is not explicit in this chapter.

Radiant flux (Φ) is the amount of energy passing through a region of space over time, also known as *power*, and is measured in watts [W] or joules per second [J/s]. It is defined as the limit of differential energy per differential time:

$$\Phi = \lim_{\Delta t \to 0} \frac{\Delta Q}{\Delta t} = \frac{\mathrm{d}Q}{\mathrm{d}t}. \tag{2.1}$$

Flux is generally used to describe the total emission from a light source. Figure 2.2 illustrates the flux and shows an example of flux being emitted from a point light source measured at spheres that surround the light. The total amount of flux measured on the smaller sphere is the same as on the bigger sphere — although less energy passes through any local region of the larger sphere.

Figure 2.2: Radiant flux Φ shown in the left image (*from Novák [69]*). The right image (*from Pharr et al. [74]*) illustrates radiant flux emanating from a point light source measured at spheres surrounding the light.

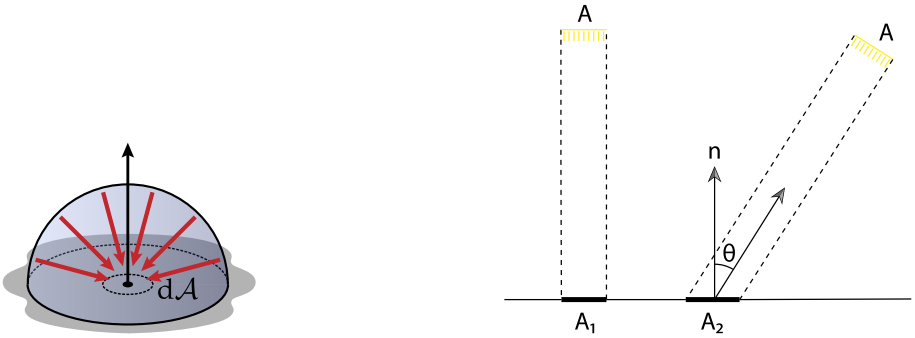
Irradiance (E) can be defined as the average density of power over a finite area A, as illustrated in Figure 2.3a. This quantity is referred to as either irradiance, the area density of flux arriving at a surface, or *radiant exitance* (M), the area density of flux leaving

a surface. In the general case, these both terms can be defined as the differential flux per differential area at point $\mathbf{x}$:

$$E(\mathbf{x}) = \frac{\mathrm{d}\Phi(\mathbf{x})}{\mathrm{d}A}. \tag{2.2}$$

For the point light source example in Figure 2.2, the area density of radiant flux incident on a sphere depends on its radius r: $E = \Phi/4\pi r^2$. In other words, irradiance at any point on the inner sphere is greater than the irradiance at any point on the outer sphere, since the surface area of the outer sphere is larger.

In case of a light source with area A and flux Φ that is illuminating a surface with normal n, as illustrated in Figure 2.3b, the irradiance equation shows so-called *Lambert's law*. If the light is shining directly down on the surface then the illuminated area on the surface A_1 is equal to A. Irradiance at any point inside A_1 is then $E = \Phi/A$. However, when the light is shining at an angle on the surface, the illuminated area on the surface is larger. For small A, the area receiving flux, A_2, is roughly $A/\cos\theta$. For any point inside A_2, the irradiance is then $E_2 = (\Phi\cos\theta)/A$, in [W/m^2].



(a) Irradiance $E(\mathbf{x})$ as the differential flux over a differential area A. (*Taken from Novák [69].*)

(b) Lambert's law: irradiance incident at a surface is proportional to the cosine of the angle between the light direction and the surface normal. (*Image from Pharr et al. [74].*)

Figure 2.3: The left image illustrates the irradiance, the right image explains the Lambert's law.

Radiant intensity (I) is the angular density of power emitted by an infinitesimal light source centered within the unit sphere. This quantity is measured in [W/sr], where *sr* (*steradians*) is the unit to measure solid angles, for example, the entire unit sphere is a solid angle of 4π sr, and a hemisphere is 2π sr. Intensity over the entire sphere of directions is then:

$$I = \frac{\Phi}{4\pi}. \tag{2.3}$$

Figure 2.4: Radiant intensity $I(\omega)$ (*from Novák [69]*)

More generally, it is defined as the differential flux emitted along a differential cone of directions (shown in Figure 2.4):

$$I = \frac{\mathrm{d}\Phi}{\mathrm{d}\omega}, \tag{2.4}$$

where ω is a normalized vector indicating the directions on the unit sphere.

Radiance (L) measures irradiance (or radiant exitance) with respect to solid angles, it is a measure of electromagnetic radiation in a single ray. In comparison to irradiance, which gives the differential power per differential area at point $\mathbf{x}$, radiance additionally distinguishes the directional distribution of power. It is defined as:

$$L(\mathbf{x},\omega) = \frac{\mathrm{d}E_\omega(\mathbf{x})}{\mathrm{d}\omega} = \frac{\mathrm{d}\Phi(\mathbf{x},\omega)}{\mathrm{d}A^\perp \mathrm{d}\omega}, \tag{2.5}$$

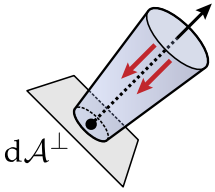


Figure 2.5: Field radiance (*from Novák [69]*)

where E_ω denotes irradiance at the surface that is perpendicular to the direction ω, hence the $\cos\theta$ term from Lambert's law is hence unnecessary. This type of radiance is sometimes more specifically called *field radiance*. In terms of flux, radiance is the density of flux with respect to both area and solid angle, then $dA^\perp$ is the projected area dA on a hypothetical surface perpendicular to ω as shown in Figure 2.5.

Radiance is the final and most fundamental of all radiometric quantities. If radiance is given, then all other quantities can be computed as integrals of radiance over areas and directions.

In shading equations, radiance often appears in two forms: *incident radiance* and *outgoing radiance* (as illustrated in Figure 2.6). The incident radiance, denoted by $L_i(\mathbf{x},\omega)$, describes radiance arriving at point $\mathbf{x}$ on the surface of an object from direction ω. The outgoing (or exitant) reflected radiance from the surface at that point is denoted by $L_o(\mathbf{x},\omega)$. In both cases, by convention, the direction vector ω points away from $\mathbf{x}$.

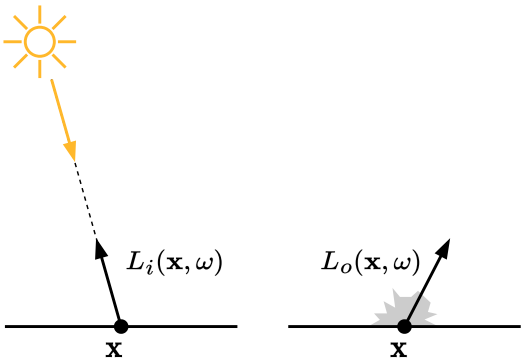


Figure 2.6: The incident radiance function $L_i(\mathbf{x},\omega)$ describes the distribution of radiance arriving at a point as a function of position and direction. The exitant radiance function $L_o(\mathbf{x},\omega)$ describes the distribution of radiance leaving the point. By convention, ω points away from the surface. (*Image adapted from Pharr et al. [74].*)

While radiometry deals with purely physical quantities, a related field, called *photometry*, takes into account the sensitivity of the human eye. It is the study of visible electromagnetic radiation in relation to its perception by the human visual system. For each radiometric quantity there is a corresponding photometric quantity. To compute a photometeric quantity, a spectral radiometric quantity is integrated against the spectral response curve, which describes the relative sensitivity of the human eye to various wavelengths.

Luminance is often used as a measure of how bright flat surfaces appear to a human eye. For example, street lighting has a luminance around 10 nits, high dynamic range (HDR) screens typically range from 500 to 1000 nits, while clear sky has a luminance of about 8000 nits. Table 2.1 provides a summary of radiometric quantities and their photometric equivalents.

Radiometric		**Unit**	**Photometric**	**Unit**
Radiant energy	Q	joule (J)	Luminous energy	talbot (T)
Radiant flux	Φ	watts (W)	Luminous flux	lumen (lm)
Intensity	I	W/sr	Luminous intensity	lm/sr = candela (cd)
Irradiance	E	W/m^2	Illuminance	lm/m^2 = lux (lx)
Radiance	L	W/(m^2 sr)	Luminance	lm/(m^2 sr) = cd/m^2 = nit

Table 2.1: Radiometric quantities and their corresponding photometric quantities.

2.2 Interaction of Light with Media

Computer graphics, for convenience reasons, often assume that scenes are composed of solid objects separated by vacuum, however, this assumption is inaccurate in many real-world situations: fog, smoke, and earth's atmosphere attenuate and scatter light from particles that are distributed throughout the space. Volumes or regions in 3D space with a large amount of very small particles inside that participate in the overall light transport are called *participating media* (or simply "media"). And it is these interactions that give media and physical object the appearance we observe. Figure 2.7 shows examples of such media in the real-world.

Figure 2.7: Examples of various types of participating media. Credit: second image from the left by Roban Kramer, flicker.com; the third image from the left by Ferdinand Kozeluh, pixabay.com.

To goal of this section is to arrive at volume rendering equation, an equation that is used to render participating media. For that, first, the properties of participating media are introduced, then the radiative transfer equation (RTE) is formulated. The last subsections deal with the terms used in VRE and with the various models of scattering.

This section is based on the information found in literature that studies volume rendering in production: Fong et al. [28], Chapter 11 from Pharr et al. [75].

2.2.1 Properties of Participating Media

Volumes or media are a collection of particles, ranging from atoms and molecules to any particle size. To simulate light transport in these media, interactions between the particles and photons of light are modeled as probabilistic processes. Under certain assumptions the collisions of the photons with the particles can be viewed as statistically independent.

When photons travel through such a volume, they collide with the particles that make up the volume. The results of these collisions define the radiance distribution throughout the volume. Instead of simulating each collision separately, the collisions are modeled as probability fields. The chance that a photon collides with a particle at location $\mathbf{x}$ is

defined by a coefficient $\sigma(\mathbf{x})$, the probability density of collision per unit distance traveled in the medium. Its units are reciprocal distance (m^{-1}) and it is usually a spectrally varying quantity.

There are three main physical processes happening when photon collides with a particle (illustrated in Figure 2.8) that affect the distribution of radiance inside the volume:

- *Absorption*: describes the collision when a photon is absorbed by the volume, expressed with the absorption coefficient $\sigma_a(\mathbf{x})$. This causes a reduction in the amount of radiance along a ray. Physically, the energy of the photon is converted to another form of energy, such as heat.
- *Scattering*: describes when a photon as a result of a collision is scattered into a different direction defined by the *phase function*. The radiance along a ray can either decrease due to the photons being deflected out of the ray, also called *out-scattering*, or it can increase when photons are scattered into the path of the ray, called *in-scattering*. Both of these effects are expressed using the scattering coefficient $\sigma_s(\mathbf{x})$.
- *Emission*: causes an increase in radiance due to chemical, thermal, or nuclear processes that convert energy of the photon into visible light. The change in radiance is also determined by how efficiently is light absorbed and emitted by the volume.

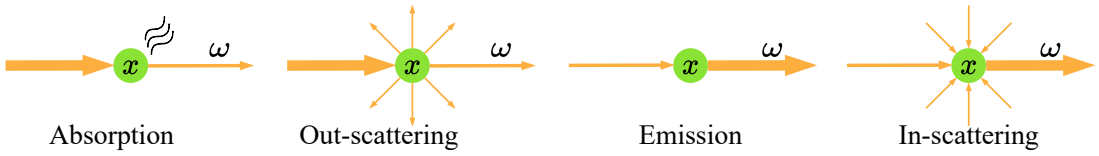


Figure 2.8: Illustration of the different events that can happen when light ray propagating in direction ω collides with a particle at location $\mathbf{x}$ resulting in a change in radiance.

The properties of a volume describing how light interacts with the volume are defined by absorption and scattering coefficients, together with the phase function and emission. These properties can be constant, then the medium is called *homogeneous* medium, or they can vary throughout space, then the medium is *inhomogeneous*. The described processes may also behave differently at different wavelengths of light.

As a collective term for the total reduction in radiance due to absorption and out-scattering, so-called *attenuation* (or *extinction*, also informally referred to as the density), is introduced. Physically, it can be interpreted as a collision where both scattering and absorption are involved. It is defined as sum of the absorption and scattering coefficients:

$$\sigma_t = \sigma_a + \sigma_s. \tag{2.6}$$

Related to the extinction coefficient is the *single-scattering albedo* or simply *medium's albedo* ρ, defined as:

$$\rho = \frac{\sigma_s}{\sigma_s + \sigma_a} = \frac{\sigma_s}{\sigma_t}. \tag{2.7}$$

It is a measure of the medium's reflectivity. It determines the amount of scattered radiance in the medium. In another words, it is the ratio or probability of scattering versus absorption

at a scattering event. The value of ρ is in range [0, 1]: a value close to 0 indicates that most of the light is absorbed, resulting in a murky medium, such as dark exhaust smoke; a value close to 1 indicates that instead of the light being absorbed, it is mostly scattered, resulting in a brighter medium, such as clouds (as shown in the rightmost image in Figure 2.7), or the earth's atmosphere. Water is another example of a medium that is highly absorptive, as shown in the second image from the left in Figure 2.7. Figure 2.9 illustrates the effects of absorption and scattering on color of a liquid inside a container. Highly scattering and low absorptive liquids exhibit milky appearance, while highly absorptive liquids with low scattering are very murky. The combination of both high absorption and scattering gives a muddy appearance.

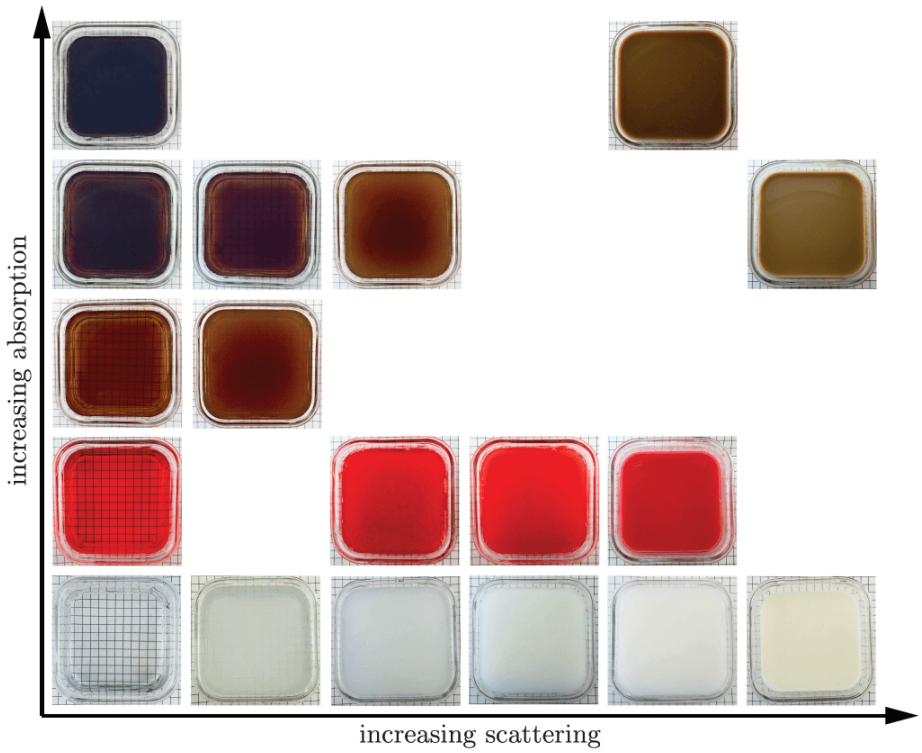


Figure 2.9: Photos of containers with liquid with varying absorption and scattering. (*Image from Real-time Rendering, 4 th edition [6].*)

2.2.2 Light Propagation in Participating Media

As mentioned in the previous section, there are three events that affect the radiance along a light ray propagating through a medium: absorption, scattering, and emission. Then the change of radiance along the light ray traveling in direction ω from point $\mathbf{x}$ due to these events is given by the *radiative transfer equation* (RTE) [21]:

$$\frac{\mathrm{d}L(\mathbf{x},\omega)}{\mathrm{d}\mathbf{x}} = \underbrace{\sigma_a(\mathbf{x})L_e(\mathbf{x},\omega)}_{\text{emission}} + \underbrace{\sigma_s(\mathbf{x})L_i(\mathbf{x},\omega)}_{\text{in-scattering}} - \underbrace{\sigma_a(\mathbf{x})L(\mathbf{x},\omega)}_{\text{absorption}} - \underbrace{\sigma_s(\mathbf{x})L(\mathbf{x},\omega)}_{\text{out-scattering}} \tag{2.8}$$

$$= \underbrace{L_v(\mathbf{x},\omega)}_{\text{contribution}} - \underbrace{\sigma_t(\mathbf{x})L(\mathbf{x},\omega)}_{\text{extinction}} \tag{2.9}$$

The RTE defines the distribution of radiance in media. In other words, the derivative in radiance along the ray as the amount of radiance lost due to absorption, or out-scattering, denoted by the extinction term, and the amount of radiance gained, L_v, due to in-scattering (light coming from any direction scattered towards ω) and emission [28].

The RTE provides an intuitive way to understand light transport in media. In order to compute the medium contribution for a ray of given length the RTE needs to be integrated into the *volume rendering equation* (VRE). The integration of RTE to arrive at VRE is studied, for instance, in Fong et al. [28].

Then the general VRE can be defined in the following form:

$$L(\mathbf{x},\omega) = \underbrace{T(\mathbf{x},\mathbf{x}_s)L_s(\mathbf{x}_s,\omega)}_{\text{surface contribution}} + \underbrace{\int_{t=0}^{S} T(\mathbf{x},\mathbf{x}_t)L_v(\mathbf{x}_t,\omega)\ \mathrm{d}t}_{\text{volume contribution}}, \tag{2.10}$$

$$L_v(\mathbf{x},\omega) = \sigma_a(\mathbf{x})L_e(\mathbf{x},\omega) + \sigma_s(\mathbf{x})L_i(\mathbf{x},\omega). \tag{2.11}$$

It is the measurement of the total radiance $L(\mathbf{x},\omega)$ along a ray propagating in the direction ω starting from $\mathbf{x}$. This is illustrated in Figure 2.10. The integrals are parameterized along the negative part of the ray, with the increasing ray parameter t ending at a boundary (surface or other volume) with the ray parameter S. The surface contribution term expresses the contribution due to the incident radiance at the end of the ray L_s at point $\mathbf{x}_s = \mathbf{x} - S\omega$ that is attenuated by the *transmittance* function T (described in Section 2.2.3). The integral on the right side accumulates all the emitted or in-scattered radiance (described in Section 2.2.4) by the volume into ω, attenuated by T at the parameterized location $\mathbf{x}_t = \mathbf{x} - t\omega$.

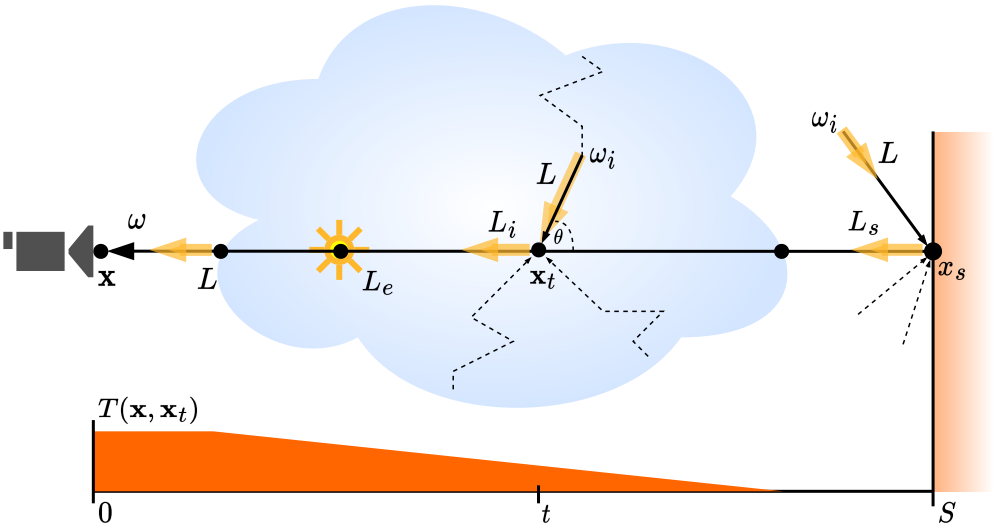


Figure 2.10: Illustration of the volume rendering equation. A ray is propagating along a negative direction ω starting at point $\mathbf{x}$ with parameter t along the points $\mathbf{x}_t = \mathbf{x} - t\omega$ ending at a surface point $\mathbf{x}_s = \mathbf{x} - S\omega$. The radiance at point $\mathbf{x}$ is the result of the incident radiance L_s at $\mathbf{x}_s$, the emitted L_e and the in-scattered radiance L_i all attenuated by the medium, depicted as a cloud-like shape.

The VRE describes the radiance in terms of where it comes from, rather than where it is traveling to, as expressed by the RTE. The VRE "looks" backwards accumulating all

of the contributions with integrals while the direction ω points in the direction of radiance flow.

The incident radiance at a surface point in direction ω can be expressed as

$$L_s(\mathbf{x}, \omega) = \sigma_a(\mathbf{x}) L_e(\mathbf{x}, \omega) + \int_{\Omega^+} f_r(\mathbf{x}, \omega, \omega_i) L(\mathbf{x}, \omega_i)(\mathbf{n} \cdot \omega_i) \, \mathrm{d}\omega_i, \tag{2.12}$$

where f_r is a hemispherical function called the *bidirectional reflectance distribution function* (BRDF), a surface equivalent to phase function, and $\mathbf{n}$ is the surface normal. The surface contribution is gathered from the upper hemisphere of directions Ω^+, in respect to $\mathbf{n}$. In the absence of any participating medium, i.e., $\sigma_a = \sigma_s = 0$, and $L_e = 0$, the volume contributions are zero hence giving the standard rendering equation for surfaces (Kajiya, 1986 [47]). Section 2.3.3 describes the BRDF in more detail. Transmittance function T and the terms related to scattering, such as in-scattering L_i and phase function are further described in the sections below.

2.2.3 Transmittance

The extinction term in the RTE (in Equation 2.9) describes the amount of radiance lost along a ray due to out-scattering or absorption,

$$\frac{\mathrm{d}L(\mathbf{x}, \omega)}{\mathrm{d}\mathbf{x}} = -\sigma_t(\mathbf{x}) L(\mathbf{x}, \omega), \tag{2.13}$$

can be computed using the *transmittance* T, which gives the fraction of radiance that is transmitted (able to get through) between two points in a medium:

$$T(\mathbf{x}_a, \mathbf{x}_b) = e^{-\tau} \text{ , where } \tau = \int_{\mathbf{x}=\mathbf{x}_a}^{\mathbf{x}_b} \sigma_t(\mathbf{x}) ||\mathrm{d}\mathbf{x}||. \tag{2.14}$$

This relationship is known as the *Beer-Lambert law*. The *optical thickness* τ (or sometimes *optical depth*) is unitless and represents the amount of light attenuation. For a constant σ_t in a homogeneous medium, the integral can be evaluated, giving simply *Beer's law*:

$$T(\mathbf{x}_a, \mathbf{x}_b) = e^{-\sigma_t ||\mathbf{x}_a - \mathbf{x}_b||}. \tag{2.15}$$

Transmittance has certain observable properties. It is always between 0 and 1. Transmittance from and to the same point is 1, $T(\mathbf{x}, \mathbf{x}) = 1$, and in vacuum $\sigma_t = 0$ and so $T(\mathbf{x}, \mathbf{x}') = 1$ for all $\mathbf{x}'$. If the attenuation coefficient satisfies the directional property $\sigma_t(\omega) = \sigma_t(-\omega)$, i.e., it does not vary with direction but can vary with position, then the transmittance between the same points, $T(\mathbf{x}_a, \mathbf{x}_b) = T(\mathbf{x}_b, \mathbf{x}_a)$, is the same in both directions. Furthermore, transmittance is multiplicative along points on a ray in any media:

$$T(\mathbf{x}_a, \mathbf{x}_c) = T(\mathbf{x}_a, \mathbf{x}_b) T(\mathbf{x}_b, \mathbf{x}_c) \tag{2.16}$$

for all points $\mathbf{x}_b$ between $\mathbf{x}_a$ and $\mathbf{x}_c$. This is a useful property for implementations, since it allows for computation of transmittance between two bounding points along a ray by taking the product of transmittance at multiple points in between the bounding points along the same ray. Figure 2.11 shows the transmittance function at various depths for different values of the extinction coefficient.

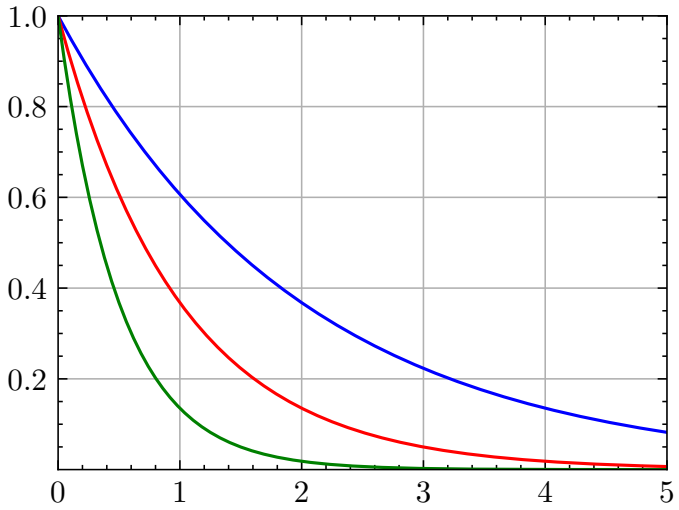


Figure 2.11: Transmittance function as a function of depth and extinction $\sigma_t = 0.5$ (blue), $\sigma_t = 1.0$ (red), and $\sigma_t = 2.0$ (green). The higher the extinction, the less light travels through medium.

2.2.4 Scattering and Phase Functions

Both the radiative transfer equation (RTE) and volume rendering equation (VRE) (in Equations 2.9, 2.11) contain the out-scattering and in-scattering terms. While out-scattering accounts for reduction in radiance along a ray due to scattering in different directions, in-scattering accounts for an increase in radiance due to scattering from other directions (as illustrated in Figure 2.10):

$$L_i(\mathbf{x}, \omega) = \sigma_s(\mathbf{x}) \int_\Omega p(\mathbf{x}, \omega, \omega_i) L(\mathbf{x}, \omega_i) \, \mathrm{d}\omega_i, \tag{2.17}$$

where Ω denotes the spherical domain around the position $\mathbf{x}$ and p is the phase function. The σ_s in front of the integral models the incoming radiance of all the directions that scattered into the direction of the ray.

Phase function models the scattering of light at a macro level. It describes the probability and angular distribution of scattering directions. In other words, it is the probability that a light's forward direction of travel will scatter in a given direction. Figure 2.12 illustrates how the light, traveling in forward direction, is scattered at point x based on a distribution of scattering directions characterized by the phase function $p(\theta)$ (in red) with two lobes — a smaller backward-scattering and larger forward-scattering lobe. The light is scattered backwards to camera A at angle, θ_A, between the light's direction (orange arrow) and the direction to the camera. Analogously, the light is scattered also to camera B, but in greater amounts due the larger forward-scattering lobe.

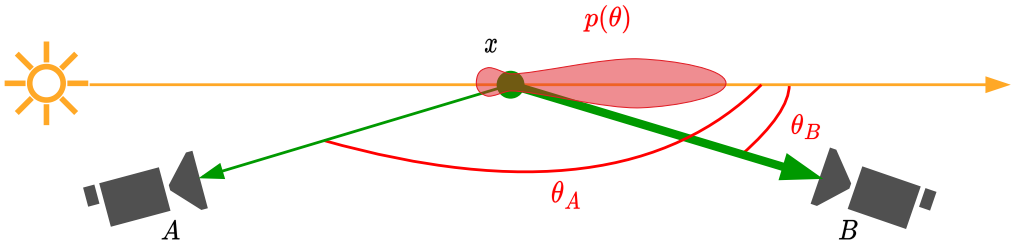


Figure 2.12: Light is scattered according to a phase function (in red). Camera A receives a small portion of light due to backward-scattering lobe, while camera B receives a larger amount due to stronger forward-scattering lobe.

For a phase function to be a probability distribution it must be energy-conserving and -preserving. A phase function p between two directions ω and ω' is normalized over the sphere domain Ω:

$$\int_{\Omega} p(\omega, \omega') \, \mathrm{d}\omega' = 1. \tag{2.18}$$

Phase functions have also the property called *reciprocity*, which means, that the interchange of the two directions ω and ω' results in the same value of the phase function.

There are two types of phase functions based on the modeled angular distributions: isotropic and anisotropic. *Isotropic* phase functions have an equal probability of scattering incoming light into any direction independent of the two input directions. The *isotropic phase function* is one of the simplest phase functions:

$$p(\theta) = \frac{1}{4\pi}. \tag{2.19}$$

Anisotropic phase functions can exhibit complicated angular distribution. These are typically modeled accurately by *Mie scattering* or approximated by *Rayleigh scattering*. Depending on the relative size of a particle $s_p = \frac{2\pi r}{\lambda}$, with radius r and the considered wavelength of light λ, different types of scattering are identified:

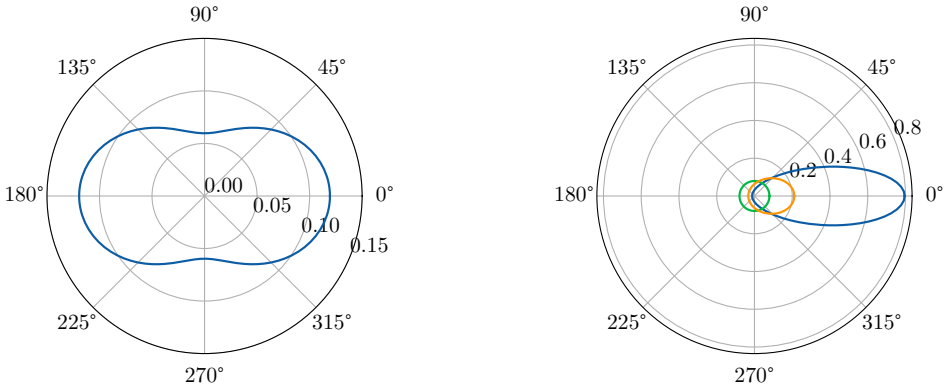
- $s_p \ll 1$: Rayleigh scattering (e.g. air),
- $s_p \approx 1$: Mie scattering,
- $s_p \gg 1$: Geometric scattering.

There are also other components that can influence the scattering result, such as the index of refraction, participating media content, etc.

Rayleigh Scattering was derived to model scattering of light from molecules in the air and describes scattering by spherical particles that are much smaller than the wavelength of the light. It is usually used to describe light scattering in the earth's atmosphere. The function (in Equation 2.20) has two lobes, as shown in Figure 2.13a, a backward and forward scattering one, relative to the light direction.

$$p(\theta) = \frac{3}{16\pi}(1 + \cos^2 \theta). \tag{2.20}$$

Rayleigh scattering is also highly wavelength-dependent. Scattering coefficient of Rayleigh scattering shows that short wavelengths, such as blue or violet light, are scattered much more than long wavelengths, such as red light [40].



(a) Polar plot of the Rayleigh phase function as a function of angle θ between the incident light and out-scattering directions. It is a two-lobed function with the same chance of forward and backward scattering.

(b) Polar plot of the Henyey-Greenstein phase as a function of angle θ. The parameter g increases from 0 (green) to 0.3 (orange) and 0.6 (blue), resulting in a strong forward scattering lobe to the right.

Figure 2.13: Polar plots of Rayleigh (on the left) and Henyey-Greenstein (on the right) phase functions, representing Rayleigh scattering and Mie scattering, respectively.

Mie Scattering is used to model scattering by particles that are about the same size as the light's wavelength. Phase function for a specific particle modeled by Mie scattering has typically complex distribution with strong and sharp directional lobes making them expensive to compute. Instead, such phase functions are most commonly approximated by *Henyey-Greenstein* (HG) phase function [38]. HG was designed to be a good fit to measured scattering data that exhibit strong backward or forward scattering. For instance, smoke, fog, or dust-like participating media. The HG phase function is evaluated using

$$p_{HG}(\theta, g) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 - 2g\cos\theta)^{1.5}}, \tag{2.21}$$

where the parameter $-1 \leq g \leq 1$ is used to control the asymmetry of the phase function or the distribution of the scattered light in the following directions: backward ($g < 0$), isotropic ($g = 0$), or forward ($g > 0$). Figure 2.13b shows a polar plot of the HG with the parameter g increasing from 0 (green) to 0.3 (orange) and 0.6 (blue). To represent more complex phase functions, for example, with forward and backward scattering lobes at the same time, it is possible to blend between multiple HG [40].

Geometric Scattering occurs for very large particles and it is usually simulated on a macro level using complex scattering phase functions. Rainbow effect is an example of this type of scattering. It is caused by internal reflection of light inside water particles in the air, dispersing the sun light into a visible spectrum over a small visual angle (≈ 3 degrees) of the resulting backward scattering [40].

2.3 Interaction of Light with Surfaces Bounding Water Body

This section develops the equations that describe how light is scattered by surfaces that bound the water body, such as water surface or an opaque sea floor. Section 2.3.1 discusses the basics of reflection and refraction of light by a level water surface. Section 2.3.2 then describes the equations that govern the amount of light that is transmitted and reflected. Lastly, Section 2.3.3 introduces the bidirectional reflectance distribution function, the fundamental quantity describing how an opaque surface reflects light. Most of the information in this section is based on Chapter *The Level Sea Surface* from Ocean Optics Web Book [63], and on Chapter 9 from Real-Time Rendering, 4th edition [7]. The rest is cited where appropriate.

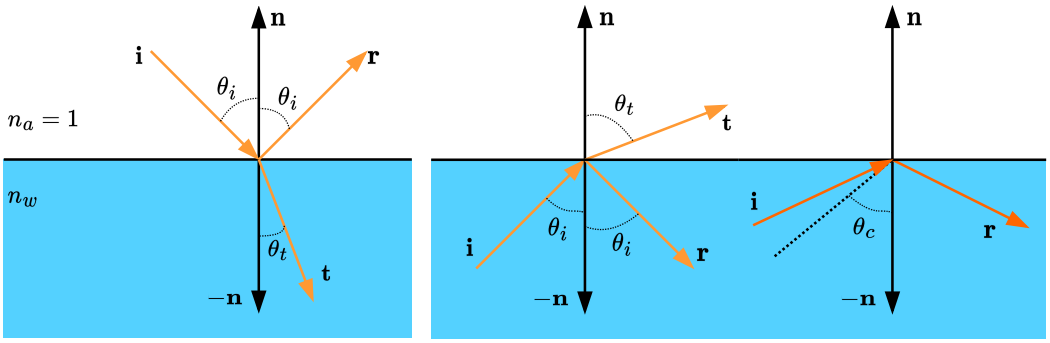
The concepts described below are modeled from the point of view of geometric optics (i.e., light is abstracted in form of rays). Any wavelength-dependency of these concepts is taken as implicit. Geometric optics provides sufficient level of abstraction that allows a real-world wind-blown sea surface to be modeled as a collection of locally flat patches that can be arbitrarily tilted.

2.3.1 Interaction of Light with Planar Air-Water Interface

From an optical perspective, an object surface is represented by a two-dimensional interface separating two volumes. When a ray of light strikes a surface, two aspects of the surface determine the resulting behavior: the substances on either side and the geometry of the surface. In this section an air-water interface, i.e. a level water surface separating air and water, is considered as an example of a perfectly plat plane (as illustrated in Figure 2.14a).

The media on each side are described by an optical property, called *index of refraction* (IOR). The IOR is an abstraction of the interactions of light at molecular level. It defines how the medium affects light of a given wavelength. In its full form, it is defined as a complex number $n + i\kappa$, where κ is the *attenuation index* and it refers to the absorption of the medium, and the real part of the IOR, denoted by n, determines the direction of the refracted ray. Only the real part of IOR will be considered in any further relations. For an interface separating two media, the incident light can originate on either side of interface. The difference in values of the indices of refraction causes a discontinuity in density, which causes the incident light to scatter differently based on the different wavelengths that make up the light (this wavelength-dependency is taken as implicit and is not explicitly represented).

Figure 2.14a illustrates a planar surface with normal $\mathbf{n}$ separating two media: air, and water, denoted by indices of refraction n_a, and n_w, respectively. A ray of light, originating on the air side, strikes the surface from direction $\mathbf{i}$ causing the ray to reflect and refract in directions $\mathbf{r}$ and $\mathbf{t}$, respectively. Because the directions of the incident, reflected and refracted rays lie in the same plane, the amount of incident light is divided into its reflected and refracted directions. Figure 2.14b illustrates the analogous case but now the light is incident from the water side.



(a) Light incident on air-water interface is reflected and refracted.

(b) Light incident on water-air interface (orange) is reflected and refracted. The red (or darker orange) direction shows total internal reflection when the incident angle θ_i is greater than the critical angle θ_c.

Figure 2.14: Illustrates a planar interface between air and water with indices of refraction n_a and n_w, respectively. The left image shows the light incident from the air side, while the right image shows light incident from the water side. Incident, reflected, and refracted directions are denoted $\mathbf{i}, \mathbf{r}$, and $\mathbf{t}$, respectively, and $\mathbf{n}$ is the surface normal.

The angle between the incident direction and the normal is called the *incident angle*:

$$\theta_i = \cos^{-1}(|\mathbf{i} \cdot \mathbf{n}|). \tag{2.22}$$

Based on the *Law of Reflection*, the reflected angle is always equal to the incident angle: $\theta_r = \theta_i$.

The incident and transmitted angles are related by *Snell's law*:

$$n_1 \sin\theta_1 = n_2 \sin\theta_2, \tag{2.23}$$

where subscripts 1 and 2 refer to the media on the side where the incident light originates, and on the side the light is transmitted to, respectively.

For air-incident light, as illustrated in Figure 2.14a, the IOR of air is: $n_1 = n_a = 1$ and the IOR of water is $n_2 = n_w$ with $n_a < n_w$. According to Snell's law, now simplified as $\sin\theta_i = n_w \sin\theta_t$, then the angle of transmission θ_t is given by:

$$\theta_t = \sin^{-1}\left(\frac{1}{n_w}\sin\theta_i\right). \tag{2.24}$$

The reflected and refracted direction are given by the following equations:

$$\mathbf{r} = \mathbf{i} - 2(\mathbf{i} \cdot \mathbf{n})\mathbf{n}, \tag{2.25}$$

$$\mathbf{t} = \frac{1}{n_w}(\mathbf{i} - c\mathbf{n}), \tag{2.26}$$

where $c = \mathbf{i} \cdot \mathbf{n} + \sqrt{(\mathbf{i} \cdot \mathbf{n})^2 + n_w^2 - 1}$.

For the water-incident case, as illustrated in Figure 2.14b, Snell's law reads $n_w \sin\theta_i = \sin\theta_t$, which gives the angle of transmission as:

$$\theta_t = \sin^{-1}(n_w \sin\theta_i). \tag{2.27}$$

The relations between the reflected and refracted directions are then:

$$\mathbf{r} = \mathbf{i} - 2(\mathbf{i} \cdot \mathbf{n})\mathbf{n}, \tag{2.28}$$

$$\mathbf{t} = n_w \mathbf{i} - c\mathbf{n}, \tag{2.29}$$

where $c = n_w \mathbf{i} \cdot \mathbf{n} - \sqrt{(n_w \mathbf{i} \cdot \mathbf{n})^2 - n_w^2 + 1}$.

In the case where $n_1 < n_2$, as seen in the air-water interface in Figure 2.14a, the incident light originates on the side with the lower IOR, this results in a reflection that is called *external reflection*. In the other case, where the light originates on the denser water side (as shown in Figure 2.14b), when $n_1 > n_2$ this case of reflection is called *internal reflection*.

In most cases, when internal reflection occurs the light is transmitted from the more dense medium to the less dense. However, for certain values of the incident angle there is no refraction and all light incident onto the water side is reflected back into the water. This is called *total internal reflection*. This phenomenon occurs for values of incident angle θ_i that are greater than the *critical angle for total internal reflection*,

$$\theta_c = \sin^{-1}\left(\frac{1}{n_w}\right), \tag{2.30}$$

in these cases there is no real solution for the inverse sine in Equation 2.27. The right part of Figure 2.14b illustrates this with the dotted line representing the critical angle.

For the air-water case, when $\theta_i = 0°$, the incident light is perpendicular to the surface, $\mathbf{i} = \mathbf{n}$, this is called *normal incidence*. When light strikes the surface at grazing incident angle of $\theta_i = 90°$ then it is transmitted into the water at the critical angle: $\theta_t = \theta_c$. This causes the light from the entire sky to be transmitted through the surface into a cone of half angle θ_c, which is known as *Snell's cone* or *Snell's window* (shown in Figure 2.15).

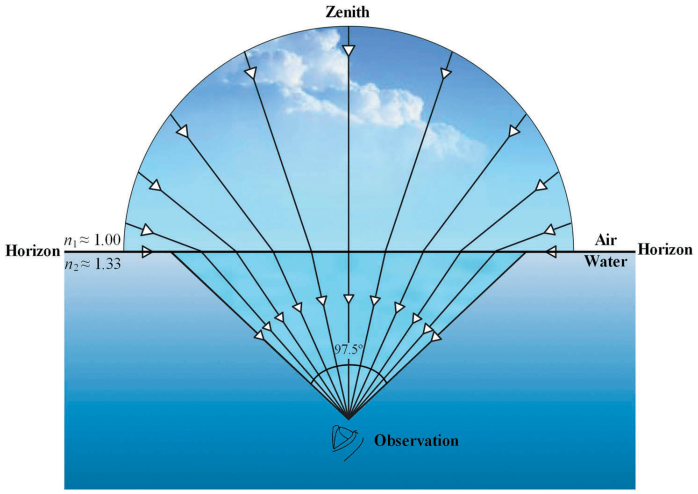


Figure 2.15: Illustrates the incident light being refracted into a cone of directions causing a phenomenon called *Snell's window*. (*Image from Cheng et al. [22].*)

2.3.2 Fresnel's Equations for Unpolarized Light

Previous section described the relations between the angles and directions of the incident, reflected and transmitted light. However, the amount reflected or transmitted light was not addressed. This information is given by *Fresnel's equations*.

The same notations and assumptions from the previous section also hold here: a planar interface, e.g., the flat water surface separating air and water (as shown in Figure 2.14a, both characterized by indices of refraction $n_a = 1$ and n_w, respectively), and a beam of unpolarized incident light with some irradiance measured on a surface normal to the direction of propagation. The incident light on the flat surface is split into a reflected part and a refracted part. The fraction of this incident irradiance that is reflected by the air-water surface is given by the *Fresnel reflectance* R_F:

$$R_F(\theta_i) = \frac{1}{2}\left\{\left[\frac{\sin(\theta_i - \theta_t)}{\sin(\theta_i + \theta_t)}\right]^2 + \left[\frac{\tan(\theta_i - \theta_t)}{\tan(\theta_i + \theta_t)}\right]^2\right\}, \tag{2.31}$$

which is in range 0 to 1, and holds for $\theta_i \neq 0$. For normally incident light, i.e. $\theta_i = 0$, the reflectance is

$$R_F(\theta_i = 0) = \left(\frac{n_w - 1}{n_w + 1}\right)^2. \tag{2.32}$$

The reflectance for normal incidence can be thought of as the characteristic specular color of the substance. Although, the value of $R_F(\theta_i)$ is dependent only on the incident angle, in reality, it varies continuously over the visible spectrum. The general Fresnel equations also use the complex part of IOR, however for water at near-UV to near-IR wavelengths (see Figure 2.1 for the whole electromagnetic spectrum) the difference is negligible.

Due to the conservation of energy, the reflected and transmitted energy must equal the incident energy. Therefore the transmitted fraction of the incident light is $R_T = 1 - R_F$.

Regarding the air- and water-incident light in the cases described in the previous section, Equations 2.31 and 2.32 hold for both of these cases. For water-incident light that undergoes total internal reflection, i.e., $\theta_i \geq \theta_c$, then $R_F(\theta_i) = 1$. Figure 2.16 shows the Fresnel reflectance for the range of indices of refraction of water at visible wavelengths.

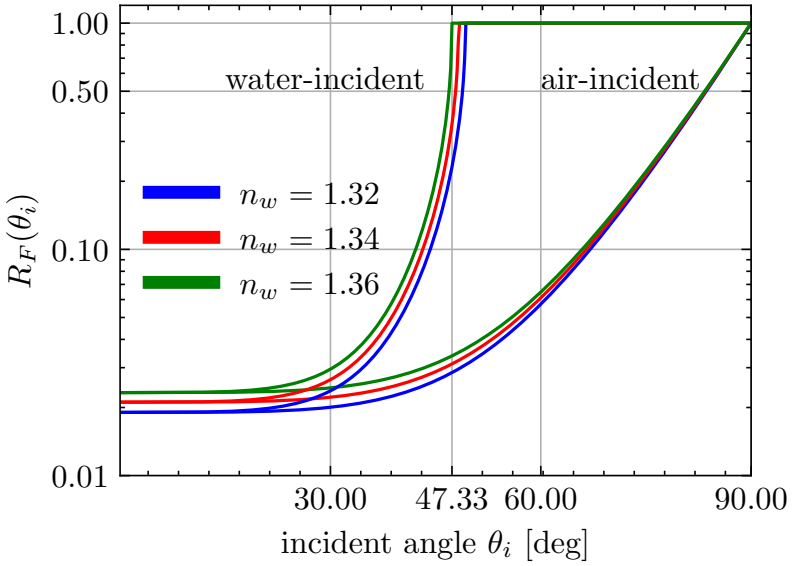
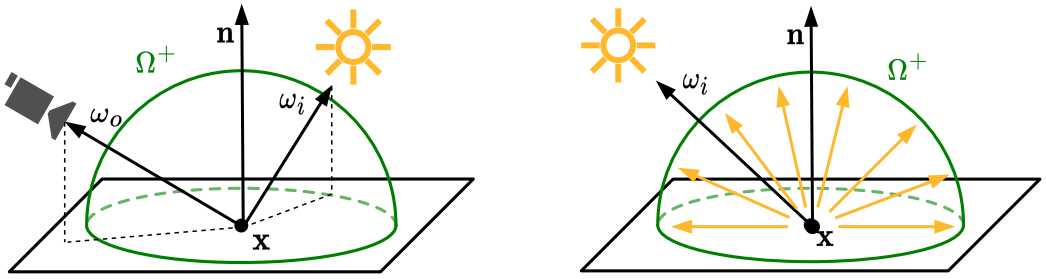


Figure 2.16: Comparison of water-incident (internal) and air-incident (external) reflectance curves at a air-water interface for various values of the index of refraction of water. The shown value $\theta_i = 47.33$ is the critical angle for $n_w = 1.36$.

2.3.3 The BRDF

Generally, surfaces (not just planar interfaces) scatter incident light into various directions and at various wavelengths. This section describes how surfaces reflect the incident light.

The *bidirectional reflectance distribution function* (BRDF) is a function describing reflection from a surface [74]. Figure 2.17a illustrates the BRDF: it gives the amount of radiance leaving the surface in the outgoing direction ω_o towards the camera, $L_o(\mathbf{x}, \omega_o)$, as a result of incident radiance along the direction ω_i, $L_i(\mathbf{x}, \omega_i)$.



(a) The BRDF is a function that describes how much incident light along ω_i is reflected from the surface in the direction ω_o.

(b) An example of a perfectly diffuse BRDF, so-called *Lambertian*, that reflects the incident light equally in all directions.

Figure 2.17: The bidirectional reflectance distribution function (BRDF) illustrated on the left image. The right image shows one of the simplest BRDF.

Formally, the BRDF is defined as the ratio of the reflected exitant radiance to incident irradiance. The differential irradiance at point $\mathbf{x}$ is the incident radiance on the surface over a differential cone of directions ω_i:

$$dE(\mathbf{x}, \omega_i) = L_i(\mathbf{x}, \omega_i)(\mathbf{n} \cdot \omega_i)\ \mathrm{d}\omega_i. \tag{2.33}$$

Then the BRDF of a surface for a pair of directions ω_i and ω_o is defined as a fraction of the incident irradiance [74]:

$$f_r(\mathbf{x}, \omega_o, \omega_i) = \frac{\mathrm{d}L_o(\mathbf{x}, \omega_o)}{\mathrm{d}E(\mathbf{x}, \omega_i)} = \frac{\mathrm{d}L_o(\mathbf{x}, \omega_o)}{L_i(\mathbf{x}, \omega_i)(\mathbf{n} \cdot \omega_i)\ \mathrm{d}\omega_i}. \tag{2.34}$$

It is the differential amount of radiance that is reflected in direction ω_o proportional to the differential irradiance at point $\mathbf{x}$.

To compute the radiance L_o leaving the surface at $\mathbf{x}$ into the direction ω_o due to the reflected radiance from the incident direction ω_i, the BRDF is integrated over the upper hemisphere of directions Ω^+ [7]:

$$L_o(\mathbf{x}, \omega_o) = \int_{\Omega^+} f_r(\mathbf{x}, \omega_o, \omega_i) L_i(\mathbf{x}, \omega_i)(\mathbf{n} \cdot \omega_i)\ \mathrm{d}\omega_i. \tag{2.35}$$

This is called the *reflectance equation*.

Physically based BRDFs have the following properties:

- *Reciprocity*: For all pairs of directions ω_i, ω_o: $f_r(\mathbf{x}, \omega_i, \omega_o) = f_r(\mathbf{x}, \omega_o, \omega_i)$, i.e., interchange of the directions with each other produces the same value.

- *Conservation of energy*: a surface cannot reflect more energy than the incident energy:

$$\int_{\Omega^+} f_r(\mathbf{x}, \omega_o, \omega') \cos\theta' \mathrm{d}\omega' \leq 1, \quad \forall \omega_o. \tag{2.36}$$

One of the simplest BRDFs is *Lambertian* [74]. It is used to model perfectly diffuse surfaces, i.e., surfaces that reflect incident illumination equally in all directions (as illustrated in Figure 2.17b). It is often used as a reasonable approximation to many real-world surfaces. The Lambertian BRDF has a constant value that when evaluated gives the following result:

$$f_r(\mathbf{x}, \omega_o, \omega_i) = \frac{\rho}{\pi}, \tag{2.37}$$

where ρ is the reflectivity of the surface and it varies from zero (completely absorbing, black surface) to one (completely reflecting, white).

The planar surface, shown in Figure 2.14a, from the previous section, is the opposite of diffuse surface — a perfectly *specular* surface, or mirror — where the incident direction $\mathbf{i}$ is always reflected about the surface normal $\mathbf{n}$.

Chapter 3

Optical Properties of Natural Waters

This chapter describes optical properties of real-world natural waters. Natural waters is collective term for earth's fresh and saline waters, such as ponds, lakes, rivers, but also seas, oceans. However, for the purpose of this thesis, the focus is mainly on seas and oceans (used interchangeably throughout this text), both of which have complex and highly variable optical properties, as shown in Figure 1.1, and both are extensively studied by the field of ocean optics. The terms described here provide baseline knowledge and guidelines for studying and designing bio-optical models of water in the next chapters. The information presented in this chapter is based on chapters from an online resource called *Ocean Optics Web Book* [66], a continually developing community resource for optical oceanography. Optical oceanography is a vast field spanning across multiple other fields, of which only a tiny fraction is mentioned in this text. However, the reader is free to consult the material if necessary.

Optical properties of natural waters are introduced in Section 3.1 and put into a larger context via the simplified "ocean-optics organization chart" in Figure 3.1. Furthermore, the optical properties of natural waters cannot be simply approximated based on the optical properties of pure water. Natural waters, both fresh and saline, contain various mixtures of dissolved and particulate matter. These solutes and particles themselves are optically significant and directly influence the optical properties of waters. Consequently, to be able to accurately reproduce the colors of natural waters, it is also necessary to consider the optical properties of such constituents. Section 3.2 describes optically important constituents in oceans.

Constituents in natural waters are highly variable in their composition and concentration, therefore show large temporal and spatial variations. Because of such great variability in the optical properties of natural waters, it is hard to find easily tabulated data or simple models. Optical oceanography utilizes coupling between constituent properties and optical properties and results from optical measurements to deduce information. These measurements cross to other areas, such as biological oceanography, marine photochemistry, underwater visibility, ocean color remote sensing, etc [59].

Then the following Sections 3.3, and 3.4 study how these constituents affect absorption and scattering in natural waters.

3.1 Optical Properties of Natural Waters

Optical oceanography divides optical properties of waters into two categories [59]: *Inherent optical properties* (IOPs) and *Apparent optical properties* (AOPs). The IOPs describe the properties of a medium, they do not depend on the ambient light field in the medium. They are the various measures of the absorption and scattering properties of a water body.

The AOPs are properties of a medium that depend both on the medium (the IOPs) and on directional structure of the ambient light field, and are regular and stable enough to be useful descriptors of the water body. Commonly used AOPs are various reflectances, average cosines, and diffuse attenuation coefficients.

The IOPs and AOPs are mutually exclusive descriptors of optical properties, i.e., an optical parameter cannot be both an IOP and an AOP. However, there are also other measures of the optical properties of a water body that do not fit into either of the categories. Radiometric variables, such as the various irradiances, are neither IOPs nor AOPs.

The radiative transfer equation (RTE) connects the optical properties of the water body and light within the water. Figure 3.1 shows how the RTE is used to compute radiometric variables based on the IOPs and boundary conditions. Then the radiometric variables can be used to compute the AOPs. The graph shows the most commonly used IOPs, AOPs, radiometric variables, and how they are related. Boundary conditions specify the radiance incident onto a water body and how it is transmitted through the water surface and reflected by the surface and the ocean bottom. Once the IOPs, along with the boundary conditions are given and the radiance distribution is known, then the RTE can be solved.

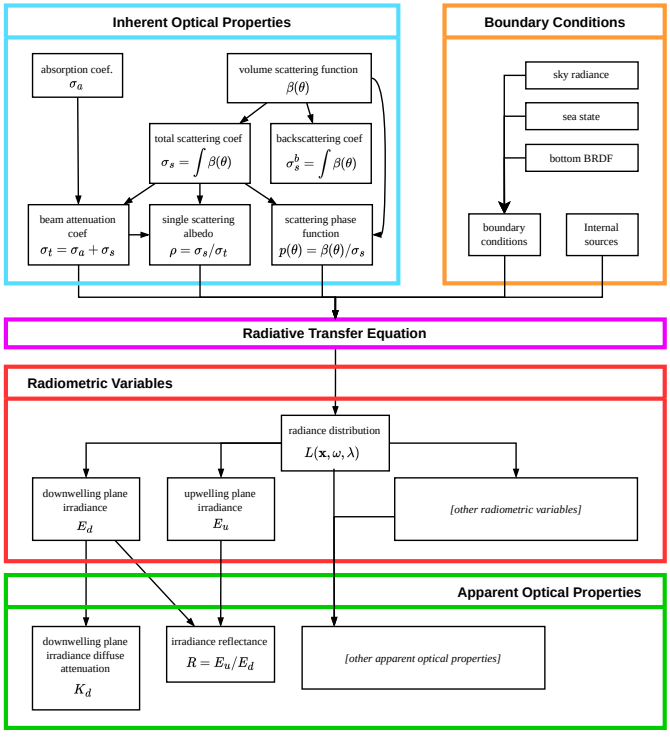


Figure 3.1: Overview of relevant ocean optical properties and their relationships. The inherent optical properties and the boundary conditions of the oceanic environment are fed into the radiative transfer equation to obtain radiometric variables, that are in turn used to obtain the apparent optical properties. (Adapted from the full diagram by Mobley [59]).

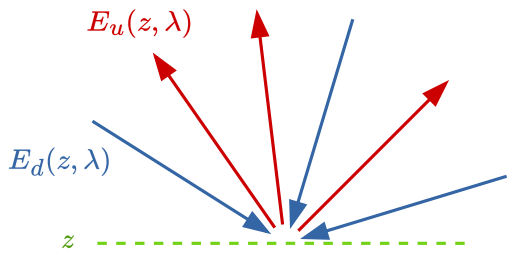
3.1.1 Apparent Optical Properties

The absorption and scattering properties of a medium such as sea water, are described by its IOPs. The fundamental IOPs that describe how water absorbs and scatters light are the *absorption coefficient* and the *volume scattering function*. Once these two IOPs are known, then it is also known how the medium interacts with unpolarized light. All other related coefficients can be derived from them [60].

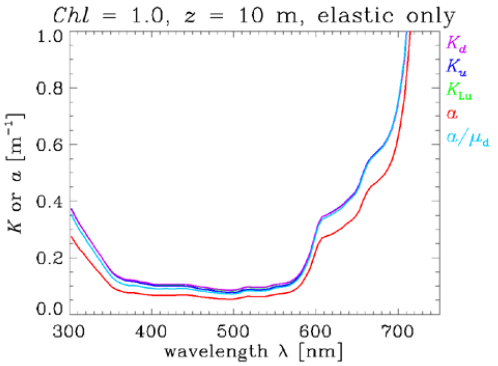
On the other hand, there are no fundamental AOPs. The most commonly used AOPs are those that are the least influenced by external environmental conditions (such as sun location or sky conditions) and those that provide the most useful information. The most commonly used AOPs are the various *reflectances* and *diffuse attenuation functions* (or "K functions") [60].

For example, the *irradiance reflectance* R can be related to absorption and backscatter coefficient and chlorophyll concentrations. Also called *irradiance ratio* is the ratio of upwelling (E_u) to downwelling (E_d) plane irradiances at certain depths (illustrated in Figure 3.2a): $R = E_u/E_d$. It is a measure of how much of the radiance traveling downwards is reflected upwards in any direction. Measurement data for this AOP are widely available, because both the E_u and E_d can be measured in a straightforward way [55].

K functions are often used as an approximation of the absorption coefficient. For example, the *diffuse attenuation coefficient for downwelling plane irradiance* K_d is used to approximate downwelling irradiance E_d at certain depths as it gets attenuated. In other words, it is a measure of the irradiance attenuated in the water volume per unit of depth. In these use cases, the K_d is typically assumed to be independent of depth, which can be a good approximation only for typical oceanic conditions in homogeneous water, and when far enough below the surface [61]. Figure 3.2b shows how K_d approximates absorption coefficient.



(a) Illustrates the irradiance reflectance R as a measure of upwelling irradiance to downwelling irradiance at depth z. The image shows how radiance along the light rays, which travel downwards and are reflected upwards, contributes to the irradiance reflectance.



(b) K functions at 10 m depth for waters with chlorophyll concentration and only elastic scattering. (*The graph is provided as is by Mobley [61].*)

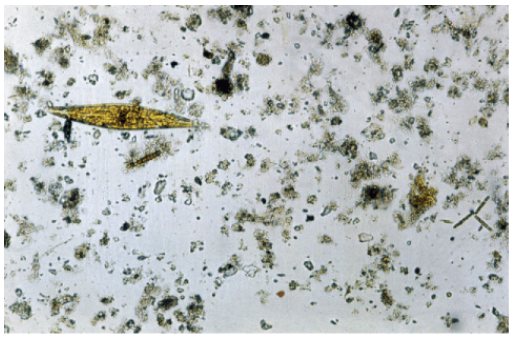
Figure 3.2: Left image illustrates the irradiance reflectance R, the right image illustrates the diffuse attenuation coefficient K_d.

3.2 Optically Important Constituents of the Ocean

Marine particles are extremely complex and vary in their composition, pigmentation, shape, internal structure, and packaging [16]. The IOPs in the ocean depend on physical characteristics of particulate and dissolved substances. Their composition determines absorption (based on the material), scattering (based on index of refraction), and different sizes and shapes scatter light differently [60]. To understand their interaction with light some important idealizations have to be made. It is impractical to study each individual particle (there are more than a billion bacteria in one ml of seawater), therefore particles are grouped based on similar properties.

In order to model the optical properties of particles, that is, to describe these properties using analytical or numerical means, empirical data is needed (size, shape, index of refraction, etc.). These can be either obtained from microscopy or need to be deduced from other measurements. The specific models for the different types of particles are usually based on observations, either in the lab or in the field, and on numerical calculations with inputs obtained from observations (e.g. using an optical model such as Mie theory to fit empirical data) [16].

The in-water constituents that affect aquatic optical properties are traditionally grouped into the following categories: sea water (water and inorganic dissolved materials), colored dissolved organic material (CDOM), phytoplankton, non-phytoplankton organic particles, inorganic particles, bubbles. The constituents are identified based on the way their optical properties are measured and are often grouped by similar optical properties [16].



(a) Particles in a drop of sea water through microscope. Largest is phytoplankton cell, rest is mineral material.

(b) A flask of clear water (left) and CDOM (right) from filtering coastal water.

Figure 3.3: Photos of ocean constituents from Mobley [17].

The most commonly modeled constituents are introduced in the list below:

- Particulate matter: collective term for the all organic and inorganic, animate and inanimate particles. Is classified into organic — all living or once-living particulate matter (such as phytoplankton and detritus) — and inorganic, all lithogenic and mineralic matter (such as sand and calcium) [16].
- Phytoplankton: microscopic, single-celled organisms (form chains or colonies), comprised of more than 10,000 species, that possess various pigments (such as chlorophyll-A) and photosynthesize. Absorb sun light, and causes scattering [20].

- Colored dissolved organic matter (CDOM): also referred to as “yellow matter”, dissolved organic materials (such as picoplankton, viruses, and colloids) that absorb and scatter light, hence the yellowish color [15].
- Non-algal particles (NAP): particulate matter that is not a phytoplankton pigment, includes all living and detrital organic matter such as the non-pigemnted portion of phytoplankton cells, detritus, viruses. May also include inorganic mineral particles (e.g., shells, clay, silt, and sand) and most recently also tiny particles of plastic [16].
- Detritus: non-living particulate matter, including dead bacteria, phytoplankton, fragments of cells, shells, marine now aggregates, etc. Both absorbs and scatters light [84].
- Minerals: also clay, silt, and sand, are fractions of typically lithogenic or mineralic particles of different sizes. Mineral particles enter ocean from river discharge, erosion of coastal cliffs, sediment resuspension, and deposition of atmospheric dust. Both absorb and scatter light [84].

The physical characteristics of the dissolved and particulate substances in the ocean vary by orders of magnitude, therefore the IOPs also vary. The absorption and scattering coefficients in turbid coastal waters, that contain high concentrations of phytoplankton, mineral particles, and dissolved organic matter, can be four orders of magnitude larger than those of pure water. The volume scattering function can also vary (at a given angle and wavelength) by orders of magnitude between open ocean and coastal waters. Optical oceanography tries to understand how the various constituents of the ocean determine IOPs [60].

3.3 Absorption in Natural Waters

Absorption is a physical process caused by the interaction of light with matter, in which case the light can disappear, with its radiant energy being converted into other forms, such as heat. Both absorption and scattering affect visibility in water in significant ways [83]. Figure 3.4 illustrates how absorption makes a distant image darker. Wavelength-dependent absorption also alters the color of a distant image, as shown in the right image of the figure.

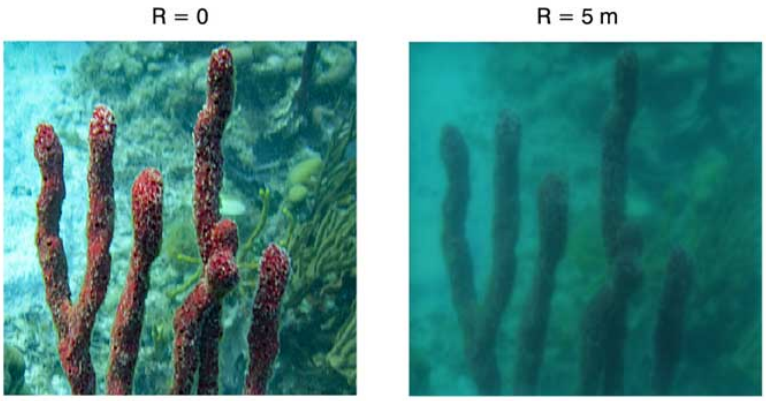


Figure 3.4: Simulated effect of absorption on a photo of a red sponge seen up close and from distance of 5 m in clear water. The image is darker and the red color has been removed, additionally scattering has blurred fine details (*Photos by Mobley [83].*)

Absorption is highly spectrally dependent compared to scattering. Concentration and composition of the particulate and dissolved constituents and water itself determine the magnitude and spectral features. Due to the IOPs being conservative, the magnitude of the absorption coefficient varies linearly with the concentration of the absorbing material. In theory, the absorption coefficient can be expressed as the sum of the absorption coefficient of each oceanic component. In reality, it is not possible to measure the absorption properties of each component individually, and thus the individual components are usually grouped into similarly absorbing constituents based upon similarity in their optical properties:

$$\sigma_a(\lambda) = \sigma_a^w(\lambda) + \sigma_a^{phyt}(\lambda) + \sigma_a^{NAP}(\lambda) + \sigma_a^{CDOM}(\lambda), \tag{3.1}$$

where the superscripts w, $phyt$, NAP, and $CDOM$ denote water, phytoplankton, non-algal particles, and colored dissolved organic matter, respectively [83].

For clear open ocean environments with very low concentrations of suspended and dissolved material, the absorption coefficient is dominated by water. Minimum in the blue region of water absorption spectrum therefore causes the blue color of the seawater. For other oceanic and/or coastal environments with high concentrations of suspended and dissolved material, the absorption coefficient is dominated by the material and, in general, the minimum absorption is in the green region, hence the green hue of such environments [83], [82].

3.4 Scattering in Natural Waters

Scattering denotes a physical process when light changes its direction on interaction with matter. There are two types of scattering in water: *elastic scattering* and *inelastic scattering*. The former describes only the change of direction, while the latter additionally considers a change of wavelength of the light. While absorption is generally more studied phenomenon than scattering, both are equally important to the prediction and understanding of light propagation. Scattering through small angles blurs the edges of an image, and can cause other effects on distant objects, such as twinkling or mirages.

In oceans, the field of ocean-color remote sensing measures light leaving the oceans due to backscattering (at angles greater than 90°) and multiple forward scattering (at angles of a few tens of degrees or greater). Scattering diverts the direction of light from a geometric straight-line, thus making the effect of absorption more pronounced. Any interaction of light with matter can cause scattering — a change in direction or wavelength. Light is scattered by particles of oceanic constituents of all kinds of sizes [62].

3.4.1 Scattering by Oceanic Constituents

Fundamentally, elastic scattering occurs when there is a change in the real part of the index of refraction (IOR), as described in Section 2.3.1. This happens when light travels from a region with one IOR into a region with a different IOR. The term *surface scattering* refers to scattering caused by the change in IOR at the boundary between two media, such as at the air-water interface. On the other hand, the term *volume scattering* refers to scattering caused by a change in IOR by the presence of discrete particles in the volume, by thermal fluctuations in density, or by turbulent mixing of fluids with different physical properties. Often, scattering by particles and by pure water is modeled.

Scattering by homogeneous spherical particles of some IOR within a different IOR is typically modeled using *Mie Theory* (introduced in Section 2.2.4). Mie Theory is also being applied to approximate scattering by non-spherical and/or non-homogeneous particles.

Such particles are common in nature: ice crystals in clouds, atmospheric dust, resuspended sediments in water [64].

Scattering by pure water is often modeled by *Einstein-Smoluchowski Theory*. Even the purest water has some scattering. Water exhibits fluctuations in density and in the IOR due to the random movement of water molecules. The Einstein and Smoluchowski formula is used as the volume scattering function. Elastic scattering by water has also similarities in angular shape and spectral behavior to *Rayleigh scattering* (introduced in Section 2.2.4). There is some effect of salinity on the scattering coefficient by water (30 % increase for range of salinities normally observed in the ocean), much less so of temperature (4 % between 0 and 26 $^\circ C$, and even less for pressure [13].

3.4.2 Index of Refraction of Water

Index of refraction (IOR), in its full form, defines how a medium affects light of a given wavelength. The complex part relates to the absorption of water and varies with wavelength. The real part of IOR of water also varies with wavelength in the visible spectrum. The IOR of water is influenced more or less by the following quantities: temperature, salinity and pressure. Higher temperature and salinity causes higher scattering. On the other hand, pressure has small effect on scattering, because water is very incompressible.

For near-surface waters, IOR is in range of 1.33 to 1.35. Including extreme depths, overall, IOR of water varies by less than 3 % over the entire range of these parameters [13].

3.4.3 Volume Scattering Function

This section is based on Chapters [62], [65] from Ocean Optics Web Book [66].

Volume scattering function (VSF) describes the angular distribution of unpolarized light scattered from its incident direction ω into direction ω' at a wavelength λ. For turbulent aquatic environments, scattering is almost always assumed to be *azimuthally symmetric* about the incident direction. In order words, the scattering is dependent only on the angle θ between the two directions. This holds true for spherical particles or randomly oriented non-spherical particles for unpolarized incident irradiance, which are present in such environments.

The VSF is denoted by $\beta(\theta, \lambda)$ and its units are $[\mathrm{m}^{-1}\ \mathrm{sr}^{-1}]$. In comparison to the *volume scattering phase function* (or simply phase function, as introduced in Section 2.2.4), $p(\theta, \lambda)$ $[sr^{-1}]$, the VSF is related to radiant intensity emanating at an angle θ from a volume, rather than from a single particle. It is the scattered intensity per unit incident irradiance per unit volume. Phase function and the VSF are related in the following way:

$$p(\theta, \lambda) \equiv \frac{\beta(\theta, \lambda)}{\sigma_s(\lambda)} \quad [sr^{-1}]. \tag{3.2}$$

This can be seen as partitioning the $\beta(\theta, \lambda)$ into the magnitude of the total scattering, expressed by $\sigma_s(\lambda)$ in $[\mathrm{m}^{-1}]$, and the angular distribution of the scattered light, which is $p(\theta, \lambda)$ in $[\mathrm{sr}^{-1}]$. In other words, a phase function can be thought of, as the corresponding VSF without the magnitude of the intensity of the scattered light.

Integrating $\beta(\theta, \lambda)$ over all directions (solid angles) gives the *scattering coefficient*, or the total scattered power per unit incident irradiance and unit volume of water:

$$\sigma_s(\lambda) = 2\pi \int_0^\pi \beta(\theta, \lambda) \sin\theta \ \mathrm{d}\theta. \tag{3.3}$$

The term 2π is based on an assumption, that the scattering process is *azimuthally symmetric* about the incident direction, that is, the scattering depends only on the *scattering angle* $0 \leq \theta \leq \pi$. This assumption holds if the medium is isotropic for unpolarized incident light. This is often taken as a reasonable assumption for natural waters in which the particles are randomly oriented by turbulence.

This integration can be divided into *forward scattering*, $0 \leq \theta < \pi/2$, and *backward scattering*, $\pi/2 \leq \theta \leq \pi$, part. Then the corresponding forward and backward scattering coefficients are, respectively:

$$\sigma_{fs}(\lambda) = 2\pi \int_0^{\pi/2} \beta(\theta, \lambda) \sin\theta \, \mathrm{d}\theta, \tag{3.4}$$

$$\sigma_{bs}(\lambda) = 2\pi \int_{\pi/2}^{\pi} \beta(\theta, \lambda) \sin\theta \, \mathrm{d}\theta. \tag{3.5}$$

Figure 3.5 illustrates the range of VSFs that can be found in various oceanic waters. VSFs are plotted as functions of the scattering angle, with the ordinate being in logarithmic scale to highlight the wide range of VSFs and to show the smallest scattering angles. Oceanic VSFs typically vary by 5 or 6 orders of magnitude over the range of very small to large scattering angles. The different VSFs can also vary by over two orders of magnitude at a given scattering angle. The scattering phase functions are plotted in the rightmost graph. The functions appear similar at low angles due to the logarithmic ordinate, but they vary by almost an order of magnitude at large scattering angles [56]. The asymmetry parameter g of the phase function is often used as a convenient measure of the shape of the phase function. Typical ocean waters have g values in range of 0.8 to 0.95. In other words, oceanic phase functions have very large values for small angles θ.

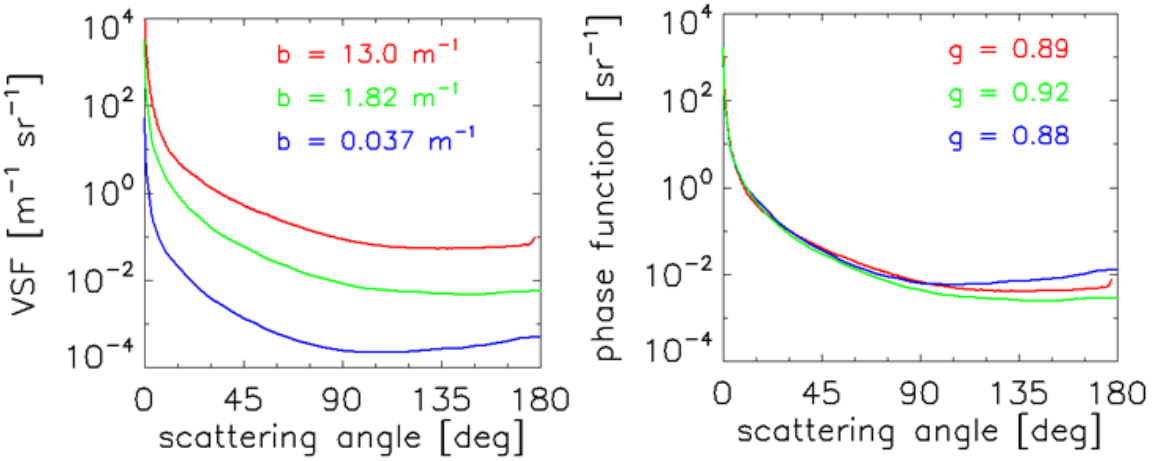


Figure 3.5: Example of volume scattering functions (on the left) and the corresponding scattering phase functions (on the right). Measured in clear, open ocean water (the blue curve), in a harbor (the green curve), in very productive coastal waters (the red curve). The corresponding scattering coefficients b and the asymmetry parameter g are also shown. (*The graphs are provided only in this form by Mobley [56].*).

3.5 Classification of Natural Waters

Optical oceanography classifies waters into water types based on their optical properties. For instance, waters can be classified based their colors, then the underlying optical property is primarily absorption (as scattering contributes to intensity but less to color). The most common classifications to water types are based on [14]:

- Visual comparison of the water color to the color of a standard color sample using human eye.
- Analysis of AOP spectra (such as Jerlov water types).
- The relative importance of chlorophyll in determining the optical properties of water (Case 1 vs. Case 2).
- Analysis of IOPs.

The most commonly used classifications in bio-optical models, which are used for rendering waters, are the *Jerlov classification* and *Case 1 versus Case 2 water types*. These two classifications are described in the sections below.

3.5.1 Jerlov Classification

The Jerlov classification divides water bodies based on the spectral diffuse attenuation coefficient K_d. The K_d and the absorption coefficient are related and are linked to the ocean color observed. Observations of the K_d were discretized into five typical open-ocean types, labelled, I, IA, IB, II, and III, and nine typical coastal types, labelled 1 to 9 [14].

Today, the Jerlov water types are commonly used as a convenient one-parameter classification scheme describing water clarity for essentially all natural bodies of water [88]. Solonenko and Mobley, 2015 [88] present a method to determine IOPs for all Jerlov water types. They provide tables of sets of absorption and scattering coefficients for the different water types.

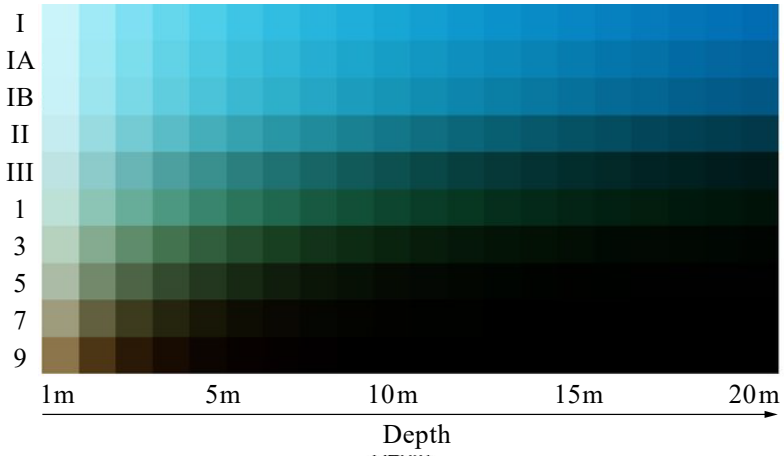


Figure 3.6: Illustrates the color of a completely white object seen in each of the 10 Jerlov water types at varying depths. (*Image adapted from Toledo et al. [91].*)

3.5.2 Case 1 versus Case 2 Water Types

The changes in spectral shape of the irradiance reflectance $R = E_u/E_d$ (described in Section 3.1.1) separates waters into two types or *cases*. Case 1 type are waters with concentration of phytoplankton that is high compared to other particles. In contrast, in case 2 the inorganic particles are dominant. In both cases, the dissolved yellow substance (CDOM) is present in variable amounts. An ideal case 1 is water that is pure culture of phytoplankton and an ideal case 2 is a suspension of nonliving material with a zero concentration of pigments.

Today, the definitions of the two types are used in relation to the influence of constituents on optical parameters:

- Case 1: waters whose optical properties are determined primarily by phytoplankton and other constituents — CDOM and detritus and degradation produces — then covary together with the phytoplankton concentration.
- Case 2: rest of the waters, that is, waters whose optical properties do not covary with the phytoplankton concentration, but instead are significantly influenced by other constituents such as mineral particles, CDOM, or microbubbles.

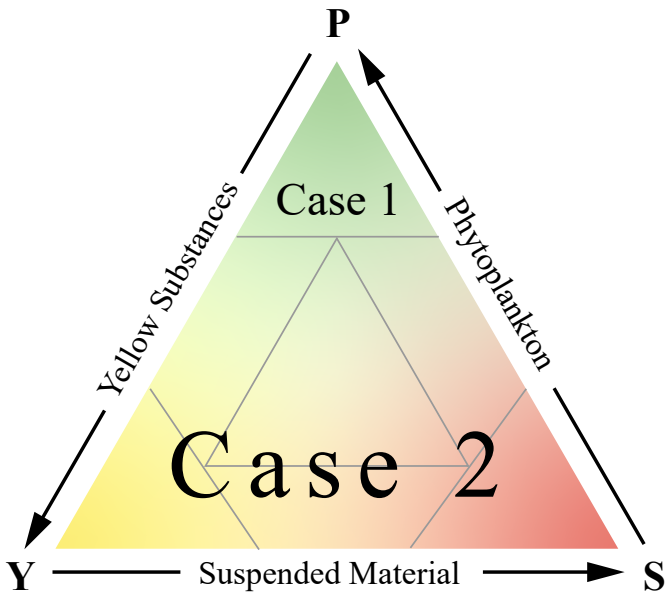


Figure 3.7: Illustration of Case 1 and Case 2 waters as top and bottom parts of a spectrum between varying concentrations — the representative colors — of the constituents. (*Image adapted from IOCCG [42].*)

In reality, natural waters are not simply either Case 1 or Case 2, but there is a continuous spectrum of water optical properties and their dependence on chlorophyll and other water constituents [14]. Figure 3.7 shows an illustration of such a spectrum using a gradient.

Chapter 4

Existing Approaches for Real-Time Ocean Simulation and Rendering

This chapter provides an overview of real-time ocean simulation and rendering techniques in computer graphics. These techniques are concerned with a range of topics: modeling, animation and simulation of ocean's surface or water body particles, rendering of the ocean surface with or without the light transport inside the water body, rendering of effects across the ocean surface, such as foam and spray, and lastly rendering of water volumes underwater with or without the underwater phenomena, such as caustics. The survey presented by Darles et al. [24] is the testament to the sheer volume of these techniques till the year 2011, since then the number of techniques has only increased.

The previous chapter showed the complexity and variability of inherent optical properties of oceans. Showing that the light transport in such media is dependent not just on the optical properties of the media itself, but also on the boundary conditions, that is, the conditions of the sky, the water surface, and the ocean bottom. Also, having in mind the aim of this thesis, the following types of techniques are the focus of this thesis:

- water surface simulation and animation for deep ocean waters, in Section 4.1,
- bio-optical models used for modeling properties of water volumes, in Section 4.2,
- participating media rendering of water volumes, either from above or below the surface, along with more general approaches that might be suitable, in Section 4.3,
- procedural generation topics, that are used for modeling or simulation of both water surfaces and surfaces of the ocean bottom, in Section 4.4,
- underwater effects, such as caustics, are the subject of the next chapter.

4.1 Ocean Surface Simulation Techniques for Deep Waters

Ocean surface simulation refers to methods that model and animate the detailed surfaces of water bodies, or general fluid simulations that can be applied to ocean simulation.

Ocean is complex due to its highly dynamic behavior, ranging from quiet sea, to agitated ocean with small turbulent waves to enormous shorebreaks. This motion is influenced by multiple phenomena occurring at small and large scales. Physically-based methods utilize information obtained by the field of *oceanography*. Oceanography defines the behavior of the

ocean surface (or ocean dynamics) depending on its location. It classifies the wave motion of the ocean surface into categories depending its location: deep ocean waters (far from the coast), intermediate areas or shallow water areas (close to the shore). Waves of deep waters exhibit characteristic oscillatory behavior, while waves of shallow waters interact with shallow ocean floor, causing them to break near the shore. Each category is modeled using a different set of techniques and approaches [24].

Darles et al. [24] provides a survey of ocean simulation methods. They divide the methods into two categories based on the models built to simulate certain types of ocean dynamics:

- **Models for deep water simulation**. These approaches use empirical laws from oceanographic research to build parametric, spectral or hybrid models. They, generally focus on simulation of deep water waves, without breaking waves.
- **Fluid simulation for shallow water**. These are physically-based methods, that rely on *Navier-Stokes Equations* (NSE) to simulate the dynamic behavior of ocean waves as they interact with the bottom. They are mainly used to simulate breaking waves near the shore.

This section further investigates methods for simulation of deep ocean waves of the free ocean surface. These methods build models either based on oceanographic observations, such as the parametric models, or based on (primarily) oceanographic measurements, such as spectral models. The goal of these methods is to compute the representation of the surface in the spatial domain, that is, to compute the concrete horizontal and vertical coordinates of each point on the surface. Methods based on parametric models, typically describe the surface directly in the spatial domain. While the methods based on spectral models, describe the surface in spectral domain, that is, the distribution of amplitudes and frequencies according to the modeled wave spectrum, and then they must convert this representation to the spatial domain [24].

4.1.1 Spatial Domain Approaches

These methods generate and animate the ocean surface directly in spatial domain. The surface is usually computed as sum of periodical functions that evolve temporarily using a phase difference.

Sinusoidal Wave Model serves as a basis for later techniques. It simplifies the complex motion of the ocean surface as a two-dimensional ideal surface represented by a sine wave with constant parameters. It also assumes uniform mean depth and small amplitudes in relation to the size of the water body. The wave travels in the horizontal direction and is vertically displaced for each horizontal coordinate x at time t giving the elevation as:

$$\eta(x,t) = A\cos(kx - \omega t), \tag{4.1}$$

where A is the amplitude, $k = \frac{2\pi}{\lambda}$ is the *wavenumber*, $\omega = 2\pi f$ is the *angular frequency* in [rad/s], $f = \frac{1}{T}$ is the *wave frequency* in [Hz], and T is the wave period in [sec]. Figure 4.1 shows such a sinusoidal wave. One may assume a fixed position on the wave, then the surface elevation for this point evolves through time. Or, one may assume a fixed point in time, then the surface elevation can be observed for all positions. The highest points of the wave height are called *crests*, and the lowest points are called *troughs*. This model thus provides a wave where all crests and troughs have the same elevation [30].

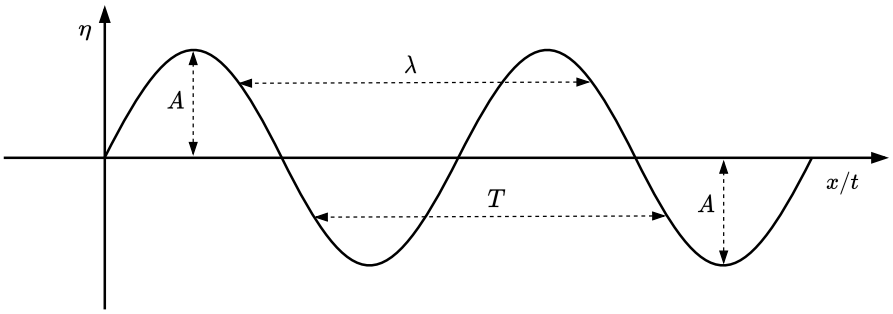


Figure 4.1: A sinusoidal wave model represented by surface elevation η at position x or at time t with amplitude A, wavelength λ, and wave period T.

This method can be extended into a sum of three-dimensional waves represented as a height map for each horizontal point (x, z):

$$h(x, z, t) = -y_0 + \sum_{i=1}^{N} A_i \cos(k_i^x x + k_i^z z - \omega_i t), \tag{4.2}$$

then the wavenumber becomes *wavevector*, $\mathbf{k} = (k_x, k_z)$, the direction of travel of the wave, the wavenumber becomes the magnitude of the wavevector: $k = ||\mathbf{k}||$, and y_0 is the height of the free surface [24].

Instead of sinusoids, Fournier [29] propose method called **Gerstner waves**. This approach can represent a variety of wave shapes, including waves with a sharp crest. The method allows for control of the waves to even simulate shape of breaking waves. To yield a more realistic result, a sum of Gerstner waves is commonly evaluated [24]. Figure 4.2 shows an ocean surface obtained using the method by Fournier et al. [29].

Figure 4.2: Ocean surface obtained using the Gerstner waves method by Fournier et al. [29].

4.1.2 Spectral Domain Approaches

Spectral approaches describe the ocean surface using *wave spectrum*, a distribution of wave energy among different wave frequencies or wavelengths obtained from theoretical or measured data. The spectral representation is then transformed into the spatial representation using *inverse fast Fourier transform* (IFFT).

The ocean surface is assumed to be a linear superposition of a large number of sinusoids with many different wavelengths, frequencies and phases, each traveling in different

directions. The wave spectrum defines the distribution of wave energy among the different wave frequencies or wavelengths.

Tessendorf [90] computes wave height field at horizontal point $\mathbf{x} = (x, z)$ as the sum of sinusoids with complex, time-dependent amplitudes:

$$h(\mathbf{x}, t) = \sum_{\mathbf{k}} \tilde{h}(\mathbf{k}, t) e^{i\mathbf{k}\cdot\mathbf{x}}, \tag{4.3}$$

where t is the time, $\mathbf{k}$ is the wavevector. The height amplitude Fourier components, $\tilde{h}(\mathbf{k}, t)$, determine the structure of the surface and are based on oceanographic data. Furthermore, the slopes of the wave height field can also be computed using IFFT, similarly to finite differences.

This approach produces waves on a patch with horizontal dimensions $L_x \times L_z$ of certain number of discrete sample points on a grid $N \times M$. This patch is perfectly periodic, so one such patch can be tiled seamlessly. There are certain restrictions on the dimensions to fully utilize the computational power of FFT: only square patches of sampled points can be produced: $N = M$, ranging from 16 to 2048, and only in powers of two [90]. Figure 4.3 shows an example of the computed wave height field.

There are various wave spectra to model deep sea waves at various states, e.g. turbulent sea, or less crested waves of the free ocean. For instance, Tessendorf [90] presents a method to generate a random ocean wave height field using Gaussian pseudo-random generator and a theoretic wave spectrum, called *Phillips spectrum*. This method is based on the observed Gaussian distribution of waves in deep waters and it was later used in famous computer-generated scenes for *Waterworld* and *Titanic productions* [24], as shown in Figure 4.3. Gamper [30] provides a comparison of various wave spectra.

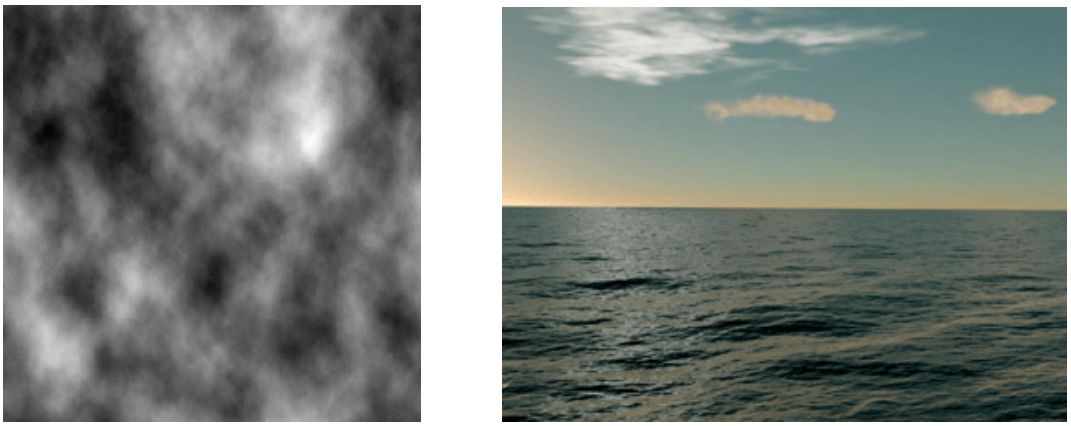


Figure 4.3: The right image shows the wave height field generated with Phillips spectrum. The left image shows a rendered ocean surface. (*Renderings by Tessendorf [90].*)

Typically, the waves produced by the FFT methods have rounded peaks and troughs, but in even in fairly good weather the waves are sharply peaked at their tops, and flattened at the bottoms. Tessendorf [90] also describes method called **Choppy waves** to create more realistic waves. This is achieved by additional displacement of the waves in horizontal dimensions. The displacement vector field is computed using the Fourier amplitudes of the height field. The method can cause some waves to pass through itself and invert due to large enough amplitudes. This can be tested for by computing the Jacobian of the transformation from the original horizontal point to the displaced one.

4.1.3 Limitations and Mitigation Strategies for Spectral Domain Approaches

Although the results obtained look realistic, there are limitations of these methods due to the fixed resolution of the height map: the waves look unrealistic on close-up, visible tiling or repetition artefacts are produces when look from a distance. This requires a new surface to be generated with higher resolution, which might not be feasible. Instead, implementations employ various LOD schemes and/or methods to remove the tiling artifact.

There are hybrid approaches that use combination of spatial and spectral definitions to gain advantages of both. For instance, using a combination of a spectrum and a sum of sinusoids to simulate breaking waves, and Tessendorf's approach [90] for deep waters.

Darles et al. [24] provides an overview of LOD approaches to Tessendorf's method [90]. Some approaches represent the ocean with multiple maps at various resolutions to account for low frequencies (i.e., the global motion of the waves) and high frequencies (the fine details). Other approaches use adaptive sampling of spectrum depending on the distance to the viewer.

Rydahl [85] presents an approach to remove tiling by transforming the world coordinates using fractal noise before sampling the displacement maps of the surface. Nvidia, 2011 [70] blends Perlin noise and FFT generated crests to gain detail at near sight (due to FFT) and remove tiling patterns at distance (due to Perlin noise). Dupuy and Bruneton [25] instead compute a number of patters, each sampling a different part of the wave spectrum to remove artifacts viewed at near and far distances. This way the ocean surface has an analytical expression.

4.2 Bio-Optical Models

This section provides an overview of various bio-optical models employed by different techniques for rendering light transport in water volumes, not just real-time techniques. The bio-optical model is the term for the underlying physical model used in these techniques. It is made up of the optical parameters of the ocean, thus being physically-based. These parameters are properties of participating media, e.g., absorption or scattering coefficients, of various oceanic constituents, as described in Chapter 3.

This section focuses solely on the way the absorption and scattering are modeled: which constituents are chosen, and how are their optical parameters modeled. Typically, these include the optical parameters shown in Table 4.1. Furthermore, only elastic scattering is considered.

Symbol	Description
σ_a	Absorption coefficient
σ_s	Scattering coefficient
σ_t	Extinction coefficient
$p(\theta)$	Phase function
σ_{bs}	Backscattering coefficient
K_d	Diffuse attenuation coefficient
C	Chlorophyll concentration

Table 4.1: The notation used for the various optical parameters.

Premože and Ashikhmin [78] use bio-optical model that supports many different water types, such as deep ocean water, muddy coastal water, and fresh water bodies. They assume homogeneous water bodies and define values of four parameters: $\sigma_s, \sigma_a, \sigma_{bs}$, and K_d (see Table 4.1 for the used notation). All the optical parameters are wavelength-dependent, with the exception of the scattering coefficient, which is derived from single-wavelength measurements for certain types of waters.

The absorption coefficient is approximated directly from the value of K_d. Any depth dependence on the parameter is ignored. Scattering coefficient is computed separately for two cases of waters, each with a different major cause of scattering: 1) Cloudy or muddy waters due to suspended particles (of silt, plant fibers, sawdust, chemicals, and microorganisms) from single-wavelength measurements, 2) waters with the abundance of phytoplankton and other particulate and dissolved material. The second case, takes into account the molecular scattering coefficient of water, and parameterizes the spectral scattering coefficient of the pigment (chlorophyll A and phytoplankton) by their concentrations. The renderings using this model are shown in Figure 4.4.

Figure 4.4: Renderings of different water types, from left to right: two variations of open ocean deep water, tropical water, and muddy coastal water. (*Renderings from Premože [78].)*

Cerezo and Serón [19] model waters with high phytoplankton concentration — case 1 waters — in which optical parameters can be correlated to chlorophyll concentration.

Absorption coefficient is the sum of the spectral absorption coefficient of pure water, and the chlorophyll-specific absorption coefficient parameterized by chlorophyll concentration C.

Scattering coefficient is modeled using single total scattering coefficient, which is only related to C. The phase function takes into account: 1) scattering by pure water, modeled by Rayleigh scattering, that accounts for the anisotropy of the water molecules, 2) scattering by particulate matter, modeled from measured data to account for high peaks in the forward direction of the phase function. Scattering coefficient of water is derived from single-wavelength measurements, and the scattering coefficient of particulate matter is taken simply as the difference between the total scattering coefficient and scattering coefficient of water. Figure 4.5 shows renderings of varying values of the optical parameters.

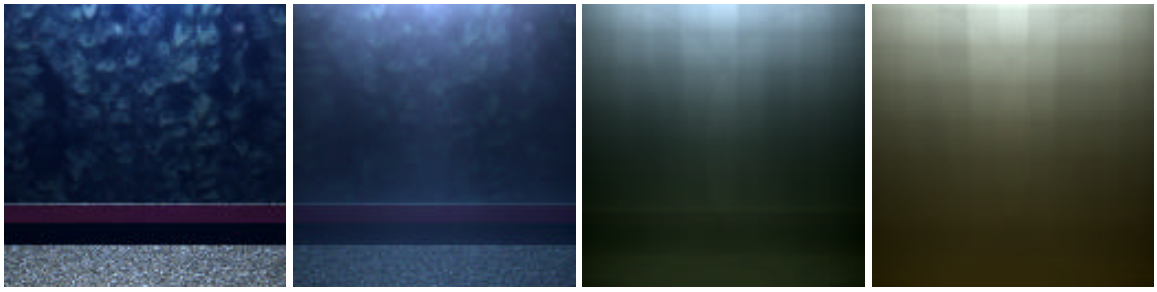


Figure 4.5: Renderings of seawater with different concentrations of phytoplankton, from left to right: pure seawater only, seawater with low concentration, seawater some concentration, seawater with high concentration. The selected values of parameters are also shown in the source. (*Renderings from Cerezo et al. [19].*).

Iwasaki et al. [43] and **Aranha** [10] model the phase function using Henyey-Greenstein phase function to account mainly for forward scattering in water. The scattering coefficient is then computed in a standard way by integrating the volume scattering phase function (in Equation 3.3).

Gutierrez et al. [36] (2005) simulates elastic and inelastic scattering in deep ocean waters by taking into account concentrations of phytoplankton (pigment chlorophyll) and CDOM (yellow matter). Absorption coefficient is modeled as the sum of absorption coefficients of the following constituents from spectral measurements: pure seawater, phytoplankton, and CDOM. Only the absorption coefficient of CDOM is computed from single-wavelength empirical absorption. Varying concentrations of chlorophyll and CDOM, both wavelength-independent, are taken account for both absorption and scattering. The used phase function accounts for: 1) scattering by pure seawater, modeled by Rayleigh scattering, in the same manner as Cerezo et al. [19], 2) scattering by suspended particles, modeled using Henyey-Greenstein phase function, with a high value of the mean cosine parameter g to account for high forward scattering.

Gutierrez et al. [37] (2008) builds on the work of Gutierrez et al. [36] (2005) and extends the bio-optical model to also account for complex inelastic scattering events that occur underwater, such as *fluorescence* and *Raman scattering*. Both absorption and scattering coefficients are sum of the respective coefficients of the following constituents: pure water, phytoplankton, CDOM (no elastic scattering), and detrius and minerals.

The spectral absorption coefficient of pure water is taken from measurements. Both spectral absorption coefficients of CDOM and of detritus and minerals are approximated by fitting an exponential curve to the corresponding single-wavelength measurements. Absorption by phytoplankton is approximated by modulating the spectral absorption coefficient of pigment chlorophyll-A with a varying chlorophyll concentration parameter.

Spectral scattering coefficients of: a) pure water is based on single-wavelength measurements of the scattering by pure water, b) phytoplankton is only related to its concentration in water, c) detritus and minerals are provided in form of measured data.

Spectral phase function is modeled as sum of the phase functions of each of the constituents: a) pure water, modeled by Einstein-Smoluchowski theory, b) phytoplankton, modeled by isotropic phase function, c) detritus and minerals, modeled based on the Mie theory with weighted two-terms Henyey Greenstein phase function to reproduce backscat-

tering more accurately. Figure 4.6 shows selected renderings with the colors being mainly caused by absorption and elastic scattering.

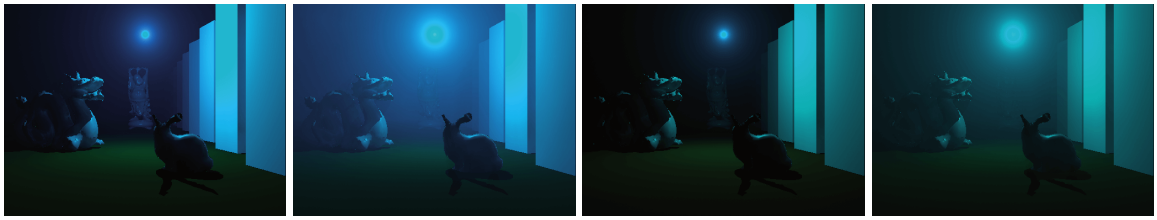


Figure 4.6: Renderings with varying concentrations of chlorophyll and the minerals and detritus turbidity, from left to right: no concentration, low concentration of chlorophyll, low turbidity and no concentration, low concentration of both. The source provides values of the simulated parameters. (*Renderings from Gutierrez et al. [37]*).

Sedlazeck and Koch [46] simulate deep sea underwater images based on physical models. The authors focus on deep sea scenarios, where no natural light exists, but the medium is illuminated in a few meters radius by point lights before the light gets attenuated. The model uses absorption coefficients from measurements and parameterized phase function. The scattering coefficient is computed by integrating the phase function. All the optic parameters are wavelength-dependent.

The absorption coefficient is computed as the sum of each considered group of matter: water molecules, dissolved salt, CDOM, phytoplankton, and anorganic particles caused by erosion and similar processes.

They model forward scattering and backward scattering separately. The forward scattering is approximated by a linear filter (in essence, a low pass filter). On the other hand, backward scattering involves an explicit use of phase function. The phase function used is a combination of the Einstein-Smoluchowski model and the *Kopelevich model*. The former is used to describe scattering by pure water (from single-wavelength measurements), and the latter is used to add scattering by large and small particles in the water (also from single-wavelength measurements of their respective phase functions and varying concentrations). Figure 4.7 shows renderings of varying turbidity.

Figure 4.7: Renderings with increasing turbidity from left to right, caused by the increased concentration of scattering particles. (*Renderings from Sedlazeck et al. [46].*)

Ma et al. [52] employ a bio-optical model to render ocean color in the *euphotic zone* of the ocean, the uppermost layer that receives the most sunlight. Their model includes constituents: pure water, phytoplankton, CDOM and inorganic suspended particles. They also consider non-uniform vertical distribution of all bio-optical properties based on typical chlorophyll concentration in such waters. The vertical distribution of chlorophyll concentration C is then approximated for each depth according to a Gaussian function for some initial concentration and depth range.

The spectral absorption coefficient is the sum of absorption coefficient of: a) pure water, taken from wavelength-dependent measurements, b) phytoplankton, computed from spectral chlorophyll-specific absorption coefficient of phytoplankton based on C, c) CDOM, which is derived from C by fitting an exponential curve to the measured input data, similarly to Gutierrez et al. [37].

The scattering coefficient is the sum of spectral scattering coefficients of pure water, which is obtained from measurements, and of particulate matter, which is computed using C. They model two phase functions to account for: a) scattering by pure seawater, which is modeled based on Rayleigh scattering, b) for scattering by particulate matter, which is derived from measured data. The final phase function is then the sum of these two phase functions weighted by the ratio of their corresponding scattering coefficients to the (total) scattering coefficient. Renderings with this model are shown in Figure 4.8.

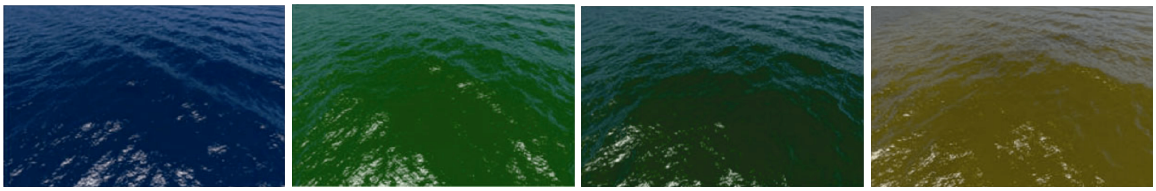


Figure 4.8: Renderings of ocean color with varying optical parameters, from left to right: pure seawater, seawater with high chlorophyll concentration, seawater with high CDOM concentration, and seawater with high concentration of suspended particulate matter. (*Renderings from Ma et al. [52].*)

4.3 Participating Media Rendering

This section describes existing approaches that compute light transport in participating media. First, the scope of the techniques is defined. Then, the related approaches are divided into the following groups:

1. Approaches that focus on light transport inside the water volume, computed either from above or below the water surface.
2. Approaches based on computer vision algorithm.
3. Approaches that approximate single scattering in any media using volumetric textures.
4. Approaches that approximate multiple scattering.

This section uses the notation presented in Table 4.1.

There is a vast amount of techniques concerning participating media rendering in computer graphics, even those techniques that are concerned with rendering of waters [24]. Light transport in volumes is governed by the radiative transfer equation (RTE, described in Section 2.2). As the light enters the water volume, it is scattered multiple times inside the water volume by particulates and molecules and quickly attenuated. Some of the scattered light is propagated back through the surface into the sky due to backscattering, contributing to the radiance field in the environment.

Tessendorf [90] in his notes discusses this radiosity problem and provides a schematic form to compute radiance at any point below the water surface:

$$L_{below} = R_T L_D + R_T L_I + L_{SS} + L_M, \tag{4.4}$$

where R_T is the *Fresnel transmissivity* of the incident light at each point and angle on the water surface, L_D and L_I is the direct light from the sun and the indirect light from the atmosphere that penetrates into the water, respectively, L_{SS} is the single-scattered light, from the sun and the atmosphere, that is scattered once in the volume before arriving at any point, and L_M is the multiply-scattered light, i.e., the single-scattered light that undergoes multiple scattering events in the volume. All the terms depend on each other in complex ways: single-scattered light depends on the direct and indirect light, the multiply-scattered light depends on the single scattered.

Darles et al. [24] divides ocean rendering techniques into two categories based on the order of approximation:

- First order approximations. These techniques apply geometrical optics laws to render light-water interactions. The water is assumed to be homogeneous and isotropic.
- Multiple order approximations. Techniques that simulate the phenomena that occur inside the water volume by simulating complex light-water interactions.

The multiple order approximations usually solve the RTE in some form. This field is full of offline approaches that are able to directly simulate global illumination phenomena, such as caustics. Such techniques are, e.g., photon mapping, or path tracing. These techniques are out of scope of this work, the focus is solely on volume scattering approaches for real-time.

4.3.1 First Order Approximations

The first order approximations, as introduced above, often target real-time. The common workflow of these techniques is following: Ray travels from a light source to an intersection point on a water surface, where it is reflected and refracted (as described in Section 2.3.1). The refracted ray then travels inside the water volume, either intersecting the bottom, or more commonly, an environment map is sampled using the direction of travel. The attenuation along the ray is computed using Beer's law. Finally the resulting radiance is weighted using Fresnel transmissivity. There might other simplifications or tweaks, such as parallel incident rays, or water surface is represented as a horizontal plane [24].

Tessendorf [90] provides a simplified approach to render ocean surface based on the optical parameters of deep ocean waters. Applies Beer's law to compute the radiance along a ray as it travels inside the volume accounting for absorption and out-scattering. The out-scattering part is viewed as an approximation of the intensity loss due to multiple scattering:

$$L_{uw} = \sigma_s I e^{-\sigma_t d} e^{-K_d z} = \sigma_s I e^{-\sigma_t d - K_d z}, \tag{4.5}$$

where I is the intensity of the ray, d is the distance traveled in the volume, and z is depth below the surface. The volume is treated as Lambertian reflector with a single value for its reflectivity.

Jensen and Goliáš [45] introduce real-time rendering approach of oceans. Specifically, they focus on rendering surfaces of deep ocean waters. After a ray of light hits the water

surface, a reflected and refracted directions are computed. The reflected direction is computed along the surface normal and the refracted direction is computed based on Snell's law, with a fixed index of refraction of water. To render the refraction effect on the water surface, a cube map of the underwater environment is used. For local reflections and refractions, i.e. when the ray should intersect objects underwater or above the water surface, a scene is rendered into a texture as if it were reflected or refracted from a flat plane placed below or above the surface. The reflected or refracted ray is then intersected with this plane. The intensity along the reflected and refracted ray is weighted based a precomputed Fresnel term.

Baboud and Décoret [11] use the lighting (and bio-optical) model given by Premože et al. [78] to render water volumes in real-time. They assume a water volume that is enclosed by terrain and water surface, both represented as heightfields, in a small aquarium-like bounding box (see Figure 4.10 for the final renderings). Simplified ray-tracing approach is then used to find, first, the intersection of view ray with the water surface, second, the intersection of the refracted view ray with the terrain (as illustrated in Figure 4.9). The water surface is intersected only once, multiple intersection, even if they happen (e.g., due to the presence of high waves), are ignored. The terrain is assumed to be diffuse, and its color underwater is computed according to *Phong shading model*. The color of the sky along the reflected ray is sampled from an environment map.

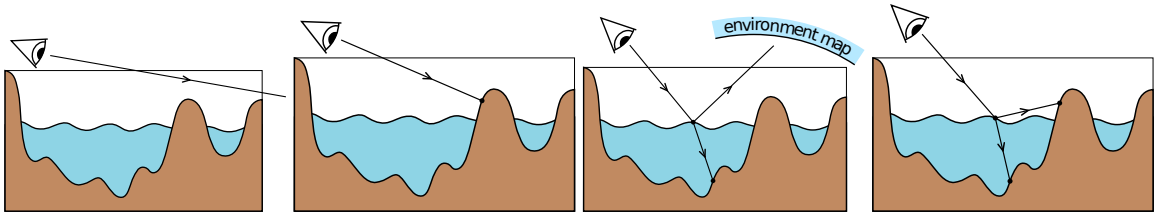


Figure 4.9: Four possible ray intersections with scene handled by Baboud et al. [11] (from left to right): hits the scene bounding box and samples the environment map representing the sky; the diffuse terrain is hit; the water surface is hit, while the reflected ray samples the environment map, the refracted ray intersects ocean floor; the reflected ray samples the terrain color. (*Images taken from Baboud et al. [11].*)

The used lighting model (Premože et al. [78]) is a crude simplification of the RTE, which introduces two separate underwater light fields: the *diffuse field radiance* L_{df}, to account for light scattered throughout the media, and the *directional radiance* — the computed radiance along a ray. The *irradiance ratio* $S = E_u/E_d$, a ratio of *upwelling irradiance* to *downwelling irradiance* (an IOP characteristic), is used, along with E_d, to compute the value of L_{df} just below the surface. The ratio S is approximated using optical parameters of the medium, and the downwelling irradiance just below the surface is approximated as a sum of sun and sky contributions. The uniform water body allows for analytical integration of the attenuation along a ray inside the water volume, while accounting for absorption with the increasing distance traveled and for out-scattering with the increasing depth.

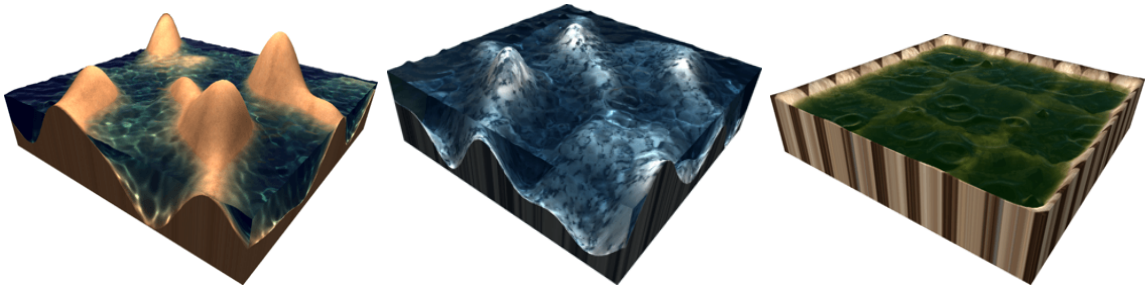


Figure 4.10: Renderings of water volumes as proposed by Baboud et al. [11], using the lighting model from Premože et al. [78].

Wenzel [93] in notes about real-time atmospheric effects elaborates on ocean shading in CryEngine [31]. In addition to the homogeneous volume, the water surface is represented as a horizontal water plane. The final color along a ray underwater is sum of the attenuated scene color and in-scattering contribution. They assume that the light entering the water always propagates straight down towards the seabed (see Figure 4.11 for illustration). Then the contribution due to in-scattering along a ray is analytically computed. The incident radiance from the sky is sampled from sky environment map.

Figure 4.11: From left to right: observer underwater with rays propagating straight down from a surface plane; underwater rendering; the same rendering with higher density. (*Images taken from Wenzel [93].*)

4.3.2 Computer Vision-based Approaches

Sedlazeck et al. [46] simulate both color and geometrical effects of deep sea underwater images using physical models. They focus on deep sea scenarios, where there is no natural light, but only point lights that illuminate only few meters before the light is attenuated. They account for light attenuation, scattering, and refraction.

They simulate underwater housing of a camera, with a flat glass port. This way the incident light is being refracted twice as it travels underwater: at the water-glass interface, and at the glass-air interface. They also account for camera transmittance and other effects such as vignetting.

To compute the color underwater they extend *Jaffe-McGlamery model*, a color model used in computer vision algorithms for the support of multiple light sources, rendering shadows, and for the use of a parameterized volume scattering function. Irradiance incident upon the pixel sensor of the camera, using the color model, is expressed as the sum of three irradiance components (shown in Figure 4.12):

$$E_{total} = E_{direct} + E_{forwardScatter} + E_{backScatter}. \tag{4.6}$$

Direct irradiance E_{direct}, accounts for point light sources immersed in the media illuminate nearby objects and for the reflected light from visible objects, that are not shadowed by any other objects. The incident ray enters the camera while being attenuated due to out-scattering and absorption. Light being scattered forward at small angles is approximated with a distance dependent Gaussian filter.

The Forward scattered irradiance is again approximated using the distance dependent Gaussian filter of E_{direct}.

The backscatter (or *veilling light*) irradiance approximates the light being scattered in all directions multiple times, causing ambient or veiling light close to a light source. This is approximated by slicing the volume into a number of planes of certain thickness that are parallel to the camera's near plane. Each such plane, has the irradiance incident upon it computed similar to $E_{direction}$, now for each position on a plane with the volume scattering function applied, and forward scattered part is also taken into account. Then the contribution due to backscatter is computed as the superposition of all the slices.

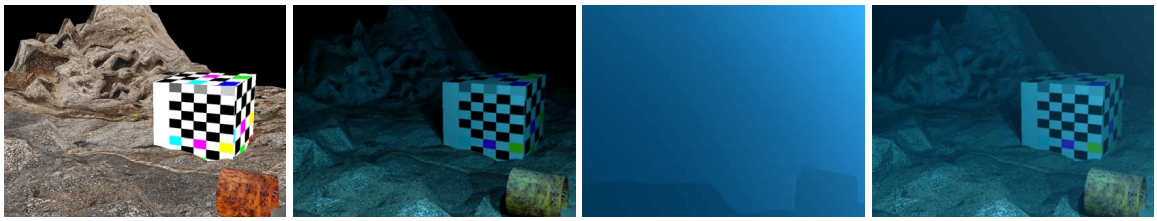


Figure 4.12: Renderings, form left to right: scene, direct signal, backscatter contribution, combined result. (*Renderings from Sedlazeck et al. [46].*)

4.3.3 Single Scattering Approximations

The following approaches render participating media using voxelized volumes. Though these approaches are used to compute atmospheric scattering (as shown in Figure 4.13), they provide a general outline of how to compute single scattering in any media.

Figure 4.13: Rendered scene with atmospheric scattering using the participating media rendering approach by Wronski [94].

Wronski [94] presents unified compute shader-based approach for rendering atmospheric scattering as participating media, called *volumetric fog*. Media have varying density and shadowing term is added, which causes light shafts or godrays. Ray marching is used to

render voxelized participating media into a volumetric texture that is positioned in the view frustum, so-called *frustum aligned 3D texture*.

The approach allows for parallelization of ray marching steps by storing intermediate results into volumetric textures between multiple passes:

1. Density estimation: generates density (attenuation) and scattering coefficient, stores the former in RGB channels and the latter in alpha channel.

2. Lighting calculation: stores the constant ambient terms from the main light and from multiple point lights that intersect the view frustum. Both are modulated by phase function and stored in RGB.

3. Ray marching: solves the scattering by linearly marching through the intermediate 3D textures. Applies Beer-Lambert law to compute transmittance which is applied on the in-scattered light and accumulated. Transmittance is stored as a single value (wavelength-independent) in the alpha channel.

The computed lighting can be applied in forward or deferred shading pipeline by a simple 3D texture bilinear look-up. The 3D texture is mapped to camera frustum in NDC space with exponential distribution of depth slices. The volume used is 160×90 with 64 or 128 depth slices, this can produce the effect in distances in world space between 5 and 128 meters.

This approach also has limitations and produces artefacts. It lacks high frequency geometry details and overall produces a rather soft effect, which is suitable for fog. The use of a discretized volume inherently introduces temporal aliasing and flickering. The author tries to mitigate this by storing only low frequency information for every depth along the ray, and uses uses quadrilinear filtering on the volumetric data to minimize edge artifacts on depth discontinuities.

To mitigate the temporal aliasing or flickering problem of high frequency source information, low-pass filter is used. Also lower resolution and smoother shadow maps are preferred, which shadow leaking being acceptable in participating media.

To mitigate the aliasing even further, especially with animated participating media, various temporal anti-aliasing techniques (e.g., termporal super-sampling) can be used. Instead of regular grid sampling, which produces aliasing, jittered grid sampling can be used. It reduces aliasing but introduces higher frequency noise. This higher frequency signal can then be filtered and blurred using lower resolution kernels (in both temporal and spatial domains).

Hillaire, 2015 [39] presents physically based and unified approach to volumetric rendering. Focuses on atmospheric scattering and fog with varying density and scattering occlusion to simulate light shafts.

Uses frustum aligned 3D textures in clip space, similarly to Wronski [94], with the same resolution as screen and with 64 depth slices. Assumes a tiled-based deferred pipeline which uses 16x16 tiles with culled light lists. The approach brings the following improvements to the each corresponding step of Wronski's algorithm:

1. Material properties: participating media properties are stored into so-called *volume buffer* or V-Buffer, which allows for heterogeneous media: stores wavelength-dependent scattering coefficient and single value for extinction (wavelength-independent). Also stores g parameter for phase functions.

2. Froxel light scattering: into a single 3D texture per voxel evaluates in-scattered light and extinction, considers indirect light, sun light and local lights, Takes into account shadows, and volumetric shadows for local lights. Introduces *temporal volumetric integration* to mitigate aliasing and flickering. This is accomplished by jittering of samples by the same offset for both the material and scattering samples along the view ray. Then, the current frame is blended with the previous frames, based on exponential moving average.

3. Final integration: for each voxel along view ray, solves the single scattering by ray marching and integrating. Introduces energy conservative integration, by analytically integrating the scattered light as a function of transmittance, to mitigate light leaking due to highly scattering participating media thus improving volumetric shadows.

Animated participating media (e.g., fluid simulation) or moving lights can cause artifacts by leaving trails behind.

4.3.4 Multiple Scattering Approximations

Computation of multiple scattering is beyond what is feasible in real-time. Most techniques that focus on real-time thus must assume crude simplifications. Only few relevant and simplest are investigated.

Hillaire, 2016 [40] approximates multiple scattering in clouds by summing multiple octaves of single scattering, while increasing the properties of participating media with each octave. It can be easily incorporated to ray marching, but it does not represent well complex behavior caused by multiple scattering: side or backward scattering, cone spread, etc.

Monzon et al. [67] approximate multiple scattering in underwater scenarios based on the commonly used AOP (apparent optical property) — *diffuse attenuation coefficient for downwelling irradiance* K_d (described in more detail in Section 3.1.1 about IOPs and AOPs). They build on the assumption that multiple scattering underwater is of very low frequency and can therefore be modeled as a function of depth and wavelength. They further assume non-emissive isotropic medium. This simplification, along with the assumption of constant incident irradiance (radiance from every direction) at every distance from water surface, allows for analytical evaluation of the volume rendering equation within the medium. Figure 4.14 shows renderings for different values of optical parameters in different waters based on the Jerlov classification (described in Section 3.5.1).

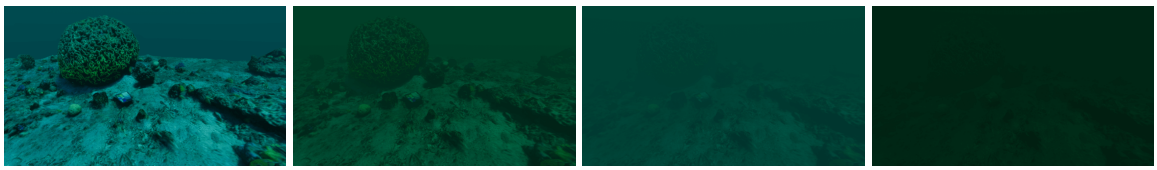


Figure 4.14: Rendered scene with different set of spectral optical parameters for different Jerlov water types. (*Images from Monzon et al. [67].*)

4.4 Procedural Noise Functions and Implicit Surfaces

Procedural generation is an alternative way of generating data or simply assets to the typical human or manual generation of assets. The term *procedural* refers to the description of entities by algorithms and program code rather than by data structures. Its goal is to approximate the seemingly random and unpredictable variety observed in real world by automatically generating various kinds of assets based on parameters and through the use of randomness or pseudo-randomness. There is a wide range of techniques used for generation of textures, models, and animation [26].

The focus of this section is to provide an overview of noise functions, their combination, and can they be used to generate or model terrains represented as implicit surfaces. The last part provides a brief discussion about visualization of such representations. Noise functions can be used to generate, animate or aid in modeling of water surfaces, or generally any surface or terrain, such as the ocean floor [26].

4.4.1 Noise Functions

The building block of procedural generation is noise, which is usually computed by interpolating between pseudo-random values. There are certain requirements or properties for a function to be noise that is useful for generation, it should not be an incoherent, completely random noise, such as white noise, because true randomness is an unusual and undesirable phenomenon. The properties of commonly used noise functions usually are [26]:

- it produces "apparently" random, but not entirely random values like white noise,
- it is repeatable, i.e., it gives the same value for the same inputs (e.g., texture coordinates),
- it is smooth enough for it to be stable and not show aliasing artifacts,
- its values are in range from -1 to 1,
- it does not exhibit obvious periodicities or regular patterns.

Commonly used noise functions range from one-dimensional to three-dimensional. There are many categories of noise functions based on the way they are computed. One such category is *Lattice noises*. These functions use pseudo-random numbers that are distributed uniformly in a lattice (a line for 1D, a grid for 2D, a 3D grid). *Value noise* is a type of noise that interpolates between the corner values of each cell to obtain noise. This is done in 1D, by interpolation between two points on a line, or in 2D, by interpolation between the four corners of a cell of the grid, and also in 3D, by interpolation between eight corners of a cube [92].

Interpolation functions are usually polynomials of a certain degree. The higher the degree, the smoother values are obtained, but also the computational cost rises. Most commonly used functions are: linear, and cubic [92].

Because value noise tends to look "blocky", another type, called *Gradient noise*, is oftentimes used instead. Typical example of gradient noise is *Perlin noise* [72]. In this case, noise is determined at a point by computing a pseudo-random gradient at each of the eight nearest vertices on the integer cubic lattice and then smoothly interpolating between them. Perlin noise is used widely in graphics because of its simplicity and performance cost.

A technique called *fractal Brownian Motion* (fBM), or simply "fractal noise" is used to produce noise that contains many incrementally finer details, as observed in nature. This is achieved by adding different iterations of noise, called *octaves*, with successively higher frequencies (*lacunarity*) and smaller amplitudes (*gain*). The simplest variant of fBM of a dimensional noise is:

$$fBM(x) = \sum_{i}^{octaves} A_i \cdot noise(x \cdot f_i), \tag{4.7}$$

where $A_i = A_{i-1} \cdot gain$ is the amplitude and $f_i = f_{i-1} \cdot lacunarity$ is the frequency for i-th octave, respectively [92].

4.4.2 Implicit Surfaces

Generally, points on a surface are described explicitly as a list of concrete parameters, such as x, y, and z. *Implicit surfaces* provide another way of representing surfaces. Instead of describing a point on the surface using concrete parameters, an implicit form is used instead:

$$f(x, y, z) = f(\mathbf{p}) = 0. \tag{4.8}$$

A point lies on the implicit surface if the function evaluates to zero. In other cases, it gives the signed distance to the implicit surface: negative distance means that $\mathbf{p}$ is inside the surface, and positive outside. Therefore, these functions are sometimes also called *signed distance functions* (SDFs).

Surfaces in their implicit form are often used for intersection testing with rays, as it is often simpler to find the intersection in this form than in the the corresponding (if any) explicit form. Implicit surfaces can be stored in a compact way and evaluated on the fly without increasing memory footprint. Objects described in this form can be easily combined in various ways, such as blending, subtracting, adding, etc [8].

Quilez [79] [80] provides an extensive list of both 2D and 3D SDFs for all kinds of primitives. Some simple examples of implicit surfaces are:

$$f_{sphere}(\mathbf{p}, r) = ||\mathbf{p}|| - r, \tag{4.9}$$

$$f_{plane}(\mathbf{p}) = p_y, \tag{4.10}$$

where f_{sphere} is an implicit function for a sphere at origin with radius r, and f_{plane} returns the vertical distance (on y-axis) to a horizontal plane.

The normal of an implicit surface is described by the partial derivatives of the function:

$$\mathbf{n} = \nabla f(x, y, z) = \left(\frac{\mathrm{d}f}{\mathrm{d}x}, \frac{\mathrm{d}f}{\mathrm{d}y}, \frac{\mathrm{d}f}{\mathrm{d}z} \right), \tag{4.11}$$

where ∇f denotes the gradient of the function. To compute the normal using this equation, i.e., analytically, the function must be differentiable, and thus also continuous. More commonly, a numerical method called *central differences* is used instead. To approximate the normal, the implicit function is sampled in each direction in small steps:

$$n_x = \nabla f_x \approx f(\mathbf{p} + \epsilon \mathbf{e}_x) - f(\mathbf{p} - \epsilon \mathbf{e}_x), \tag{4.12}$$

$$n_y = \nabla f_y \approx f(\mathbf{p} + \epsilon \mathbf{e}_y) - f(\mathbf{p} - \epsilon \mathbf{e}_y), \tag{4.13}$$

$$n_z = \nabla f_z \approx f(\mathbf{p} + \epsilon \mathbf{e}_z) - f(\mathbf{p} - \epsilon \mathbf{e}_z), \tag{4.14}$$

where $\mathbf{e}_x = (1, 0, 0)$, $\mathbf{e}_y = (0, 1, 0)$, $\mathbf{e}_z = (0, 0, 1)$, and ϵ is a small number representing the step size [8].

Primitives can be easily combined together to create whole scenes. Quilez [80] defines all kinds of operators for combining two primitives together into one primitive. For instance, union, subtraction, intersection and their smooth variants, as shown in Figure 4.15. Primitives can also be repeated over a region of space using so-called *Domain repetition operator* that uses modulo operator. Furthermore, primitives can be easily distorted, deformed or blended together.

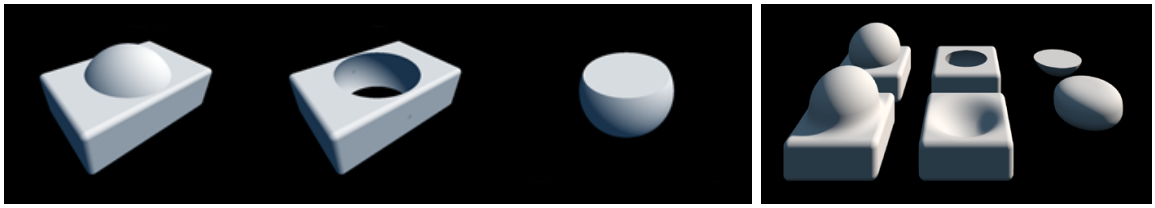


Figure 4.15: Images illustrate how primitives represented as SDFs can be combined together using various operators. The left image shows two primitives (box and sphere) and the results of operators (from left to right): union, subtraction, intersection. The right image shows the same primitives repeated with the smooth variants of the same operators applied. (*Images from Quilez [80].*)

To visualize a set of implicit functions, usually one of the two methods are employed: *ray marching* or *polygonalization*. Polygonalization converts implicit surfaces into polygon meshes using algorithms, such as *marching cubes* or *marching tetrahedra*. These meshes can then be rasterized in a traditional way [8]. Ray marching, on other hand, directly samples the implicit functions along rays, computing the final color based on "what is seen". Ray marching is described in more detail in the section below.

4.4.3 Ray Marching

Ray marching is used to find an intersection of ray with surface. It provides an alternative to analytic ray-intersection methods and rasterization with a depth buffer. In comparison to ray-intersection methods, which can be used only for primitives with analytic solutions to ray-intersection, ray marching can be thought of as an algorithm for numerical estimation of arbitrary ray-surface intersections.

Generally, ray marching works by "marching" from the ray origin $\mathbf{o}$ along the ray by sampling points $\mathbf{p} = \mathbf{o} + t\omega$ using the ray parameter t as a distance from the origin. Ray marching terminates when the sampled point intersects or crosses a surface. It is usually employed to find intersections with implicit surfaces. In this case, the marching terminates when the implicit function evaluates to zero or less.

Marching can be done with fixed steps from the ray origin, which might be too slow or miss an intersection all together, or it can use a signed distance estimator to find a safe step size it can march to avoid overstepping the intersection. The safe step size is usually determined by finding a sphere originating at point $\mathbf{p}$ that touches (i.e., is tangent to) at most one primitive. The safe step size is then the radius of the sphere, since the closest intersection is found at the surface of the sphere. This approach is called *sphere-tracing* [53]. Figure 4.16 illustrates how an intersection is found in a scene with implicit surfaces using this approach.

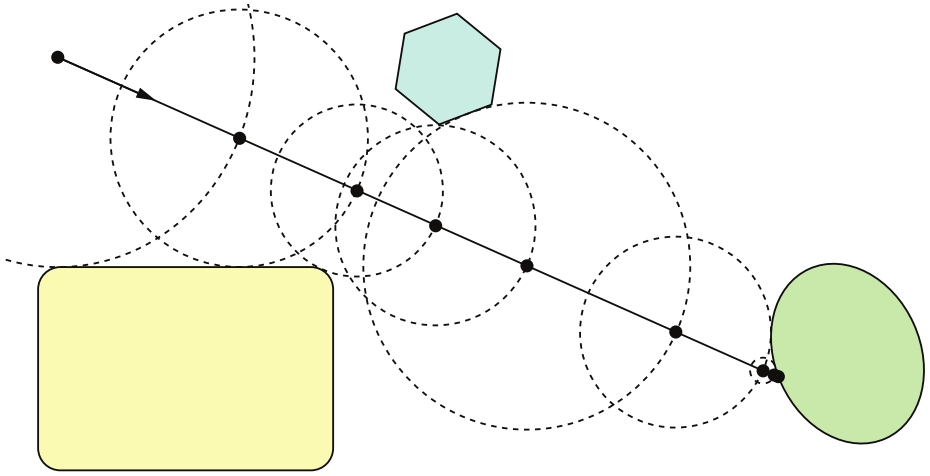


Figure 4.16: Finding an intersection with signed distance fields along a ray using sphere tracing. The dashed circles indicate a distance to the closest surface from their centers that is that is safe to advance along a ray. (*Image from Real-time Rendering, 4th edition [8].*)

Chapter 5

Real-Time Caustics Approaches

The interaction of light with curved reflective or refractive surfaces causes the light to defocus in some areas and focus in others. While the defocused areas create shadows, the focused or more dense areas of light cause a visual phenomenon called *caustics*. For instance, the reflection of light by a coffee mug causes caustics on the inside and on the outside. Or in a more relevant case, light incident onto a curved water surface is refracted and attenuated as it travels through the water volume finally converging on opaque surfaces causing caustics.

This is shown in Figure 5.1. In computer graphics, real-time especially, this phenomena is divided into two separate effects: *surface caustics* (image on the left) and *volumetric caustics* (image on the right). Furthermore caustics that are caused by reflection are called *reflected* or *reflective* and caustics caused by refraction are called *refracted* or *refractive*.

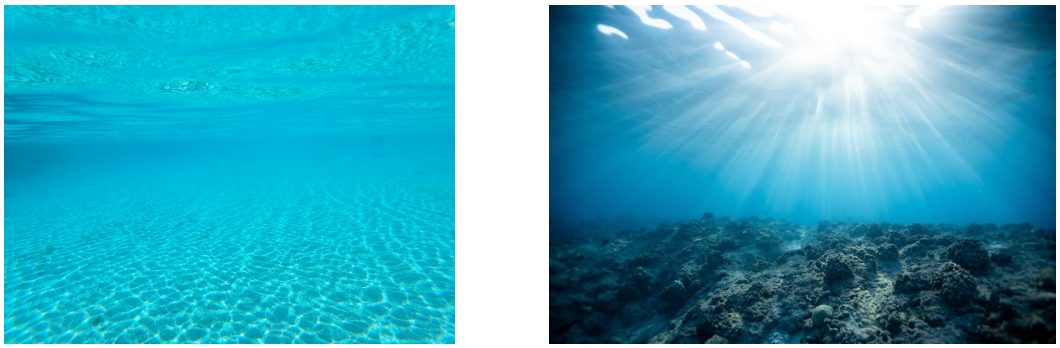


Figure 5.1: The left image shows surface caustics formed on the shallow floor (by MIRTO KON, pexels.com). The right image shows rays of light underwater, called volumetric caustics (by Jeremy Bishop, pexels.com).

Rendering caustics, especially in participating media, is a complex task that requires global information, and is therefore generally constrained to offline global illumination techniques, i.e., techniques that solve the volume rendering equation in some form (described in Section 2.2). Exact solutions are feasible only in the simplest environments. Renderers, generally, approximate this using numerical methods, such as bidirectional path tracing [74] or photon mapping [44].

Real-time rendering methods either fake the caustics altogether, or they try to produce physically-based caustics that correspond to the conditions of the scene (i.e., state of

lighting, water surface, etc.). Techniques of the latter type are commonly based on photon mapping. The common workflow of photon mapping is following [48]:

1. *photon emission*: photons are emitted from light sources into the scene,
2. *photon tracing*: photons are traced in the scene and scattered by reflective or refractive surfaces (also called reflectors and refractors), until they hit surfaces that are opaque enough, so-called caustics receivers.
3. *photon gathering*: millions of photons are gathered as they converge at certain places to obtain smooth caustics in *photon map*.

The use of photon mapping is not feasible for real-time because it implies heavy use of ray tracing and it requires a large number of photons (even tens of million) to obtain smooth images, otherwise the photon map is very noisy. Therefore, the photon mapping-based methods address these issues in various ways:

- intersection testing: from crude simplifications of surfaces by simple shapes (e.g., ocean bottom represented by a plane), to image-space tracing (e.g., ray marching depth buffers), to recent utilization of specialized ray tracing pipelines,
- photon gathering: additive blending, usually accompanied by filtering, splatting photons of various sizes (not just point primitives), beam tracing (i.e., tracing triangular beams instead of points), and recently denoising (i.e., blurring).

Section 5.1 provides an overview of certain techniques used for rendering surface caustics, while Section 5.2 describes certain techniques that render volumetric caustics in media.

5.1 Surface Caustics

The techniques presented are divided into the following categories:

- Fake caustics: approaches that produce physically incorrect, yet convincingly looking caustics,
- Image-space caustics: techniques that trace rays in image space or use other simplifications to estimate intersections with scene geometry, e.g., by marching through depth buffer from the light's point of view,
- Ray-traced caustics: recent techniques that intersect rays with the dynamic scene in world space using DirectX Raytracing (DXR) pipeline, that is supported on certain hardware.

5.1.1 Fake Caustics

These approaches are usually early attempts at real-time caustics simulation. Caustics are simulated by faking or approximating the observed look. These techniques are purely aesthetics-driven and do not consider realism, i.e., the caustics do not correspond to the real light transport in a given water volume. Caustic patterns, possible generated offline or created by artists, are stored in textures, which are then projected on surfaces and applied as light maps. These caustics textures might be even animated by using a set of such textures. This approach has been used in games before [9].

Stam [89] computes a caustic-like patterns using wave theory (see Figure 5.2). The resulting patterns are periodic both in time and space and are generated for a ground plane at various depths. Such patterns can be used in form of a animated texture that is then additively blended with an object's base textures.

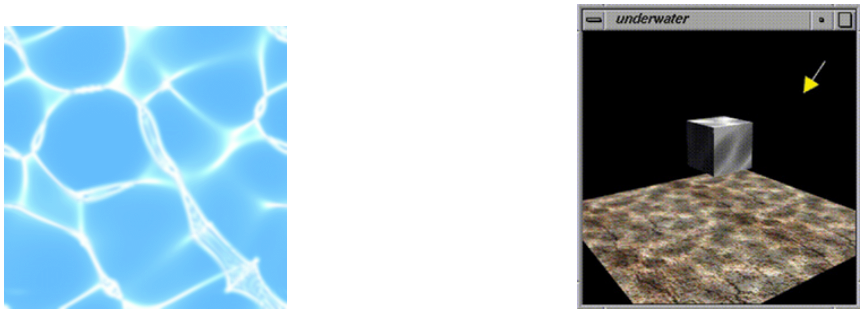


Figure 5.2: Caustics patterns created using Stam's approach [89] and stored in a texture (on the left, *image from Stam [89]*), which is then applied onto objects as a light map (on the right, *image from Guardado et al. [35]*).

Guardado et al. [35] present purely aesthetics-drive approach for the computation of underwater caustics. They consider only a subset of arriving rays by making the following assumptions: the sun is directly above, rays hit the ocean floor from directly above, light attenuation is fixed per unit distance, and the ocean floor is represented by a plane. The approach is based on simplification of the backward ray tracing, i.e., each visible point on the ground represents a ray that is vertically projected to a point on the water surface mesh above it (see Figure 5.3). The ray is then refracted by the surface and the intensity along the ray is derived based on how close the its direction is to the vertical direction (sun direction) and how far it has traveled to the surface. The resulting caustics mesh (per-vertex) or texture (per-pixel) is rendered using additive blending on top of the ocean floor (shown in the rightmost image in Figure 5.3).

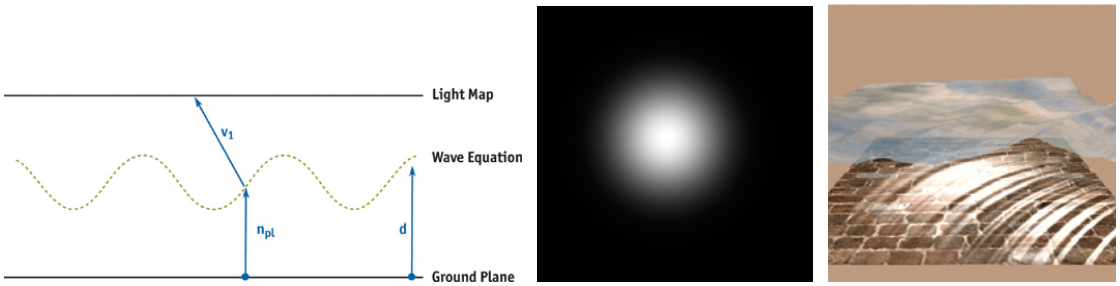


Figure 5.3: The leftmost image illustrates how points on the ground plane are vertically projected to the points on the water surface represented by a wave equation, are refracted and sample a precomputed light map (the image in the middle) that corresponds to sun shining from directly above. The rightmost image then shows the rendered caustics. (*Images from Guardado et al. [35].*)

5.1.2 Image-space Techniques

These techniques usually work by tracing photons in image-space or tracing in world space but only with simplified scene, such as a horizontal plane or other assumptions. These

approaches work similarly to shadow mapping: a scene is rendered from the light's point of view into a texture called *caustics map*. This texture consists of the visible refractive or reflective geometry. The rays are then generated by sampling the textures and the approximate intersections of the rays with the scene receivers are found. The intersection is usually found by marching the shadow map depth buffer (from the light) or the camera depth buffer (from the camera).

The techniques suffer from the same visibility limitations as shadow mapping (ignoring issues with resolution): only caustics that are visible from the light are drawn, furthermore it cannot account for reflected/refracted rays that are occluded in both the camera view and the light view.

Jensen et al. [45] compute caustics for a water surface tile represented as a mesh. They consider only first order rays and a diffuse surface at a constant depth. They use beam-tracing approach, illustrated in Figure 5.4. For each specular triangle of the water surface a caustic triangle at the bottom is computed by intersecting refracted rays with the ground plane. These caustics triangles then form a *caustic mesh* that is drawn into a texture . The final intensity of a caustic triangle is based on the ratio of areas of the specular triangle and the caustic triangle. They also make the caustics texture tileable, which also results in an applied anti-aliasing to an effect of a 2 x 2 super sampling. The texture's UV coordinates are computed by projecting the texture from the height of the water surface in the direction of a ray from the sun, similarly to a parallel projection. Finally, the sampled intensity is multiplied by dot product between the surface's normal and direction to the sun.

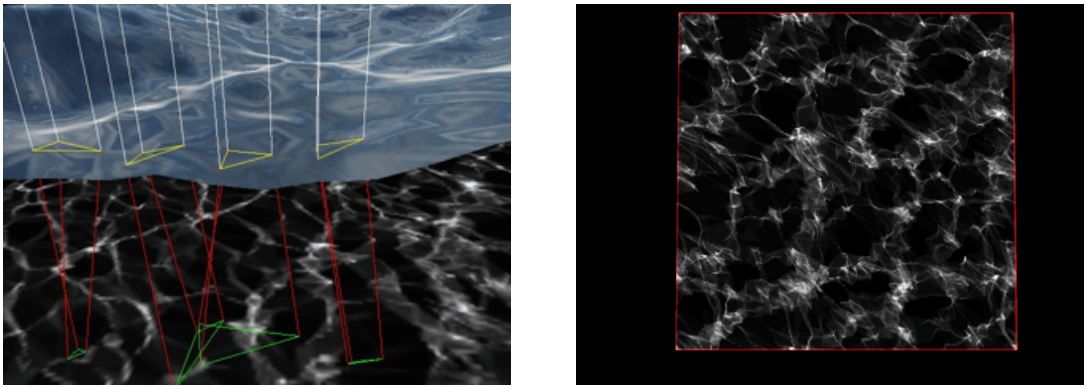


Figure 5.4: The image on the left shows caustics triangles (i.e., triangles at the bottom plane) created by intersecting the refracted rays of each vertex of surface triangles (i.e., on the water surface). The image on right shows the final seamless 256x256 caustics texture with applied 2x2 anti-aliasing. (*Images from Jensen et al. [45].*)

Baboud et al. [11] compute caustics using a two-pass photon-mapping like algorithm that considers only one specular bounce and involves both CPU and GPU. The first pass renders the water surface from the light source into a photon texture (illustrated in the leftmost image of Figure 5.5). Each texel of the texture records the coordinates of the intersection of a light ray (after being refracted by the water surface) with the ground , and a photon contribution. The contribution is computed based on the Fresnel transmittance, the traveled distance, the incident angle and a ray sampling height. The photon texture is then transferred to the CPU-side to construct an *illumination texture* by accumulating at

each sample the stored intensity to the position that is stored at that sample. The texture is then transferred back to the GPU where it is used for lighting in the final render pass.

Based on the resolution of the two textures (the photon texture and the final caustics texture) the caustics can be very noisy and exhibit aliasing. To reduce aliasing a random jittering to the light rays can be applied (as illustrated in the Figure 5.5). The caustics texture is also blurred using an *edge preserving filter* (a GPU-based *bilateral filter*) to preserve the natural high-frequencies (as shown in the two rightmost images in Figure 5.5).

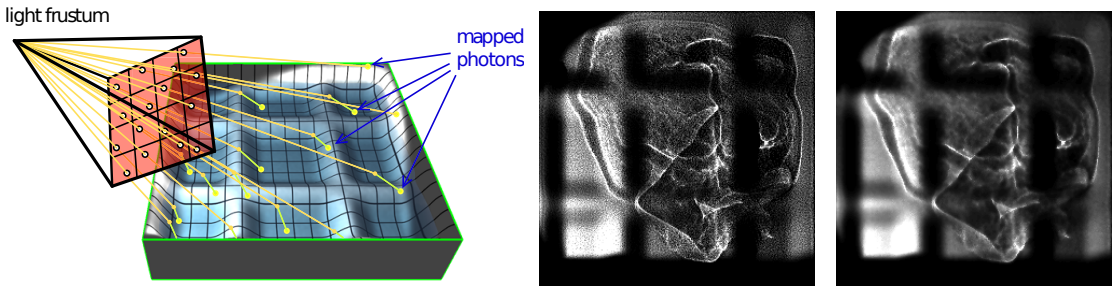


Figure 5.5: The leftmost image shows photons emitted from jittered pixel positions of the light frustum through the water surface and onto the ground. The last two images show, from left to right: raw caustics texture, its filtered variant. (*Images from Baboud et al. [11].*)

Shah et al. [86] present a real-time technique for rendering caustics from reflective and refractive objects, called *caustics mapping*. The algorithm allows for fully dynamic non-planar geometries. First, from light's view, the receiver geometry is rendered into a positions texture, then the caustic-forming objects are rendered to generate refracted rays at their vertices. Each such ray finds an estimate intersection with the receiving geometry by interatively sampling the positions texture. Points are splatted into the caustics map by accumulating the computed light intensity using additive alpha blending. The light intensity is estimated from the total flux contribution for each vertex of the caustic-forming object's mesh based on the area of triangles around the vertex to the total area of the mesh. They also account for the absorption as the ray travels through a refractive object.

The splatting of point primitives leaves gaps in between. Either the geometry of the caustics-forming objects must be further tessellated and/or the caustics map must be low pass filtered (see the Figure 5.6). Also suffers from problems similar to shadow mapping, such as: fixed resolution of the caustics map, visibility issues, temporal incoherency, etc.

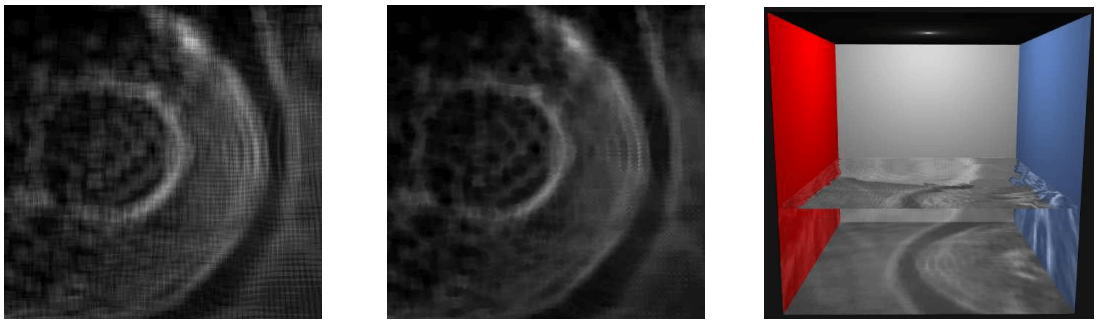


Figure 5.6: Images from left to right show: raw, unfiltered caustics map of a water ripple; the same filtered caustics map; refractive caustics from animated water surface splatted onto the receiver's geometry. (*Images from Shah et al. [86].*)

Wyman et al. [95] propose adaptive caustics maps with deferred shading. They try to mitigate aliasing issues of caustics mapping, of the similar kind to shadow mapping, by adaptively sampling photons instead of the uniform sampling approach. Refractive objects are sorted and then rendered from back to front, after the background opaque geometry, into buffers. Then the photons are generated only on the refactors by sampling the buffers.

A coarse grid of photons is adaptively sampled in 2×2 clusters of photons based on two metrics. The metrics work with the clusters, by either discarding them, or by dividing each photon in the cluster into its own cluster. The first metric tests whether all the photons in a given cluster miss the refractor, if yes, then none are output, else the photons are divided until a certain depth, in which case they are splatted into a caustics map. The second metric works as the first, but continues dividing the cluster until it converges to a single texel in the caustic map.

Furthermore, the photons are generated on the fly using the geometry shader, so there is no need for an explicit photon buffer. This allows for higher amounts of photons to be splatted, while maintaining interactive frame rates.

Meenrattanakorn et al. [54] present approach for rendering both refractive and reflective caustics caused by single specular interface, such as water surface. The approach is able to render caustics outside of the light frustum by considering *cascaded* image spaces, i.e., the rays are traced from both the light space and the view space. They assume deferred shading pipeline used with Reflective Shadow Maps (RSMs) [23] of the light view and G-buffer of the camera view, and additionally a separated depth buffer for opaque geometries behind the refractive object.

RSMs are sampled randomly using blue noise to generate rays and scale the sampled flux by the ratio of the RSMs resolution to the total number of samples. Irrelevant points are removed in order to trace rays only from specular surfaces (using roughness cutoff) and with luminance, which is calculated using the flux, that is high enough to influence the caustics. Then, a Fresnel factor is computed to probabilistically determine whether the ray is refracted or reflected.

The ray is traced by ray marching the depth buffers: first, from the light view, then the end position is projected to the camera view where it continues marching until the hit is found. The hit point is discarded if the dot product between the ray's direction and a sampled normal is negative. The final hit point is found by interpolation between information from view-space depth and the scene geometry.

The resulting hit points are then saved into a buffer as a tuple of clip-space coordinates and the irradiance is computed. The irradiance is calculated using the world-space area of the pixel, similarly to Kim [48] (described in a section below). The buffer is then used to draw the hit points as point primitives into a 2D photon map.

Finally, the photon map is denoised spatially and temporally. It uses the G-buffer's normal, depth, and history buffers along with a motion vector. Radiance towards the camera is then evaluted by considering *Lambertian reflectance* for diffuse surfaces.

Even though it can account for caustics outside the light frustum, the light path must be contiguously visible in the light and camera views, otherwise the caustics are lost. Furthermore, panning the camera might show noise in certain situations due to very sparse photon area. And finally, because the irradiance is computed in image space, other than diffuse surfaces are not physically plausible, since the incident vector is not stored in the photon map the radiance due to specular reflectance cannot be evaluated.

Figure 5.7: Rendering of two scenes with a specular interface producing refractive (emphasized by orange rectangle) and reflective (emphasized by green rectangle) caustics. The reflective shadow maps resolution is 2048×2048 with up-to 2 million photons. (*Renderings by Meenrattanakorn et al. [54].*)

5.1.3 Ray Traced Approaches

These techniques usually trace rays against the whole dynamic scene in world space using DXR. In this case, DXR bounding volume acceleration structures are assumed to be in place. Furthermore, deferred rendering system is employed, with G-buffer for rendering including the depth buffer. Due to rays being traced in world space, these techniques do not suffer from the same visibility limitations as image space techniques do.

Finally, to reach eliminate noise, denoising is applied. The objective of denoiser is to estimate the density of the photons in a particular pixel by gathering nearby photons and accumulating their contribution at each pixel. This is comparable to finding a full caustics light path [54].

Kim [48] presents an approach called *screen-space photon mapping* (SSPM) to render reflected and refracted caustics. Instead of storing the photons as volumes in the world space, as it is with image-space techniques, the photons are evaluated and stored directly in screen-space, this is advantageous for deferred shading.

Photons are emitted only to areas bounded by so-called *projection volume* — a bounding box that contains objects generating caustics. The bounding box is then projected to the negative direction of directional light to obtain a rectangular area that is used to compute emission flux for each photon proportionally to the area it takes in the photon grid.

Photons are traced until certain conditions are met or until a surface hit is opaque enough. Any photon that hits a location out of screen or that has negligibly small intensity, or is beyond screen depth is discarded. Finally, photons that were not discarded are compressed into a single pixel no the screen. The final radiance is computed by normalizing the intensity by area of the pixel in world-space units. Due to the radiance being computed in screen-space, the information stored in G-buffer can be used to evaluate specular and diffuse colors based on BRDFs of the surfaces.

The rendered result is shown in Figure 5.8. Reflection denoiser from NVIDIA GameWorks was used to gather photons based on camera data, depth buffer, roughness buffer and normal buffer. This technique produces artifacts on close-up due to the high resolution of photons near the screen frustum's near depth.

Figure 5.8: Images show refracted caustics by water surface in Cornell Box scene rendered with 2k × 2k photons. Left image shows the raw output, while the right shows the denoised variant. (*Renderings by Kim [48].*)

Gruen [34] renders one-bounce caustics in a scene with a refractive interface, e.g., water surface. The interface is assumed to be represented by a triangular mesh in order to compute the triangular ratio similarly to Jensen et al. [45]. The approach is summarized in five steps:

1. A refracted mesh of the interface mesh is computed by intersecting rays, which originate from each interface's vertex and point in the refracted direction of the incident light, with the scene's receivers. For each interface triangle there is a resulting refracted triangle making up the refracted mesh.

2. A compression ratio is computed for each of the refracted triangle as the ratio of areas of an interface triangle and its corresponding refracted triangle. This, so-called *compression ratio*, is thought of as a metric of how much the light is focused or defocused.

3. A number of refracted light rays is adaptively generated at random locations within each triangle of the interface. The number of samples is based on the compression ratio: the higher the ratio, the more samples are generated; and on the on-screen size of the refracted triangle: the closer the triangle is to the viewer, the more samples are generated.

4. Each generated rays is then again intersected with the receivers in the scene. The intersection is then projected to a screen space pixel where it is accumulated only if it corresponds to the front-most pixel at that position. A brightness value gets accumulated into a caustics buffer. The value can be scaled by the compression ratio and/or the amount of light that has been absorbed due to a medium, similarly to Baboud et al. [11].

5. Finally, the accumulated caustics buffer is denoised at half-resolution using a set of iterated cross-bilateral blurring steps, that account for differences in view-space, depth, normal and view-space positions. Figure 5.9 shows both the raw caustics buffer and the denoised buffer.

Some refracted triangles might be off-screen while some of its generated rays might still produce on-screen caustics. This can be solved by also considering (in Step 3) three additional triangles at different distances along the rays that form the beam. Furthermore, the quality of the compression ratio depends on the geometric complexity of the surface mesh.

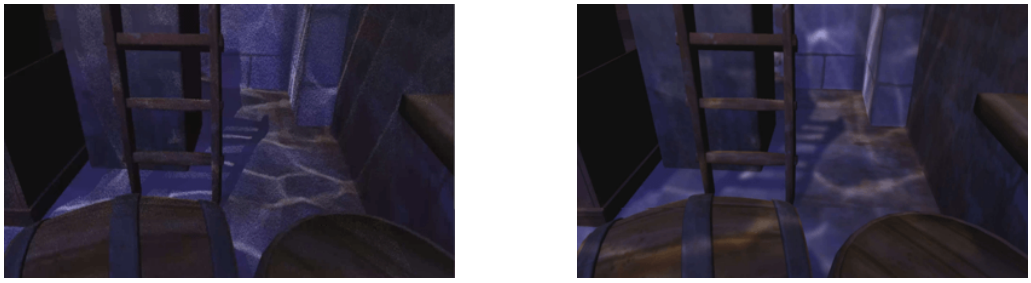


Figure 5.9: The left image shows the raw caustics buffer, and the right images shows its denoised variant. (*Renderings by Gruen [34].*)

Yang et al. [96] present two specialized approaches for rendering caustics cast by water surfaces after one-bounce reflection or refraction. They build on the work of Gruen [33], *Ray-Guided Water Caustics*, the continuation of his previous work [34] on surface caustics, (explained in Section 5.2 about volumetric caustics). They intensify the optical details in water caustics by applying any of the two approaches: *Photon Difference Scattering* (PDS) or *Procedural Caustic Mesh* (PCM). They support most light types (including area lights) and also introduce *cascaded caustics maps* (CCMs), i.e., caustics map at various resolutions to cover vast scene regions.

The PDS technique treats ray hit points as photon footprints. The initial size of the footprint is determined by the resolution of the caustics map and the desired precision set by the user, then its scaling factor, an analogy to compression ratio in Gruen's method [33], is computed. The adjacent hit points are computed using photon differentials, i.e., using the finite difference of nearby rays (as illustrated in Figure 5.10). Each photon footprint is then rendered as quad sprite and blended with the scene. The quad is placed perpendicular to the ray direction and then it is projected to the opaque surface. The intensity is cosine weighted by the incident angle and lighting is evaluated using the scene materials. Using elliptic shape instead of quad forms more continuous patterns even without any denoising (as shown in Figure 5.10). Lower resolutions of caustics map produce blurry caustics, therefore this technique is coupled with the CCMs.

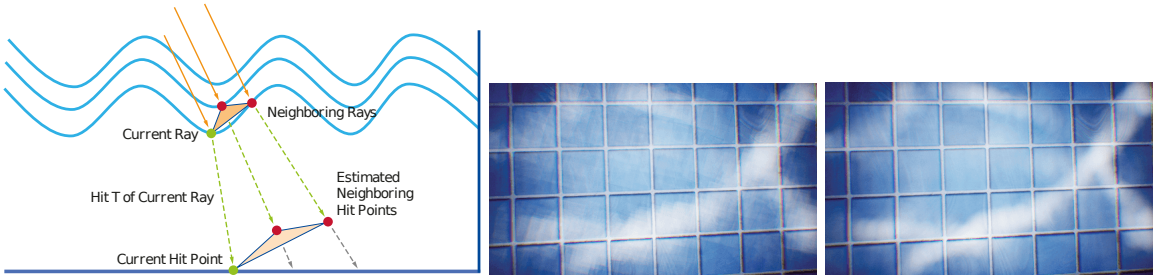


Figure 5.10: The Leftmost image illustrates the size of a photon footprint as the combination of ray's origin, direction and hit point with the adjacent rays' origins and directions. The hit points of adjacent rays (red) are estimated using photon differentials from the current ray's hit (green). The other two images show the footprints' shape: rectangular (left) and elliptic (right). (*Images from Yang et al. [96].*)

The other approach, PCM, converts the hit points of the regular grid in the caustics map to an intermediate mesh with triangular topology. Any invalid primitive is discarded and contribution and intensity is computed according to its world-space area. Finally, the

mesh is rendered into the caustics buffer and blended with the scene. It always produces sharp caustic patterns, however, it suffers from "black edge" artifacts at objects corners where the triangles are culled.

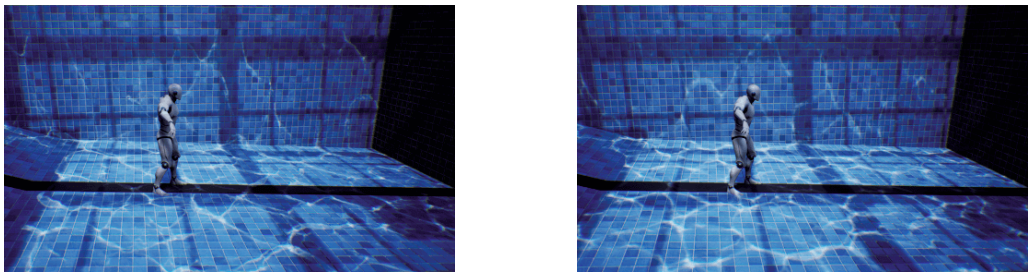


Figure 5.11: Comparison between caustics maps, with resolution 2048×2048, generated using procedural caustic mesh (left) and photon difference scattering (right) methods. (*Renderings from Yang et al. [96].*)

5.2 Volumetric Caustics in Single-Scattering Media Methods

Volumetric caustics are crepuscular rays or godrays underwater. It is the phenomenon when light rays are being visible as they propagate through a water volume. It is caused by photons being out-scattered from their path due to the presence of particles in participating medium.

This section focuses on approaches that render volumetric caustics in single-scattering water volumes. The focus is mainly on refractive caustics, while most of the algorithms support also reflective caustics.

The methods investigated here are approximations of the photon mapping algorithm and most of them compute both surface and volumetric caustics, but only volume caustics are of the interest here. The techniques described below are divided into two categories based on the level of abstraction, or more specifically, based on the primitive that is rendered: line-based approaches and beam tracing approaches.

5.2.1 Line-Based Approaches

These approaches work analogously to point splatting methods of surface caustics — by splatting line primitives instead of point primitives. Lines start the origin of the refracted ray and terminate at the intersection, while visible pixels on the rasterized line approximate in-scattering contribution. The visibility problem is solved by using a depth buffer and additive blending. Analogously to splatting point primitives, these methods also suffer from noisy results needing a greater amount of line primitives. This implies fill-rate intensive applications are instead oftentimes rendered at lower resolutions than the target screen, or must be blurred.

Papadopoulos et al. [71] utilize the image-space ray-scene intersection method by Shah et al. [86] by tracing a uniformly distributed grid of photons form the light source. Scattering is simulated per photon by rasterization of lines between the water surface and scene intersection points. The light source tracks the camera to rasterize lines only in visible parts of the camera frustum.

The final intensity along a line is computed by applying Beer's law and Mie scattering phase function, accounting for attenuation due to the distance traveled underwater and towards the viewer. The intensity from all the lines are additively blended and low pass filtered (i.e., blurred). Figure 5.12 shows the output of this method.

Figure 5.12: Rendering of method by Papadopoulos et al. [71], uses photon grid of 200×200 accumulated into 128×128 render target.

Hu et al. [41] present a line rendering method similar to Papadopoulos et al. [71] but also account for multiple refractions and heterogeneous media.

To compute intensity along a line, the line is sampled around an intersection point $\mathbf{x}_l$ (between the viewing ray the light ray, shown in Figure 5.13) at few regular steps. The final radiance is then the sum of radiances of each sample weighted by Gaussian distribution. That is, the closer the sample is to $\mathbf{x}_l$, the greater is its influence on the final radiance. For heterogeneous media, the transmittance must be sampled and accumulated by marching along the line. The step size can vary based on the frequency of changes in the medium.

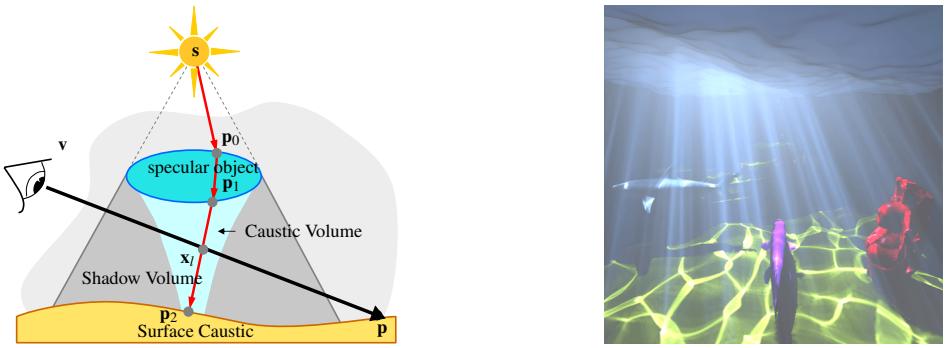


Figure 5.13: Left image shows a line drawn from exit point $\mathbf{p}_1$, on the underside of the specular object, to surface point $\mathbf{p}_2$. $\mathbf{x}_l$ is the intersection point between the view ray and the light ray. Right image shows the output with volume caustics buffer of 1024×1024. (*Images from Hu et al. [41].*)

5.2.2 Beam Tracing methods

This category of techniques traces the intensity changes inside the caustic volumes using triangular beams or more generally, prisms. Triangle beams are composed of the top part,

the specular triangle of the water surface, and the bottom triangle, the caustic triangle, which is found by intersecting the refracted directions of incident light from each vertex of the specular triangle with the ocean bottom (shown in Figure 5.14). The vertices of these two triangles are connected in a fashion that a triangular beam is formed. As the light focuses and defocuses due to the curvature of the water surface, the refracted directions at the vertices of the specular triangles point in different directions causing the beam to warp, so-called *warped caustic volumes*, forming bilinear patches on sides (shown in Figure 5.14). Such volumes cannot be easily rasterized, therefore bounding volumes are used to approximate the caustic volumes.

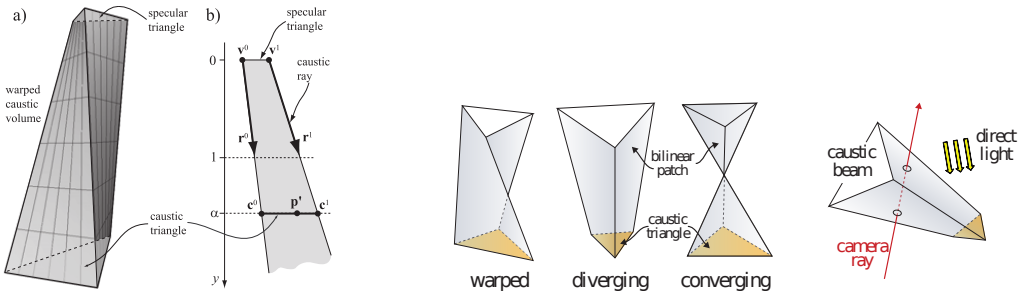


Figure 5.14: Left Image by Ernst et al. [27]: A warped triangular beam is illustrated and cut through, the top triangle is called specular triangle and bottom triangle is the caustic triangle. It is formed by intersecting the bottom with each of the refracted directions at each vertex of the specular triangle. Right image by Liktor et al. [51]: Shows the different types of caustic beams due to the divergence and convergence of light.

Jensen et al. [45] uses beam tracing to surface caustics. Triangular beams are formed with intensities according to the compression ratio: a ratio of the area of the specular triangle to the caustic triangle. The beams are intersected with horizontal planes to form volumetric caustics by computing the ratio for each new caustic triangle that is the result of the beam intersecting a plane. This results in a blocky appearance of volumes, due to the ratio being fixed throughout the volume.

Ernst et al. [27] presents a technique that uses so-called *warped volumes*, an optimization that allows for interpolation of intensities inside the volume, thus reducing the blocky appearance resulting in smoother caustics. Homogeneous media are assumed along with generators (the specular objects) represented only as triangular meshes.

The receivers are not intersected directly, rather, the specular triangles of the generators are extruded and their axis-aligned bounding boxes (so-called *bounding prisms* method, on CPU) are rasterized. Then for each pixel the bounding volume is intersected using a scan plane along the ray from the camera (illustrated in left image of Figure 5.14). Intersections of the refracted directions of each vertex of a specular triangle forms a caustic triangle. This triangle is then intersected with the camera ray to obtain a point $\mathbf{p}$ inside the triangle and the distance traveled through the beam Δ. The radiance seen from the camera is then computed analytically and accumulated using additive blending, accounting for the attenuation due to the distance d from the camera to $\mathbf{p}$, and from $\mathbf{p}$ to the specular triangle:

$$L = \Delta e^{-\sigma_t(d)} \sigma_s p(\theta) L_i(\mathbf{r}), \tag{5.1}$$

where $\mathbf{r}$ is the direction of the refracted light inside the beam, L_i is the incident radiance of the caustic beam, and θ is the angle between the camera ray and $\mathbf{r}$ directions. The L_i

is computed for point **p** by, first, transforming the point to the space of them (called *beam space*), and second, by trilinearly interpolating the radiance values at the vertices of the caustic triangles using barycentric coordinates. The beam space allows for the interpolation of the compression ratio inside the beam.
Liktor et al., 2010 [50] extend the method proposed by Ernst et al. [27] in the following ways:

- A separate triangular mesh is used for the beam generation, so-called *projected grid*, instead of the mesh geometry of generators. The grid is projected onto the specular objects by rendering them from the light source. This way, the algorithm is independent of the geometric complexity of the generator meshes.
- Provides full GPU implementation by extruding the beams in geometry shader.

The scattered radiance towards the camera is computed the same way (using the scan plane), i.e., single-scattering in homogeneous participating media, that is additively blended.
Liktor et al., 2011 [51] extend the previously mentioned technique by the authors by introducing:

- hierarchical projected grid: a projected grid that is adaptively refined in geometry shader according to heuristics, such as on the discontinuities of the specular surface, based on the distance to the viewer, and whether the beams are convergent or divergent.
- bounding volume extrusion: even more tightly fitting bounding box around the bounding volumes is found by modification of the original bounding prism method, drastically improving for the worst case when the beam is warped, leading to less fill-rate.

The radiance scattered towards the viewer is computed the same way as in the previous technique — by intersecting a beam with a scan plane from the camera. The rendering of this method is shown in Figure 5.15.

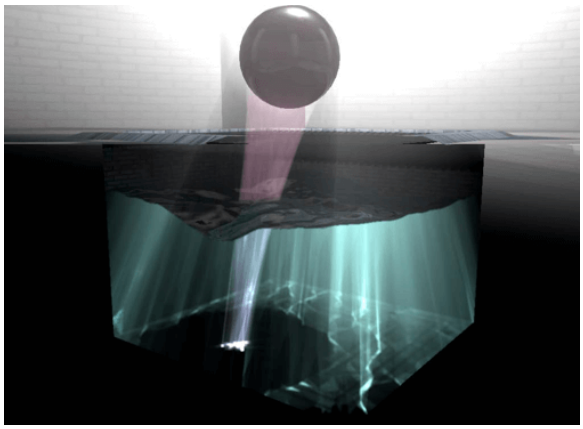


Figure 5.15: Rendering of method by Liktor et al. [51].

Gruen [33] extends his method for surface caustics [34] (described in Section 5.1) to render both surface and volumetric caustics in single scattering participating media using ray tracing and rasterization pipelines. The bounding volumes problem is approximated by adaptively tessellating the triangular beams into smaller beam volumes.

This in-scattering integral is approximated by tessellating the caustics volumes into small enough volumes that allow for the accumulation of the in-scattering term only at the frontfacing triangles of each volume. This way, there is no need for the camera to intersect the beams directly. Rather, the rastarization pipeline is utilized, along with the ray tracing pipeline to find intersections with a dynamic scene. In comparison to the surface caustics method by the author, this method is more concerned with refining the beams rather than the surface caustics. This method reuses ideas from caustics mapping, in the following way:

1. The compression ratios are computed for the refracted water mesh as the ratio of the areas of the water triangle and the refracted triangle. The water surface mesh must be detailed enough to facilitate high-enough-quality compression ratio.

2. The visible portion of the water surface is rendered from the light's point of view into *caustics map* — a texture containing positions and normals of the surface.

3. Each pixel of the caustics map forms a refracted ray that is intersected with the scene and the intersection is stored in the *refracted caustics map* at the same pixel position.

4. Surface caustics are rendered by projecting the intersection points from the previous step to screen space. Then testing, whether they are the frontmost pixel on the screen. The radiance value is computed, accounting for absorption by the distance traveled, similarly to Baboud et al. [11], scaled by the compression ratio, and accumulated into photon map using atomic add operations (analogous to additive blending, because many intersection points can end up on the same screen location).

5. The water triangles are adaptively tessellated by drawing the water mesh. The compression ratio is thought of as the likelihood that a triangular beam forms a non-convex volumes, therefore, the higher the ratio the higher the tessellation factor. Furthermore, it is scaled to account for any light leaks doe to small scene features, or to approximate the in-scattering integral enough. This is rather artistically controlled.

6. The geometry shader is used to build the triangular beam volumes by projecting the tessellated points to the caustics map and the refracted caustics map to obtain positions of the vertices of the top and bottom triangle, respectively. The vertices of the triangles are then connected to form a triangular beam.

7. Finally, the volumes are rasterized and the in-scattering is computed on the pixels of the frontfacing sides of each volume, evaluating phase function for the interpolated ray direction inside the volume, multiplied by the interpolated thickness of the volume. The radiance is additively blended into an offscreen buffer that can be directly used in deferred shading pipelines.

The resulting volumetric caustics buffer can be further denoised similarly, to the surface caustics method.

Chapter 6

Design of Technique for Rendering Underwater Scenes

The goal of this thesis is to visualize underwater scenery in high quality using a selection of graphics techniques. Theory behind rendering participating media was given in Section 2.2. The underlying field of ocean optics was studied and is described in Chapter 3.

Then, various techniques for simulation and rendering ocean environments were investigated in Chapter 4. Specifically: the simulation of water surfaces, existing bio-optical models that abstract the optical properties of waters, participating media approaches that render such media, and procedural generation for generation and representation of various surfaces that bound the water body. Chapter 5 then focused solely on the techniques that simulate lighting effects inside water bodies, i.e., caustics.

First the scene representation is proposed. Then the technique to render such scene is designed. A subset of the designed techniques was implemented and the results are shown in Chapter 7.

6.1 Scene Representation

As described in the previous chapters, light transport in underwater environments is influenced by many complex processes and components. The radiative transfer equation connects the optical properties of the water body with the light within the water. Before the light can be simulated in the water body, the boundary conditions and the optical properties of the water body must be defined. The scene is composed of the following components that try to reproduce an oceanic environment:

1. Sky model: the Preetham's precomputed sky model [77] was used.
2. Water surface model: based on the Tessendorf's choppy model [90] for simulation of waves of deep ocean waters.
3. Terrain: the ocean floor, simply terrain, is modeled as a signed distance field/function in combination with fractal Brownian Motion approach.
4. Bio-Optical Model: a physically-based model that tries to reproduce optical properties of case 1 waters. A vertical distribution of phytoplankton is defined and based on that, the other optical parameters are deduced, this results in vertically non-uniform parameters.

The following sections describe these models in more detail. The description and survey of sky models are out of scope of this thesis. The choice of the sky model was driven purely by performance and ease of use.

6.1.1 Water Surface Model

Water surface is modeled using the Choppy waves method by Tessendorf [90] for deep ocean waves. Therefore the surface is represented as a heightfield, specifically, as a wave height field computed for each horizontal point (in Equation 4.3) on a synthesized patch of ocean using IFFT. The Phillips spectrum, as proposed by Tessendorf [90], is also used. It is a semi-empirical model for wind-driven waves in a fully developed sea based on statistical analysis of observations of the ocean surface. The spectrum sampled in the fourier domain using gaussian random number generator, with mean 0 and standard deviation 1.

Apart from the wave height field, $h(\mathbf{x}, t)$, the slope vectors $\mathbf{s}(\mathbf{x}, t)$ and the displacement vectors, $\mathbf{D}(\mathbf{x}, t)$, are also computed using IFFT for each horizontal position $\mathbf{x} = (x, z)$ and time point t. Then the three-dimensional position $\mathbf{p}_w$ on a grid on the water surface at t after displacement is:

$$\mathbf{p}_w = \begin{bmatrix} x + uD_x(\mathbf{x}, t) \\ h(\mathbf{x}, t) \\ z + uD_z(\mathbf{x}, t) \end{bmatrix}^T, \tag{6.1}$$

where $u \in \mathbb{R}$ is a user controlled parameter to scale the displacement vector, usually kept at value 1.

The normals $\mathbf{n}_w$ are computed using the slopes $\mathbf{s}(\mathbf{x}, t)$. The slopes must also account for the displacement [30]:

$$\mathbf{s}(\mathbf{x}, t) = \left[\frac{\frac{\mathrm{d}h(\mathbf{x}, t)}{\mathrm{d}x}}{1 + u\frac{\mathrm{d}D_x(\mathbf{x}, t)}{\mathrm{d}x}}, \frac{\frac{\mathrm{d}h(\mathbf{x}, t)}{\mathrm{d}z}}{1 + u\frac{\mathrm{d}D_z(\mathbf{x}, t)}{\mathrm{d}z}} \right], \tag{6.2}$$

$$\mathbf{n}_w = \frac{(-s_x, 1, -s_z)}{||(-s_x, 1, -s_z)||}, \tag{6.3}$$

where D_x and D_z are partial derivatives of $\mathbf{D}$ in x and z dimensions, respectively, these must also be computed separately.

Concerning optics, the water surface is treated as collection of small planar facts, each tilted according to its normal. Then, the intersection of light ray with the surface at point $\mathbf{p}_w$ is treated according to the equations and assumptions presented in Section 2.3.1.

6.1.2 Terrain Representation and Generation

Ocean floor or seabed is modeled as a terrain. The terrain must be represented in a way that facilitates simple ray-surface intersection. This is accomplished through the combination of signed distance functions (SDFs), therefore, the terrain is a signed distance field. Then the intersection of a ray with the terrain can be found through the use of ray marching with distance estimation — sphere tracing — as described in Section 4.4.

Quilez [81] discusses how the detail obtained in a manner described by fractal Brownian Motion (fBM) can be used with SDFs. Traditional 3D noise is not a distance function, it is a signed field/function that does not measure distances. A 3D noise that is also SDF is obtained by summing multiple octaves/layers of sphere SDFs, in the following way:

1. the base noise/function for fBM must be a smooth SDF, e.g., big sphere,
2. an infinite grid of spheres of random sizes is computed using domain repetition operator, each sphere has a random radius that is smaller than half the edge-length of the grid,
3. for a given point in space, the SDF is evaluated for the 8 spheres at each corner of a grid cell,
4. instead of summing, each additional layer is combined with the base SDF using the SDF operators, such as *min, smoothmin*.

Combination of multiple layers of spheres with increasingly smaller radii can produce highly detailed 3D terrain that is also SDF.

6.1.3 Bio-Optical Model of Water Volume

The bio-optical model of the water volume is represented by the following inherent optical properties (IOPs): absorption coefficient, scattering coefficient, and phase function. Optical properties of the following constituents are considered: water, phytoplankton, and colored dissolved matter (CDOM, commonly known as yellow matter). Only elastic scattering is modeled. While both water and phytoplankton show elastic scattering, the elastic scattering produced by CDOM is shown to be negligible [15] and is therefore ignored.

The bio-optical properties are modeled to represent waters with high phytoplankton concentration, in which optical parameters can be correlated to chlorophyll concentration. It is assumed that the vertical distribution of the properties is non-uniform and that it correlates to chlorophyll concentration. The vertical distribution of *chlorophyll concentration* C at depth z is approximated using a Gaussian function, as employed by Ma et al. [52]:

$$C(z) = C_0 + \frac{h}{\sigma\sqrt{2\pi}} exp\left[-\frac{1}{2}\left(\frac{z - z_{max}}{\sigma}\right)^2\right], \tag{6.4}$$

where C_0 is the background chlorophyll concentration, z_{max} is the depth of the maximal chlorophyll concentration, σ is the standard deviation, and h determines the amplitude of the maximal chlorophyll concentration. This vertical distribution of chlorophyll concentration is then used to determine the vertical distribution of other IOPs.

Absorption

The *total absorption coefficient* $\sigma_a(\lambda, z)$ is composed as:

$$\sigma_a(\lambda, z) = \sigma_a^w(\lambda) + \sigma_a^p(\lambda, z) + \sigma_a^c(\lambda, z), \tag{6.5}$$

where superscripts w, p, and c stand for water, phytoplankton and CDOM, respectively.

The spectral absorption coefficient of water is modeled according to measurements of the absorption spectrum of pure water by Pope and Fry, 1997 [76]. Figure 6.1 shows the wavelength-dependent measurements of σ_a^w for the range wavelengths according to visible spectrum. The absorption in the blue (i.e., the blue part of the visible spectrum as shown in Figure 2.1) is very low while increasingly rising to the maximum in red.

The absorption coefficient of phytoplankton is parameterized as follows [52]:

$$\sigma_a^p(\lambda, z) = 0.06\alpha_c^*(\lambda)C(z)^{0.05}, \tag{6.6}$$

where α_c^* is *chlorophyll-specific absorption coefficient* of phytoplankton describing the absorption per unit of concentration for a particular species of phytoplankton, given in m^2/mg. The wavelength-dependent measurements of α_c^* are based on the data provided by Bricaud et al., 1995 [18]. They provide data for absorption spectra of phytoplankton normalized by chlorophyll, shown in Figure 6.2a. The absorption by phytoplankton demonstrates high spectral variations with peak in blue, decreasing approximately exponentially to the red.

Absorption coefficient of CDOM is parameterized as [52]:

$$\sigma_a^c(\lambda, z) = 0.012C(z)^{0.65}e^{-S_y(\lambda-440)}. \tag{6.7}$$

The coefficient is computed by fitting an exponential absorption curve where S_y is the slope taken as a constant, with a value of $0.014\,\mathrm{nm}^{-1}$.

Scattering

The *total scattering coefficient* $\sigma_s(\lambda, z)$ is decomposed into:

$$\sigma_s(\lambda, z) = \sigma_s^w(\lambda) + \sigma_s^p(\lambda, z), \tag{6.8}$$

where the scattering coefficient for water σ_s^w is modeled using measurements for pure seawater by Smith and Baker, 1981 [87]. The scattering coefficient phytoplankton σ_s^p is given by Gordon and Morel, 1983 [32], which has shown that even small concentrations of phytoplankton contribute to the total elastic scattering in the water:

$$\sigma_s^p(\lambda, z) = \left(\frac{550}{\lambda}\right) 0.30C(z)^{0.62}, \tag{6.9}$$

where the constant 0.30 was selected to fit the data collected from many water types, while the upper bound has a value of 0.45 [37].

The phase function is composed of two scattering phase functions:

$$p(\theta, \lambda, z) = \frac{\sigma_s^w(\lambda)}{\sigma_s(\lambda, z)} p_w(\theta) + \frac{\sigma_s^p(\lambda)}{\sigma_s(\lambda, z)} p_p(\theta), \tag{6.10}$$

where p_w and p_p are the scattering phase functions for water and for phytoplankton, respectively. The phase function for pure water is modeled using Einstein-Smoluchowski theory, as given by Gutierrez et al. [37]:

$$p_w(\theta) = 0.06225(1 + 0.835\cos^2\theta). \tag{6.11}$$

The scattering phase function for phytoplankton is modeled using the *Petzold average-particle function* based on measurements of VSFs by Petzold, 1972 [73]. The average-particle phase function, which satisfies the normalization condition, was made as the average of the three VSFs measured in different waters (clear ocean, coastal ocean, and turbid water) and is used as a reference particle phase function [57]. Figure 6.2b shows the average phase function, in general, forward scattering dominates over scattering in backward direction.

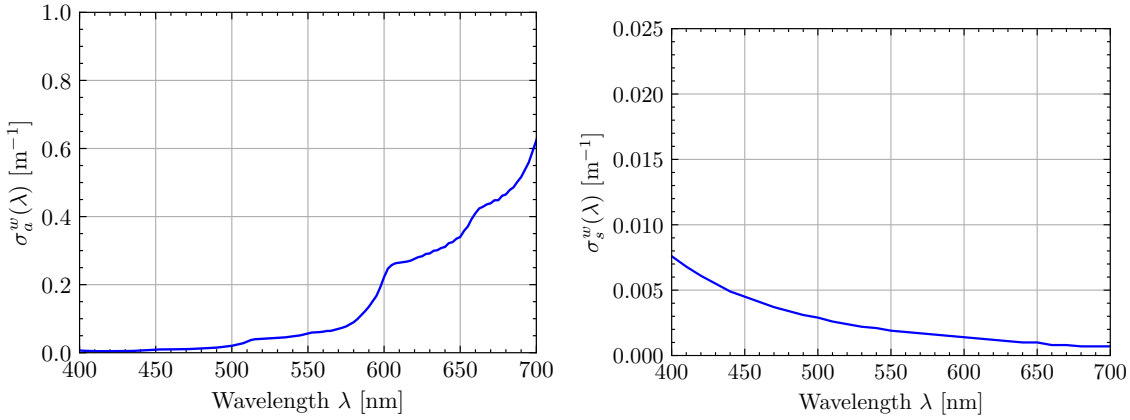
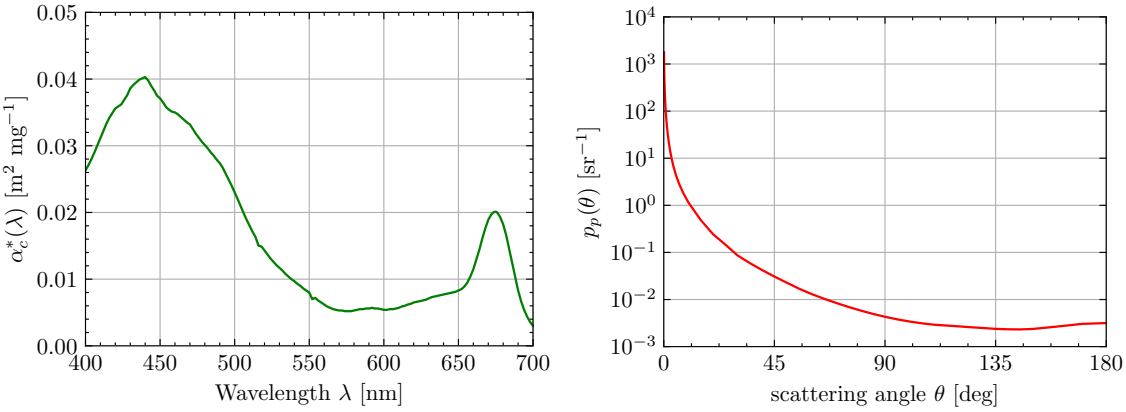


Figure 6.1: Left image shows the absorption coefficient of pure water measured by Pope and Fry, 1997 [76]. The right image shows the scattering coefficient of seawater provided by Smith and Baker, 1981 [87].



(a) The chlorophyll-specific absorption coefficient of phytoplankton describing the absorption per unit of concentration for a particular species of phytoplankton, measured by Bricaud et al., 1995 [18].

(b) Log-linear plot of Petzold average-particle phase function, used as phase functions for phytoplankton and particles. The function is the average of phase functions measured in different types of waters by Petzold, 1972 [73].

6.2 Scene Rendering

Using the designed scene representation and the designed bio-optical model, a technique is proposed to render the scene.

A bio-optical model, proposed in Section 6.1.3, tries to reproduce the optical parameters of Case 1 waters. This is accomplished by defining a vertical distribution of chlorophyll concentration that is used to determine all the other parameters. The volume rendering equation (VRE) (in Equation 2.11) must be modified to reflect this depth dependence on the optical parameters. Moreover, it is not feasible to compute the VRE directly especially if interactivity is to be reached. Therefore the VRE must be approximated appropriately for this types of participating media.

Starting from the VRE for non-emissive media (henceforth just VRE),

$$L(\mathbf{x},\omega) = \underbrace{T(\mathbf{x},\mathbf{x}_s)L_s(\mathbf{x}_s,\omega)}_{\text{surface term}} + \int_{t=0}^{S} T(\mathbf{x},\mathbf{x}_t)\underbrace{\sigma_s(\mathbf{x})L_i(\mathbf{x}_t,\omega)}_{\text{in-scattering term}}\,\mathrm{d}t, \tag{6.12}$$

by utilizing the properties of scattering in waters the VRE is approximated specifically for rendering underwater. It is approximated by two independent contributions: single scattering and multiple scattering. Then the radiance entering the camera at position $\mathbf{c}$ from direction ω, as illustrated in Figure 6.3, is then given by:

$$L(\mathbf{c},\omega) = L_{sscat}(\mathbf{c},\omega) + L_{medium}(\mathbf{c},\omega). \tag{6.13}$$

The single scattering part L_{sscat} models the contributions of the surface term and single bounce of the in-scattering term. Specifically, it accounts for: 1) L_s the radiance at the surface point $\mathbf{x}_s$, as a result of the exitant radiance L_o and one-bounce surface caustics at that point; 2) the in-scattering integral due to volume scattering is approximated by volumetric caustics method. The single scattering part is further described in Section 6.2.1.

The multiple scattering part L_{medium} models the contributions from the media by approximating the in-scattering integral based on the general scattering properties of waters. It also accounts for the non-uniform vertical distribution of bio-optical properties. This part is described in Section 6.2.2.

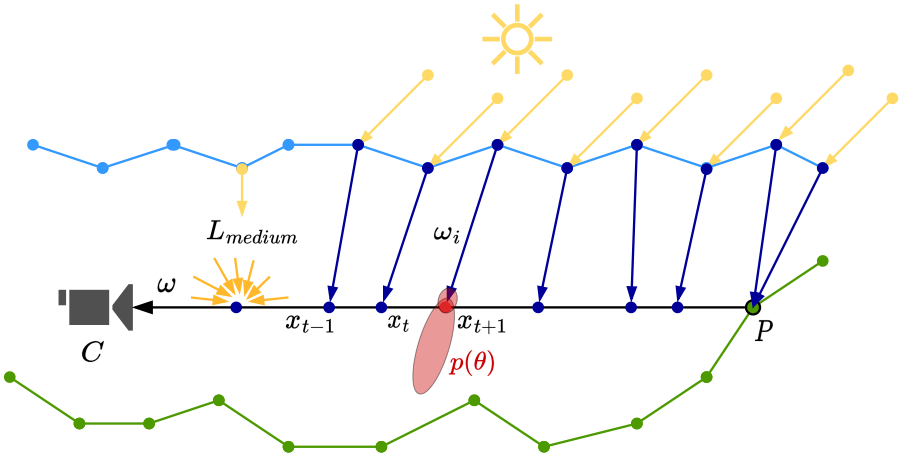


Figure 6.3: Radiance entering the camera at position $\mathbf{c}$ is computed by sampling along the ray from the camera to the closest intersection point $\mathbf{p}$ on a surface. Each sample accumulates transmittance and evaluates contribution due to multiple scattering L_{medium}. The dark blue arrows show contributions due to single scattering at each sample, this is explicitly computed using techniques that simulate caustics.

6.2.1 Single Scattering Approximation

The single scattering approximation, as computed by Wronski [94] and Hillaire [39] cannot be computed in this scenario directly. The refractive interface, represented by water surface, causes the incident light rays to be refracted and additionally, due to the surface being curved, the refracted rays point in different directions. This causes the rays to focus and defocus in various ways, thus forming caustics (as described in Chapter 5).

In the single scattering approximation, the in-scattering integral is computed by ray marching towards the light source for each sample along the view ray. However, for this scenario, there might be multiple light rays contributing to a sample from multiple refracted directions of the incident light, or there might not be any direction that contributes directly (a defocused region). Therefore, the in-scattering integral, as it is accumulated in the single scattering approximation, cannot be computed in this backward ray tracing manner, i.e., by marching the rays from the camera.

Offline approaches can compute this by tracing a large number of rays for each view sample to the water surface, effectively accumulating contributions from all possible directions. There are also real-time approaches that can approximate the in-scattering integral by making aggressive assumptions about the incident light and the interface. All of the aforementioned approaches are not feasible for the proposed representation of the scene. Instead of computing the in-scattering as part of the single scattering approximation, there are specialized techniques that approximate both surface and volumetric caustics — the result of the in-scattering integral.

The single scattering approximation, in Equation 6.14, is then split into parts, where each part is computed separately, in the following way: the surface term, and the in-scattering integral. The surface term is computed the same way as in the single scattering approximation. The evaluation of the in-scattering integral in the volume results in light shafts or godrays, called *volumetric caustics* in underwater environments. The intersection of volumetric caustics with the scene produces surface caustics. These both effects, though part of the same phenomenon, are then simulated explicitly using specialized techniques that target real-time.

$$L_{sscat}(\mathbf{c},\omega) = T(\mathbf{c},\mathbf{p})L_o(\mathbf{p},\omega) + L_{surfCaustics}(\mathbf{p},\omega) + L_{volCaustics}(\mathbf{c},\mathbf{p},\omega), \tag{6.14}$$

Surface Term

The surface term is computed by ray marching along the view ray accumulating transmittance until a point $\mathbf{p}$ on a surface is reached (as illustrated in Figure 6.3). Due to the proposed bio-optical model with non-uniform vertical distribution of media properties, transmittance cannot be computed analytically and must be iteratively accumulated by sampling along the ray.

$$L^{surf}_{sscat}(\mathbf{c},\omega) = e^{-\int_{\mathbf{x}=\mathbf{c}}^{\mathbf{p}} \sigma_t(z)||\mathrm{d}\mathbf{x}||} L_o(\mathbf{p},\omega), \tag{6.15}$$

where z is the depth below the water surface of the sample. L_o is the exitant radiance at surface point $\mathbf{p}$. Underwater surfaces are modeled as diffuse. Since there is a lack of measurements and models of the BRDF for actual ocean bottom materials, the bottom being a Lambertian reflecting surface is a typical and reasonable assumption [58]. On the other hand, the water surface is modeled as diffuse with specular highlights using *Blinn-Phong reflection model* [12]. The exitant radiance from point $\mathbf{p}_w$ on the water surface and from point $\mathbf{p}_t$ on the terrain towards the view direction ω is evaluated, similarly to Baboud et al. [11]:

$$L^w_o(\mathbf{p}_w,\omega) = (1-R_F)L_t(\mathbf{p}_t,\mathbf{t}) + R_F L_r(\mathbf{p}_w,\mathbf{r}), \tag{6.16}$$

$$L_t(\mathbf{p}_t,\mathbf{t}) = T(\mathbf{p}_w,\mathbf{p}_t)L^t_o(\mathbf{p}_t,-\mathbf{t})L_{surfCaustics}(\mathbf{p}_t), \tag{6.17}$$

$$L^t_o(\mathbf{p},\omega) = A_t(\mathbf{p})(\mathbf{n}_t\cdot\omega), \tag{6.18}$$

where R_F is Fresnel reflectance, L_t is the transmitted radiance from point on the terrain $\mathbf{p}_t$ underwater along the refracted direction $\mathbf{t}$, and L_r is the reflected radiance from the reflected direction $\mathbf{r}$ at the point on the water surface, and A_t is the albedo or reflectance of the terrain.

Surface and Volumetric Caustics

Both surface and volumetric caustics are computed as part of one algorithm, thus reusing already computed information, such as intersections. The technique reuses ideas in ray-guided volumetric caustics by Gruen [33], and in *procedural caustics mesh* (PCM) method by Yang et al. [96]. The approach are photon mapping-based and in-scattering integral is approximated using beam tracing. It works in the following steps:

1. A Photon regular grid is created, a triangular mesh independent from the geometry of the water surface.

2. Caustics map is created by rendering the water surface from the point of view of the light, storing positions and normals.

3. Refracted caustics map is then computed: each pixel [x,y] in the caustics map is treated as a ray, and is refracted along the surface. Each ray is then traced in the scene until an intersection is found, which is guaranteed. The intersection is found by evaluating the terrain SDF using sphere tracing. The world space position of the intersection is stored at the same pixel location [x,y] in the refracted caustics map. Additionally, the radiance contribution of the photon is stored.

4. Compression ratios are computed for each triangle of the photon grid by projecting each photon to the caustics map, to retrieve the specular triangle, and to the refracted caustics map, to retrieve the caustic triangle. The ratios are then stored in a buffer.

5. Surface caustics are accumulated in surface caustics map (SCM) in the similar manner as Meenrattanakorn et al. [54]: by storing the photon grid in a vertex buffer and drawing it either as point primitives, or as triangles, addively blending the contributions.

6. Volume caustics are computed by drawing the photon grid as a triangular mesh and projecting each photon to the caustics map, thus sampling the positions and normals of the water surface. Each triangle is tessellated based on the compression ratio and the on-screen size of its caustic triangle (obtained by projecting each photon to the refracted caustics map). From the tessellated vertices a ray is generated to intersect the terrain again thus forming new specular and caustic triangles. These are then built into the triangular beam, which has its frontface rasterized, additively blending computed contribution per pixel.

Both the surface caustics and volumetric caustics produce noise, and therefore must be denoised.

6.2.2 Multiple Scattering Approximation

Multiple scattering is approximated based on the rationale by Monzon et al. [67]. Multiple scattering is caused by the light being scattered a number of times between particles before

reaching the eye and it plays a crucial role in underwater illumination. It is assumed to be very low frequency and so it can be modeled as a function of depth and wavelength. Due to the designed bio-optical model (in Section 6.1.3), which assumes non-uniform vertical distribution of bio-optical properties, the proposed analytical solution by the authors cannot be used directly therefore a solution that is more suited must be derived.

The contribution from the medium (based on VRE for non-emissive media in Equation 6.12) due to multiple scattering is defined as:

$$L_{medium}(\mathbf{c}, \omega) = \int_{t=0}^{S} T(\mathbf{c}, \mathbf{x}_t) L_i(\mathbf{x}_t, \omega) \, \mathrm{d}t, \tag{6.19}$$

$$L_i(\mathbf{x}, \omega) = \sigma_s(\mathbf{x}) \int_{\Omega} p(\mathbf{x}, \omega, \omega_i) L(\mathbf{x}, \omega_i) \, \mathrm{d}\omega_i, \tag{6.20}$$

where $\mathbf{c}$ is the view camera position, ω is the direction toward the view, and S is the distance to the closest intersection along ω (as illustrated by Figure 2.10).

Assuming an isotropic media, where light is distributed uniformly in the sphere, the phase function becomes constant $p(\mathbf{x}, \omega, \omega_i) = \frac{1}{4\pi}$. Furthermore, the in-scattering integral describing radiance from every direction — the incident irradiance — is approximated by the vertical varying incident downwelling irradiance at the surface of the water volume $L_{ws}(\lambda)$ that relates to the diffuse attenuation coefficient for downwelling irradiance $K_d(\lambda)$. This is a fair assumption for general use case, since most of the downwelling irradiance underwater comes from a light cone directly above due to so-called Snell's window (shown in Figure 2.15), even under different zenith angles [67]. The incident irradiance due to multiple scattering will be constant at every depth z below the surface as:

$$L_i(\mathbf{x}, \omega) = L_{ws} e^{-K_d(\lambda)} z. \tag{6.21}$$

As described in Section 3.1.1, $K_d(\lambda)$ is a physically measurable coefficient that measures the irradiance attenuated in the water volume per meter per wavelength. Though it could be obtained from tabulated data (e.g., by Solonenko and Mobley, 2015 [88]) for case 1 waters, it would not account for the non-uniform vertical distribution of bio-optical properties posed by the designed bio-optical model, therefore K_d must be computed.

There are various relations between absorption coefficient and backscattering coefficient σ_{bs} that approximate K_d. According to Lee et al., 2005 [49] the value of K_d can be estimated using the following relation that is based on analysis and numerical simulations of the radiative transfer equation:

$$K_d(\lambda, z) = m_0 \sigma_a(\lambda, z) + \nu \sigma_{bs}(\lambda, z), \tag{6.22}$$

where the parameters m_0 and ν describe the contributions of σ_a and σ_{bs} to K_d, respectively. It was found that $m_0 approx (1 + 0.005\theta_z)$ where θ_z is the solar zenith angle above the water surface, and ν can be expressed as:

$$\nu = m_1 (1 - m_2 e^{-m_3 \sigma_a(\lambda, z)}), \tag{6.23}$$

where the values of m_1, m_2, and m_3 are 4.18, 0.52, and 10.8, respectively. To obtain the value of backscattering coefficient in waters where phytoplankton and other particulate are dominant, Morel, 1988 [68] introduced the following relations:

$$\sigma_{bs}(\lambda, z) = 0.5 \sigma_s^w(\lambda, z) + B_b \sigma_s^p(\lambda, z), \tag{6.24}$$

$$B_b = 0.002 + 0.02(0.5 - 0.25 \log_{10} C(z)) \tag{6.25}$$

Now, having all the necessary terms, the contribution from the medium, from Equation 6.19, is expanded step by step (omitting any wavelength-dependency for clarity):

$$\begin{aligned} L_{medium}(\mathbf{c},\omega) &= \int_{t=0}^{S} T(\mathbf{c},\mathbf{x}_t)\sigma_s(\mathbf{x}) \int_{\Omega} p(\mathbf{x},\omega,\omega_i)L(\mathbf{x},\omega_i)\ \mathrm{d}\omega_i\ \mathrm{d}t \\ &= \int_{t=0}^{S} T(\mathbf{c},\mathbf{x}_t)\sigma_s(z)\frac{1}{4\pi}\int_{\Omega} L(\mathbf{x},\omega_i)\ \mathrm{d}\omega_i\ \mathrm{d}t \\ &= \int_{t=0}^{S} T(\mathbf{c},\mathbf{x}_t)\sigma_s(z)\frac{1}{4\pi}L_{ws}e^{-K_d(z)z}\ \mathrm{d}t \\ &= \frac{1}{4\pi}L_{ws}\int_{t=0}^{S} T(\mathbf{c},\mathbf{x}_t)\sigma_s(z)e^{-K_d(z)z}\ \mathrm{d}t, \end{aligned} \tag{6.26}$$

where z is the depth, or distance to the water surface along the vertical axis, of each sample.

Furthermore, the proposed energy-conserving analytical integration by Hillaire, 2015 [39] is applied. It analytically integrates the scattered light over a range according to both σ_t and a scattered light sample I, as well as an integration depth. Applying this proposed analytical solution to the integration in Equation 6.26, for a light sample $I(z) = \sigma_s(z)e^{-K_d(z)z}$, gives the final form of the contribution from the medium:

$$L_{medium}(\mathbf{c},\omega) = \frac{1}{4\pi}L_{ws}\int_{t=0}^{S} \frac{I(z) - I(z)e^{-\sigma_t(z)||\mathbf{c}-\mathbf{x}_t||}}{\sigma_t(z)}\ \mathrm{d}t. \tag{6.27}$$

This integration requires less number of samples compared to the trivial sampling of the integral to make participating media look correct with increasing density (i.e., extinction).

Chapter 7

Implementation and Evaluation of Technique for Rendering Underwater Scenes

The designed technique from Chapter 6 was not implemented in its entirety. Only a subset of techniques was implemented and experimented with. A standalone C++ application with the use of Vulkan API [4] was developed for this purpose. In Figure 7.1 below is a gallery of screenshots of experiments with those techniques. The application implements deferred shading pipeline and the following components:

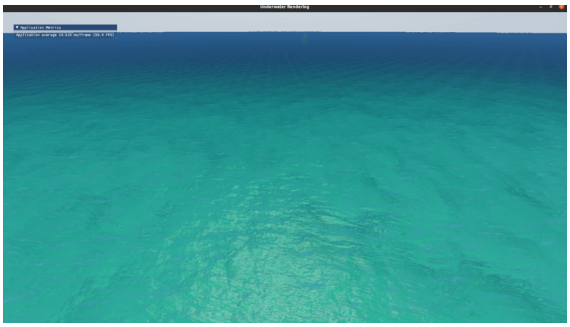
- Water Surface. Was implemented as designed. It is represented as a triangular mesh — a regular grid of 256×256 vertices that is displaced. There are two textures computed each frame: displacement map and normal map (with the same resolution as the grid). The textures are computed on CPU-side using the FFTW library [2]. The execution was parallelized and runs on all available threads. Uses both, the parallel execution of FFTW plans, and OpenMP library to parallelize for loops.
- Camera. An FPS camera was implemented. It uses Euler angles.
- Sky Model. Implements the Preetham's sky model [77].
- Camera G-Buffer. Has the following attachments: world space positions, world space normals, albedo, and depth.
- Terrain. Simple combination of multiple SDFs, possible to use multiple octaves of fBM SDF.
- Surface caustics. Implemented as described.
- Volumetric caustics. Implemented partially, without the tessellation part.
- The multiple scattering approach was implemented fully and tested.

Furthermore, the application implements its own Vulkan abstraction. And additionally uses the following libraries:

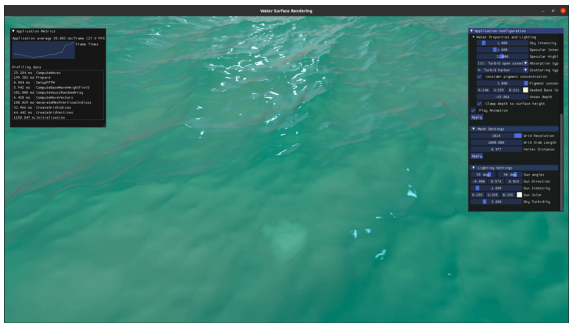
- Dear ImGuI [1]: to expose user-controllable parameters in GUI form,
- CPU side profiler,

- Shaderc library [3]: allows for compilation of GLSL shaders at runtime.

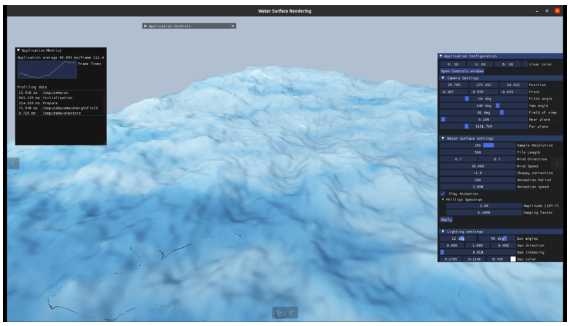
The application can be compiled using CMake. The instructions are given in a readme text file in the root folder.



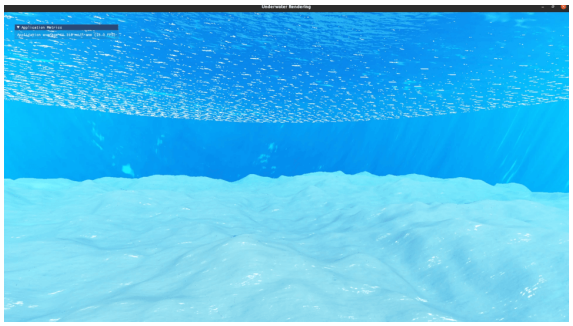
(a) Water surface rendered with camera above the surface.



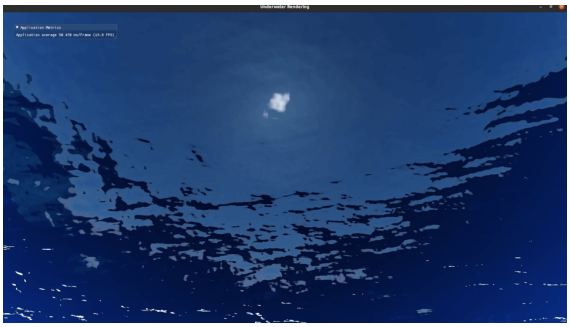
(b) Terrain refracted from above the water surface.



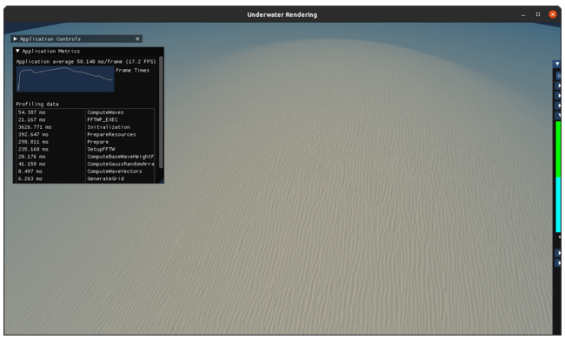
(c) Terrain below the water surface, rendered using the computer vision algorithm model by Sedlazeck et al. [46].



(d) Terrain below the water surface, rendered using the computer vision algorithm model by Sedlazeck et al. [46].

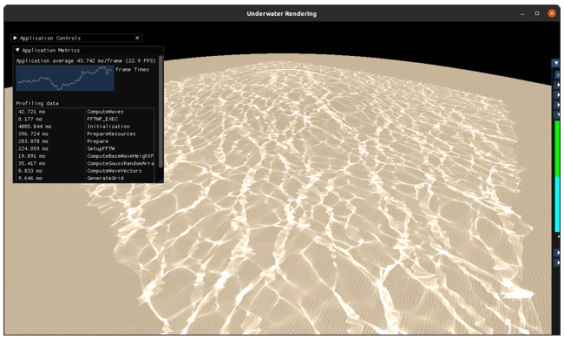


(e) Snell's window below the water surface, rendered using the computer vision algorithm model by Sedlazeck et al. [46].

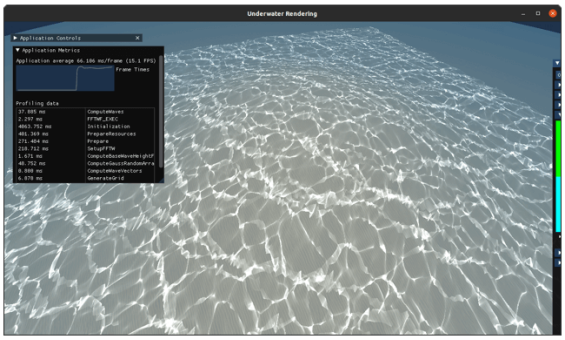


(f) SDF terrain underwater rendered using the designed multiple scattering approach and the proposed bio-optical model.

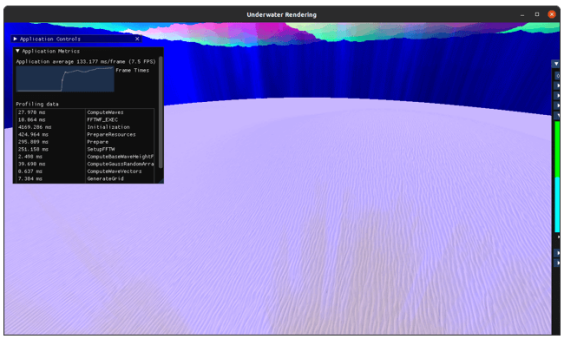
Figure 7.1: Screenshots of experiments with various techniques and methods.



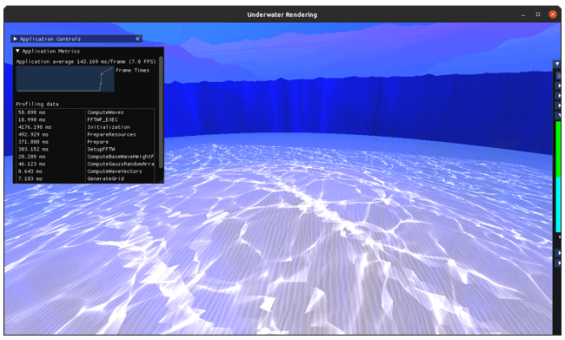
(a) Surface caustics rendered on a terrain represented as a plane. Caustics show blocky appearance and noise.



(b) Surface caustics rendered on the SDF terrain combined with the designed multiple scattering approach.



(c) Volumetric caustics using the beam tracing method intersecting terrain represented by a simple horizontal plane.



(d) The resulting combined scene: water surface, surface caustics on SDF terrain and volumetric beams.

Figure 7.2: Continuation of screenshots of experiments with various techniques and methods.

Chapter 8

Conclusion

The goal of this work is was study, design and implement an application that is able to render underwater environments in real time. Existing solutions for real-time rendering and simulation of oceans were extensively investigated. Furthermore, techniques that simulate underwater caustics were investigated.

A technique was designed to render underwater scenes in high-quality and in real-time. Single scattering and multiple scattering approximations were designed that use ray marching to render water volumes. A physically-based bio-optical model for oceanic waters was designed. This model defines non-uniform vertical distribution of bio-optical parameters. The multiple scattering approximation was specifically designed by exploiting the properties of waters and accounting for the proposed bio-optical model.

The designed technique was not implemented in its entirety. Only a subset of techniques were implemented and experimented with. The implemented techniques run in real-time on relatively modern hardware. Research and investigation of existing techniques and the subsequent designed was prioritized instead.

Bibliography

[1] *Dear ImGui*. 2023 [cit. 2023-23-05]. Available at: https://github.com/ocornut/imgui/.

[2] *FFTW*. 2023 [cit. 2023-23-05]. Available at: http://fftw.org/.

[3] *Shaderc*. 2023 [cit. 2023-23-05]. Available at: https://github.com/google/shaderc.

[4] *Vulkan API*. 2023 [cit. 2023-23-05]. Available at: https://www.vulkan.org/.

[5] Akenine Möller, T., Haines, E., Hoffman, N., Pesce, A., Iwanicki, M. et al. Real-Time Rendering 4th Edition. In:. Boca Raton, FL, USA: A K Peters/CRC Press, 2018, chap. 8, p. 1200. ISBN 978-1-13862-700-0.

[6] Akenine Möller, T., Haines, E., Hoffman, N., Pesce, A., Iwanicki, M. et al. *Real-Time Rendering 4th Edition*. Boca Raton, FL, USA: A K Peters/CRC Press, 2018. 1200 p. ISBN 978-1-13862-700-0.

[7] Akenine Möller, T., Haines, E., Hoffman, N., Pesce, A., Iwanicki, M. et al. Real-Time Rendering 4th Edition. In:. Boca Raton, FL, USA: A K Peters/CRC Press, 2018, chap. 9, p. 1200. ISBN 978-1-13862-700-0.

[8] Akenine Möller, T., Haines, E., Hoffman, N., Pesce, A., Iwanicki, M. et al. Real-Time Rendering 4th Edition. In:. Boca Raton, FL, USA: A K Peters/CRC Press, 2018, chap. 17, p. 1200. ISBN 978-1-13862-700-0.

[9] Akenine Möller, T., Haines, E., Hoffman, N., Pesce, A., Iwanicki, M. et al. Real-Time Rendering 4th Edition. In:. Boca Raton, FL, USA: A K Peters/CRC Press, 2018, chap. 14, p. 1200. ISBN 978-1-13862-700-0.

[10] Aranha, M. Realistic Underwater Visualisation. In:.

[11] Baboud, L. and Décoret, X. Realistic Water Volumes in Real-Time. In: *Eurographics Workshop on Natural Phenomena*. 2006.

[12] Blinn, J. F. Models of Light Reflection for Computer Synthesized Pictures. In: *Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA: Association for Computing Machinery, 1977, p. 192–198. SIGGRAPH '77. DOI: 10.1145/563858.563893. ISBN 9781450373555.

[13] Boss, E. *Ocean Optics Web Book: Water*. 2020 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/optical-constituents-of-the-ocean/water.

[14] Boss, E. *Ocean Optics Web Book: Classification Schemes*. 2021 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/inherent-and-apparent-optical-properties/classification-schemes.

[15] Boss, E. *Ocean Optics Web Book: Colored Dissolved Organic Matter*. 2021 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/optical-constituents-of-the-ocean/colored-dissolved-organic-matter.

[16] Boss, E. *Ocean Optics Web Book: Introduction to Optical Constituents of the Ocean*. 2021 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/optical-constituents-of-the-ocean/introduction-to-optical-constituents-of-the-ocean.

[17] Bowers, D. *Ocean Optics Web Book: Optical Properties of Shelf Seas and Estuaries*. 2021 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/optical-constituents-of-the-ocean/level-2/optical-properties-of-shelf-seas-and-estuaries.

[18] Bricaud, A., Babin, M., Morel, A. and Claustre, H. Variability in the chlorophyll-specific absorption coefficients of natural phytoplankton: Analysis and parameterization. *Journal of Geophysical Research*. 1995, vol. 100, p. 13321–13332.

[19] Cerezo, E. and Serón, F. J. Rendering Natural Waters: Merging Computer Graphics with Physics and Biology. january 2002. DOI: 10.1007/978-1-4471-0103-1_31.

[20] Cetinic, I. *Ocean Optics Web Book: Phytoplankton*. 2021 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/optical-constituents-of-the-ocean/phytoplankton.

[21] Chandrasekhar, S. *Radiative Transfer*. Dover Publications, 1960. Dover Books on Intermediate and Advanced Mathematics. ISBN 9780486605906.

[22] Cheng, H., Chu, J., Zhang, R., Tian, L. and Gui, X. Underwater polarization patterns considering single Rayleigh scattering of water molecules. *International Journal of Remote Sensing*. Taylor & Francis. 2020, vol. 41, no. 13, p. 4947–4962. DOI: 10.1080/01431161.2019.1685725.

[23] Dachsbacher, C. and Stamminger, M. Reflective shadow maps. In: *ACM Symposium on Interactive 3D Graphics and Games*. 2005.

[24] Darles, E., Crespin, B., Ghazanfarpour, D. and Gonzato, J.-C. A Survey of Ocean Simulation and Rendering Techniques in Computer Graphics. *Computer Graphics Forum*. 2011, vol. 30, no. 1, p. 43–60. DOI: 10.1111/j.1467-8659.2010.01828.x.

[25] Dupuy, J. and Bruneton, E. Real-time animation and rendering of ocean whitecaps. In: *SIGGRAPH Asia 2012 Technical Briefs, Singapore, November 28 - December 1, 2012*. ACM, 2012, p. 15:1–15:3. DOI: 10.1145/2407746.2407761.

[26] Ebert, D. S., Musgrave, F. K., Peachey, D., Perlin, K. and Worley, S. *Texturing and Modeling: A Procedural Approach*. 3rdth ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002. ISBN 1558608486.

[27] Ernst, M., Akenine Möller, T. and Jensen, H. W. Interactive Rendering of Caustics Using Interpolated Warped Volumes. In: *Proceedings of Graphics Interface 2005*. Waterloo, CAN: Canadian Human-Computer Communications Society, 2005, p. 87–96. GI '05. ISBN 1568812655.

[28] Fong, J., Wrenninge, M., Kulla, C. D. and Habel, R. Production volume rendering: SIGGRAPH 2017 course. *ACM SIGGRAPH 2017 Courses*. 2017.

[29] Fournier, A. and Reeves, W. T. A simple model of ocean waves. *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. 1986.

[30] Gamper, T. *Ocean Surface Generation and Rendering*. 2018. Master's thesis. Institute of Computer Graphics and Algorithms, Vienna University of Technology. Available at: https://www.cg.tuwien.ac.at/research/publications/2018/GAMPER-2018-OSG/.

[31] GmbH, C. *CryEngine*. 2023 [cit. 2023-23-05]. Available at: https://www.cryengine.com/.

[32] Gordon, H. R. and Morel, A. Remote Assessment of Ocean Color for Interpretation of Satellite Visible Imagery: A Review. In:. 1983.

[33] Gruen, H. Ray-Guided Volumetric Water Caustics in Single Scattering Media with DXR. *Ray Tracing Gems*. 2019.

[34] Gruen, H. Real-Time Ray-Traced One-Bounce Caustics. *GPU Zen 2: Advanced Rendering Techniques*. 2019.

[35] Guardado, J. and Sanchez Crespo, D. Chapter 2. Rendering Water Caustics. In: *GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics*. Pearson Higher Education, 2004.

[36] Gutierrez, D., Muñoz, A., Anson, O. and Serón, F. J. Non-linear volume photon mapping. In: *Eurographics Symposium on Rendering*. 2005.

[37] Gutierrez, D., Serón, F. J., Muñoz, A. and Anson, O. Visualizing Underwater Ocean Optics. *Computer Graphics Forum*. 2008, vol. 27.

[38] Henyey, L. C. and Greenstein, J. L. Diffuse radiation in the Galaxy. *The Astrophysical Journal*. 1940, vol. 93, p. 70–83.

[39] Hillaire, S. Physically-based & Unified Volumetric Rendering in Frostbite. *ACM SIGGRAPH 2015 Physically Based Shading in Theory and Practice course*. july 2015.

[40] Hillaire, S. Physically Based Sky, Atmosphere & Cloud Rendering in Frostbite. *ACM SIGGRAPH 2016 Physically Based Shading in Theory and Practice course*. 2016.

[41] Hu, W., Dong, Z., Ihrke, I., Grosch, T., Yuan, G. et al. Interactive Volume Caustics in Single-Scattering Media. In: *I3D '10: Proceedings of the 2010 symposium on Interactive 3D graphics and games*. ACM, 2010, p. 109–117.

[42] Ioccg, I. Remote Sensing of Ocean Colour in Coastal, and Other Optically-Complex, Waters. In:. January 2000.

[43] Iwasaki, K., Dobash, Y. and Nishita, T. A Volume Rendering Approach for Sea Surfaces Taking into Account Second Order Scattering Using Scattering Maps. In: *Proceedings of the 2003 Eurographics/IEEE TVCG Workshop on Volume Graphics*. New York, NY, USA: Association for Computing Machinery, 2003, p. 129–136. VG '03. DOI: 10.1145/827051.827071. ISBN 1581137451.

[44] Jensen, H. W. Realistic Image Synthesis Using Photon Mapping. In:. 2001.

[45] Jensen, L. S. and Goliáš, R. Deep-Water Animation and Rendering. In:. 2001.

[46] Jordt Sedlazeck, A. and Koch, R. Simulating Deep Sea Underwater Images Using Physical Models for Light Attenuation, Scattering, and Refraction. In: *International Symposium on Vision, Modeling, and Visualization*. January 2011, p. 49–56. DOI: 10.2312/PE/VMV/VMV11/049-056.

[47] Kajiya, J. T. The Rendering Equation. In: *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA: Association for Computing Machinery, 1986, p. 143–150. SIGGRAPH '86. DOI: 10.1145/15922.15902. ISBN 0897911962.

[48] Kim, H. Caustics Using Screen-Space Photon Mapping. *Ray Tracing Gems*. 2019.

[49] Lee, Z.-P., Darecki, M., Darecki, K. L., Davis, C. O., Stramski, D. et al. Diffuse attenuation coefficient of downwelling irradiance: An evaluation of remote sensing methods. *Journal of Geophysical Research: Oceans*. 2005, vol. 110, C2. DOI: https://doi.org/10.1029/2004JC002573.

[50] Liktor, G. and Dachsbacher, C. Real-Time Volumetric Caustics with Projected Light Beams. In:. 2010.

[51] Liktor, G. and Dachsbacher, C. Real-Time Volume Caustics with Adaptive Beam Tracing. In: *Symposium on Interactive 3D Graphics and Games*. New York, NY, USA: Association for Computing Machinery, 2011, p. 47–54. I3D '11. DOI: 10.1145/1944745.1944753. ISBN 9781450305655.

[52] Ma, C., Xu, S.-L., Wang, H., Tian, F. and Chen, G. A real-time photo-realistic rendering algorithm of ocean color based on bio-optical model. *Journal of Ocean University of China*. 2016, vol. 15, p. 996–1006.

[53] McGuire, M. *The Graphics Codex: Ray marching*. 2021 [cit. 2023-23-05]. Available at: https://graphicscodex.courses.nvidia.com/app.html?page=_rn_rayMrch.

[54] Meenrattanakorn, K. and Lambers, M. Real-Time Caustics Using Cascaded Image-Space Photon Tracing. In: Bender, J., Botsch, M. and Keim, D. A., ed. *Vision, Modeling, and Visualization*. The Eurographics Association, 2022. DOI: 10.2312/vmv.20221212. ISBN 978-3-03868-189-2.

[55] Mobley, C. *Ocean Optics Web Book: Reflectances*. 2020 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/inherent-and-apparent-optical-properties/reflectances.

[56] Mobley, C. *Ocean Optics Web Book: Visualizing VSFs*. 2020 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/inherent-and-apparent-optical-properties/visualizing-vsfs.

[57] Mobley, C. *Petzold's Measurements*. 2020 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/scattering/petzolds-measurements.

[58] Mobley, C. *Lambertian BRDFs*. 2021 [cit. 2023-23-05]. Available at: https://oceanopticsbook.info/view/surfaces/lambertian-brdfs.

[59] Mobley, C. *Ocean Optics Web Book: Inherent and Apparent Optical Properties*. 2021 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/inherent-and-apparent-optical-properties/introduction.

[60] Mobley, C. *Ocean Optics Web Book: Inherent Optical Properties*. 2021 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/inherent-and-apparent-optical-properties/inherent-optical-properties.

[61] Mobley, C. *Ocean Optics Web Book: K functions*. 2021 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/inherent-and-apparent-optical-properties/k-functions.

[62] Mobley, C. *Ocean Optics Web Book: Scattering Overview*. 2021 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/scattering/scattering-overview.

[63] Mobley, C. *Ocean Optics Web Book: The Level Sea Surface*. 2021 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/surfaces/the-level-sea-surface.

[64] Mobley, C. *Ocean Optics Web Book: The Physics of Scattering*. 2021 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/scattering/physics-scattering.

[65] Mobley, C. *Ocean Optics Web Book: The Volume Scattering Function (VSF)*. 2021 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/inherent-and-apparent-optical-properties/volume-scattering-function-vsf.

[66] Mobley, C., Boss, E. and Roesler, C. *Ocean Optics Web Book*. 2020 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/.

[67] Monzón, N., Akkaynak, D., Gutiérrez, D. and Muñoz, A. Multiple Scattering Approximation for Real-time Underwater Spectral Rendering. In: Cabiddu, D., Schneider, T., Allegra, D., Catalano, C. E., Cherchi, G. et al., ed. *Smart Tools and Applications in Graphics - Eurographics Italian Chapter Conference*. The Eurographics Association, 2022. DOI: 10.2312/stag.20221264. ISBN 978-3-03868-191-5.

[68] Morel, A. Optical modeling of the upper ocean in relation to its biogenous matter content (case I waters). *Journal of Geophysical Research: Oceans*. 1988, vol. 93, C9, p. 10749–10768. DOI: https://doi.org/10.1029/JC093iC09p10749.

[69] Novák, J. *Efficient Many-Light Rendering of Scenes with Participating Media*. 2014. Dissertation. Karlsruhe Institute of Technology„.

[70] NVIDIA. *Ocean Surface Simulation. NVIDIA Graphics SDK 11 Direct3D*. 2011 [cit. 2023-23-05].

[71] Papadopoulos, C. and Papaioannou, G. Realistic Real-time Underwater Caustics and Godrays. In:. 2009.

[72] Perlin, K. An Image Synthesizer. In:. New York, NY, USA: Association for Computing Machinery, 1985, p. 287–296. SIGGRAPH '85. DOI: 10.1145/325334.325247. ISBN 0897911660.

[73] Petzold, T. J. Volume Scattering Functions for Selected Ocean Waters. In:. 1972.

[74] Pharr, M., Jakob, W. and Humphreys, G. *Physically Based Rendering: From Theory to Implementation*. 3rdth ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2016. ISBN 0128006455.

[75] Pharr, M., Jakob, W. and Humphreys, G. Physically Based Rendering: From Theory to Implementation. In:. 4thth ed. Cambridge, Massachusetts, London, England: The MIT Press, 2023, chap. 11, p. 1312. ISBN 0262048027.

[76] Pope, R. M. and Fry, E. S. Absorption spectrum (380-700 nm) of pure water. II. Integrating cavity measurements. *Applied optics*. 1997, 36 33, p. 8710–23.

[77] Preetham, A. J., Shirley, P. and Smits, B. A Practical Analytic Model for Daylight. In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. USA: ACM Press/Addison-Wesley Publishing Co., 1999, p. 91–100. SIGGRAPH '99. DOI: 10.1145/311535.311545. ISBN 0201485605.

[78] Premoze, S. and Ashikhmin, M. Rendering Natural Waters. *Computer Graphics Forum*. Blackwell Publishers Ltd and the Eurographics Association. 2001. DOI: 10.1111/1467-8659.00548. ISSN 1467-8659.

[79] Quilez, I. *2D distance functions*. 2013 [cit. 2023-23-05]. Available at: https://iquilezles.org/articles/distfunctions2d/.

[80] Quilez, I. *2D distance functions*. 2013 [cit. 2023-23-05]. Available at: https://iquilezles.org/articles/distfunctions/.

[81] Quilez, I. *FBM detail in SDFs*. 2019 [cit. 2023-23-05]. Available at: https://iquilezles.org/articles/fbmsdf/.

[82] Roesler, C. *Ocean Optics Web Book: Absorption by Oceanic Constituents*. 2021 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/absorption/absorption-by-oceanic-constituents.

[83] Roesler, C. *Ocean Optics Web Book: Absorption Overview*. 2021 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/absorption/absorption-overview.

[84] Roesler, C. *Ocean Optics Web Book: Non-Algal Particles (NAP)*. 2021 [cit. 2023-23-05]. Available at: https://www.oceanopticsbook.info/view/optical-constituents-of-the-ocean/non-algal-particles-nap.

[85] Rydahl, B. A VFX ocean toolkit with real time preview. In:. 2009.

[86] Shah, M. A., Kontinnen, J. and Pattanaik, S. N. Caustics Mapping: An Image-Space Technique for Real-Time Caustics. *IEEE Transactions on Visualization and Computer Graphics*. 2007, vol. 13.

[87] Smith, R. and Baker, K. Optical properties of the clearest natural waters (200–800 nm). *Applied optics*. january 1981, vol. 20, p. 177–84. DOI: 10.1364/AO.20.000177.

[88] Solonenko, M. G. and Mobley, C. D. Inherent optical properties of Jerlov water types. *Appl. Opt.* Optica Publishing Group. Jun 2015, vol. 54, no. 17, p. 5392–5401. DOI: 10.1364/AO.54.005392.

[89] Stam, J. Random Caustics: Natural Textures and Wave Theory Revisited. In: *ACM SIGGRAPH 96 Visual Proceedings: The Art and Interdisciplinary Programs of SIGGRAPH '96*. New York, NY, USA: Association for Computing Machinery, 1996, p. 150. SIGGRAPH '96. DOI: 10.1145/253607.253883. ISBN 0897917847.

[90] Tessendorf, J. Simulating Ocean Water. *SIGGRAPH'99 Course Note*. january 2001.

[91] Toledo, E., Vaz, E. and Drews Jr, P. Water Classification Based on Underwater Monocular Image. In:. October 2021, p. 282–287. DOI: 10.1109/LARS/SBR/WRE54079.2021.9605362.

[92] Vivo, P. G. and Lowe, J. *The Book of Shaders*. 2015 [cit. 2023-23-05]. Available at: https://thebookofshaders.com/.

[93] Wenzel, C. Real-Time Atmospheric Effects in Games. In: *ACM SIGGRAPH 2006 Courses*. New York, NY, USA: Association for Computing Machinery, 2006, p. 113–128. SIGGRAPH '06. DOI: 10.1145/1185657.1185831. ISBN 1595933646.

[94] Wronski, B. Volumetric fog: Unified, compute shader based solution to atmospheric scattering. *ACM SIGGRAPH 2014*. july 2014.

[95] Wyman, C. and Nichols, G. Adaptive Caustic Maps Using Deferred Shading. *Computer Graphics Forum*. 2009, vol. 28.

[96] Yang, X. and Ouyang, Y. Real-Time Ray Traced Caustics. *Ray Tracing Gems II*. 2021.