THESE

Pour l'obtention du Grade de

DOCTEUR DE L'UNIVERSITE DE POITIERS

(Faculté des Sciences Fondamentales et Appliquées)
(Diplôme National - Arrêté du 25 mai 2016)

Ecole Doctorale : MIMME

Secteur de Recherche : Traitement du Signal et des images

Présentée par : Ruqin Xiao

....

****

Advanced Simulation of Optical Wireless Channel in Complex Environment

****

Directeur de Thèse : Lilian Aveneau

Co-Directeur de Thèse : Pierre Combeau

****

Soutenue le 20 Septembre 2023

devant la Commission d'Examen

****

JURY

M. Ali Khalighi	MCF-HDR à l'École Centrale de Marseille	Rapporteur
M. Yannis Le Guennec	MCF-HDR à Grenoble-INP	Rapporteur
M. Luc Chassagne	Professeur à l'Université de Versailles Saint Quentin	Examinateur
Mme. Anne Julien-Vergonjanne	Professeur à l'Université de Limoges	Examinateur
M. Lilian Aveneau	Professeur à l'Université de Poitiers	Examinateur
M. Pierre Combeau	MCF-HDR à l'Université de Poitiers	Examinateur

DECLARATION

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is outcome of work done in collaboration with others, except as specified in the text and Acknowledgement. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Ruqin Xiao
October 17, 2023

ACKNOWLEDGEMENTS

I would like to thank all the members of the jury: M.Ali Khalighi, M.Yannis Le Guennec, M.Luc Chassagne, Mme.Anne Julien-Vergonjanne, for reviewing my dissertation.

I would also like to express my deepest gratitude to my supervisors, Professor Lilian Aveneau and associated Professor Pierre Combeau. It is through their unwavering support and guidance that I was able to embark on a three-year Ph.D. journey and explore the fascinating field of optical wireless communication. Their expertise, patience, and constant encouragement have been invaluable to my academic and personal growth.

I am immensely grateful to all my colleagues from the Xlim lab at Université de Poitiers. Their camaraderie, intellectual discussions, shared experiences, and collaborative spirit have enriched both my academic and personal life. In addition, I want to thank Nicolas Prouteau for his help in my study and life.

To my family, I owe a debt of gratitude for their constant support and unconditional love. To my mother, Daixi Xiao, my father, Shitou Wang, and my brother, Haibing Wang, thank you for always being there for me, believing in my abilities, and providing a loving environment that allowed me to focus on my studies.

A special mention goes to my grandmother, uncles, aunts, and cousins in China for their unwavering support and genuine concern for my well-being. I would like to extend a special thank you to my cousin, JinFeng Peng, and her husband, Gongbao Wang, for their invaluable advice and support in my life and study.

Last but not least, I want to acknowledge and thank my friends for their presence and encouragement. In particular, I would like to thank Lu Li for her help in my study and life.

ABSTRACT

Optical Wireless Communication (OWC) is currently knowing a high interest in different complex environments, encompassing participating media like water, fog, cloud, smoke, and biological tissues *etc*. The design of efficient OWC systems needs to test the communication protocols on different scenarios via a lot of simulations. This thesis aims to develop fast and accurate simulation tools, focusing specifically on Underwater Wireless Optical Communication. The most used simulation algorithms are based on Prahl's 1989 method for light propagation through human tissue, relying on Monte Carlo simulation to track the path of a photon in such media. According to this, we propose a mathematical formalization of light propagation in water or other media in the form of an integral, considering only scattering. We then propose 2 new algorithms, namely Xiao 1 and Xiao 2, with improved performance in terms of reduced variance. The former removes the importance sampling on the last propagating distance. On this basis, the latter directly samples a solid angle enclosing the receiver with importance. Considering both scatterings and reflections, two extended algorithms are devised by generalizing the path space, namely Xiao 1-R and Xiao 2-R. The simulation results show that the 4 new solutions exhibit better performance in terms of accuracy and efficiency, as compared to Prahl's algorithm with or without reflections, especially for Xiao 2 and Xiao 2-R. They also provide good accuracy in the application of underwater localization.

Keywords: Communication channels, Communication systems, Monte Carlo methods, Optical communication, Optical propagation, Optical propagation in absorbing media, Optical propagation in dispersive media, Simulation, Underwater communication, Underwater optical communication, Underwater optical propagation, Underwater localization.

RÉSUMÉ

Les communications optiques sans fil (OWC) suscitent actuellement un grand intérêt dans différents environnements complexes, englobant des milieux participants tels que l'eau, le brouillard, les nuages, la fumée, les tissus biologiques, etc. La conception de systèmes OWC efficaces nécessite de tester les protocoles de communication sur différents scénarios via un grand nombre de simulations. Cette thèse vise à développer des outils de simulation rapides et précis, en se concentrant spécifiquement sur la communication optique sous-marine sans fil. Les algorithmes de simulation les plus utilisés sont basés sur la méthode de Prahl de 1989 pour la propagation de la lumière à travers les tissus humains, s'appuyant sur la simulation de Monte Carlo pour suivre le chemin d'un photon dans de tels milieux. Dans cette optique, nous proposons une formalisation mathématique de la propagation de la lumière dans l'eau ou d'autres milieux sous la forme d'une intégrale, en ne tenant compte que de la diffusion. Nous proposons ensuite deux nouveaux algorithmes, à savoir Xiao 1 et Xiao 2, dont les performances sont améliorées en termes de réduction de la variance. Le premier supprime l'échantillonnage d'importance sur la dernière distance de propagation. Sur cette base, le second échantillonne directement un angle solide entourant le récepteur avec importance. En considérant à la fois les diffusions et les réflexions, deux algorithmes étendus sont conçus en généralisant l'espace de chemin, à savoir Xiao 1-R et Xiao 2-R. Les résultats des simulations montrent que les 4 nouvelles solutions présentent de meilleures performances en termes de précision et d'efficacité, par rapport à l'algorithme de Prahl avec ou sans réflexions, en particulier pour Xiao 2 et Xiao 2-R. Elles offrent également une bonne précision dans l'application de la localisation sous-marine.

Mots-clés: Canaux de communication, Systèmes de communication, Méthodes de Monte Carlo, Communication optique, Propagation optique, Propagation optique dans les milieux absorbants, Propagation optique dans les milieux dispersifs, Simulation, Communication sous-marine, Communication optique sous-marine, Propagation optique sous-marine, Localisation sous-marine.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

IS	Importance Sampling
BL	Beerlambert
BRDF	Bidirectional Reflectance Distribution Function
CDF	Cumulative Density Function
DB	Database
FOV	Field Of View
MC	Monte Carlo
MCRS	Monte Carlo Ray Shooting
MCRT	Monte Carlo Ray Tracing
MDF	Marginal Density Function
Meas	Measurement
NEE	Next Event Estimation
OWC	Optical Wireless Communication
PDF	Probability Density Function
RTE	Radiative Transfer Equation
ULP	Underwater Light Propagation
UVLP	Underwater Visible Light Positioning
UWC	Underwater Wireless Communication
UWOC	Underwater Wireless Optical Communication
VSF	Volume Scattering Function

LIST OF NOTATIONS

Φ_e	Radiated engery expressed in $[J]$
Φ	Radiant power or flux, expressed in $[W = J \cdot s^{-1}]$
E	Irradiance expressed in $[W \cdot m^{-2}]$
$\Omega_{S/2}$	Half direction integration space
L	Radiance expressed in $[W \cdot m^{-2}] \cdot sr^{-1}$
$\mathscr{U}(\overrightarrow{\omega_{T_x}}, \overrightarrow{n_{T_x}}, m)$	Lambertian radiation pattern
m	Lambertian order
$\overrightarrow{n_{T_x}}$	Unit normal vector of transmitter
$\overrightarrow{\omega_{T_x}}$	Unit emitting direction from transmitter
$\mathscr{U}_s(\overrightarrow{\omega_{T_x}}, \overrightarrow{n_{T_x}}, m)$	Radiation pattern of disc transmitter
$\Phi_{1/2}$	Half power angle of the disc transmitter
A_{T_x}	Transmitter's surface
$\mathscr{U}_l(\overrightarrow{\omega_{T_x}})$	Radiation pattern of lineic transmitter
$\theta_{\max}$	Maximum radiation angle of lineic transmitter
$\mathscr{A}(\overrightarrow{\omega_{R_x}}, \overrightarrow{n_{R_x}})$	The acceptance of the disc receiver
$\overrightarrow{n_{R_x}}$	Unit normal vector of receiver
θ_{R_x}	The receiveing angle at the disc receiver
A_{R_x}	Receiver's surface
$R(x, \vec{\omega}', \vec{\omega})$	BRDF
ρ	The surface albedo
σ_s	Scattering coefficient (m^{-1})
σ_a	Absorption coefficient (m^{-1})
Ω_S	Full direction integration space, equivalent to unit sphere points
$\beta_{HG}(\theta, g)$	Henyey-Greenstein volume scattering function
g	HG asymmetry factor
$\beta_{TTHG}(\theta, g_F, g_B)$	Two Term Henyey-Greenstein volume scattering function
g_F	Factor for forward-directed scattering
g_B	Factor for backward-directed scattering

α	The weight of the forward-directed HG volume scattering function
$\beta_R(\theta)$	Rayleigh volume scattering function
σ_t	Extinction coefficient (m^{-1})
$T(\Delta s)$	Transmittance: ratio of outgoing on ingoing radiance in a given transparent material or participating medium of length Δs
$C_g(T_x, R_x)$	Gain of the UWOC channel
$h(T_x, R_x, t)$	Channel impulse response for the link (s^{-1})
$P_n(T_x, R_x, t)$	Flux received at R_x coming from T_x after n scattering events (W/s)
$\widehat{P_n}(T_x, R_x, t)$	Monte Carlo estimator of $P_n(T_x, R_x, t)$ (W/s)
T_x	Transceiver location
R_x	Receiver location
x_i	Point location in the participating medium
x_0	Point location on the transmitter
x_r	Point location on the receiver
$\overline{y}$	Propagation path generally with n scattering events $(x_0, x_1, \ldots, x_n, x_r)$
τ	Propagation time of path $\overline{y}$ (s)
$\overrightarrow{\omega_i}$	Unit direction from point x_i to x_{i+1}
$L_e(x_0, \overrightarrow{\omega_0}, t)$	Radiated radiance from point x_0 of transmitter and per solid angle $\overrightarrow{\omega_0}$ (W/st/m^2/s)
l_i	Distance, between points x_i and x_{i+1} (m)
f_i^n	Flux received at R_x coming from T_x after n scattering events, version $i \in [0 \ldots 4]$ (W/st/m^2/s)
$p(X)$	Probability density function of the random variable X
$\overline{y_k}$	Random variable corresponding to a propagation path $\overline{y}$
τ_k	Propagation time of path $\overline{y_k}$ (s)
$\delta(x)$	Dirac delta function, equals to 0 when $x \neq 0$ and with integral 1
$T(l)$	Transmittance: ratio of outgoing on ingoing radiance in a given transparent material or participating medium of length l
$\beta(\overrightarrow{\omega_i}, \overrightarrow{\omega_s})$	Volume Scattering Function

$V_{is}(x,y)$	Visibility function, equals to 1 when points x and y are mutually visible, otherwise 0
V	Volume integration space
A	Surface integration space
L	Length integration space ($L = \mathbb{R}^{+}$)
Ω_{R_x}	Direction integration space reduced to R_x
NF	VSF normalization factor
$NF_{(HG)}$	Henyey-Greenstein VSF normalization factor
$CNF_{(HG)}$	Cumulative Henyey-Greenstein VSF normalization factor
$NF_{(TTHG)}$	Two-term Henyey-Greenstein VSF normalization factor
$NF_{(Rayleigh)}$	Rayleigh VSF normalization factor

Chapter 1

Introduction

1.1 Context

Light, with its captivating and intricate nature, has fascinated humans for centuries. From the mesmerizing colours of a rainbow to the stunning hues of a sunset, light permeates our world, shaping our visual experience and providing a means of communication and exploration. In the modern era, Optical Wireless Communication (OWC) is present in many applications, ranging from indoor wireless networks [3–8] to vehicular communication [9, 10], underwater communication [11–13] to aerospace communication, and even satellite communication [14–16]. Conventional wireless communication has relied on Radio Frequency (RF) technique, which offers widespread coverage and the ability to penetrate obstacles for long-distance connectivity. Meanwhile, it also faces issues like electromagnetic interference, limited bandwidth owing to finite RF spectrum and potential health risks, *etc*. As a promising complementary solution to this technique [5–7], OWC provides unregulated bandwidth, enhanced security by avoiding penetration through obstacles, and low latency. On the other hand, OWC suffers some limitations: the limited coverage, mobility, requirements of Line-Of-Sight (LOS) link for high data rates, *etc*.

The implementation of an OWC system requires a study of the light propagation channel behaviour, like obstacles influence. This study can be done based on massive measurement campaigns, directly conducted in the propagation environments. Nevertheless, they can be extremely costly, hazardous and challenging, as in the case of underwater channels. Thus, it becomes necessary to design the efficient OWC system in such environments, incorporating accurate modelling of the behaviour of the propagation channel. Consequently, fast and accurate simulation tools are needed to test the developed communication protocols on different scenarios.

In many real-world scenarios, the presence of participating media, such as cloud, skin, smoke, water, can affect the propagation of light and so the performance of OWC channel [2]. This kind of channel is characterized by different physical phenomena, inherent to "participating media": light absorption, due to particles that typically absorb and convert the light flux into heat; light scattering, due to particles that deflect the luminous flux in other directions, causing the dispersion of the light. Besides, surface reflections may also impact this channel due to the presence of obstacles.

These phenomena imply that the OWC channel in such complex environments can become very difficult to simulate. In the previous works, most are based on Monte Carlo Simulations, where photons are tracked in the scene and counted when they reach the receivers. The modern algorithm widely used for this purpose was first proposed by Prahl in 1989 [1] for simulating light propagation in layered tissue. This method is presented in an organizational structure, which elucidates its subsequent adoption as a starting point for further research. It requires a large amount of photons to achieve good results with low error. In fact, there exist some optimization techniques to reduce variance with Monte Carlo integration. Unfortunately, owing to the lack of a well-defined mathematical foundation using integrals, it is extremely difficult to suggest improvements for this old algorithm. To address this issue, it needs to develop a mathematical approach capable of representing the light propagation channel as an integration problem.

1.2 Objectives and contributions

The objective of this thesis is to realistically and efficiently simulate light propagation in some environments containing participating media. The adopted approach is to consider configurations of increasing complexity, *i.e.* first a participating medium of infinite size containing no surface, then the general case of a participating medium containing surfaces. Staring from Prahl's algorithm, this thesis aims to improving this algorithm by applying variance reduction techniques.

In the context of this thesis, the first important task was to establish a robust mathematical framework for modelling light propagation channel as an integration problem in the presence of participating media and considering only scatterings. From this new formalization, the link with Prahl's Monte Carlo algorithms used in previous works was explained, showing the correctness of this formalism. Building on this foundation, novel algorithms were proposed: the first one by eliminating unnecessary importance sampling used in Prahl's algorithm; and the second one by explicitly sampling the receiver using importance sampling. These two new algorithms were evaluated across various scenarios, studying their variance reduction capabilities and computation times.

Furthermore, the path space was generalized to encompass both reflections and scatterings, leading to a more general formalization. Based on this new formalization, two algorithms were proposed as the generalization to the two previous ones. They were evaluated through a series of simulations considering various scenarios, studying their variance reduction capabilities and computation times.

Lastly, as an application, this thesis demonstrates the added value provided by these new algorithms in the framework of underwater localization problem. They exhibited better localization accuracy comparing to the simple Beer-Lambert model.

1.3 Organization of this dissertation

The remaining part of this dissertation is structured as follows:

Chapter 2 provides the fundamental principles underlying the propagation of light, including establishing the basic terminologies of light properties, providing the sensor models in terms of transmitter and receiver, explaining the behaviour of light as it interacts with surfaces and travers through media, and exploring the necessary background for underwater optical wireless communication in terms of applications, models and implementation.

Chapter 3 first proposes a mathematical formalization of light propagation in water or any participating media, allowing to retrieve well-known Prahl's algorithm [1]. Based on this formalization, a new algorithm (Xiao 1) is developed by removing importance sampling on the last propagation distance. This chapter finishes by a performance evaluation based on simulation which highlights the superior efficiency of Xiao 1 algorithm compared to Prahl's one.

Chapter 4 proposes a new light propagation simulation algorithm (Xiao 2) based on the formalization proposed in the previous chapter. It first provides a detailed explanation of this algorithm and its underlying principles and model. Then it applies this model to 3 different VSFs including Henyey-Greenstein, Two-Term Henyey-Greenstein and Rayleigh. This chapter ends with some simulations that show the better performances of Xiao 2 algorithm with respect to Xiao 1 and Prahl's ones.

Chapter 5 proposes to generalize the path space considering both scatterings and reflections, to offer an explicit expression for the UWOC channel as an integral over the paths space. Then it develops two algorithms based on this mathematical formalization. One uses the same variance reduction technique used in Xiao 1, leading to Xiao 1-R algorithm. The second does the same with Xiao 2 algorithm, leading to Xiao 2-R algorithm. The performance comparisons and conclusions are finally provided.

Chapter 6 proposes to apply the MC model corresponding to Xiao 2 algorithm, to underwater localization algorithm which is UVLP algorithm [102], which was initially based on Beer-Lambert model, i.e. a very simplified propagation model. The performance comparison to Beer-Lambert model are conducted then, showing the importance to consider a realistic channel model to ensure reliable localization.

Chapter 7 provides an overall conclusion and perspective of the thesis, summarizing the key findings and contributions.

Chapter 2

Foundations of light propagation

The study of light and its properties forms the basis of numerous scientific and technological advancements, particularly in fields of communication systems. Understanding how light propagates through various complex environments is of paramount importance for the development of efficient and reliable optical wireless communication technologies. In this chapter, we embark on a journey to explore the fundamental principles underlying the propagation of light. We begin by establishing essential terminologies, which serve as key concepts in comprehending the physical properties of light, including flux, irradiance, and radiance. Then we shift our focus to the sensor models employed in light-based communication systems, revolving around two key components: transmitter serving as light source and receiver functioning as light detector. Moving forward, we venture into the behaviour of light as it interacts with surfaces, and traverses through participating media, such as water, fog, smoke, or biological tissues. Lastly, we explore the necessary background for Underwater Optical Wireless Communication (UWOC) in terms of its applications, models, and implementation, since we consider this framework as an application of the works developed in this dissertation.

2.1 Radiometric quantities

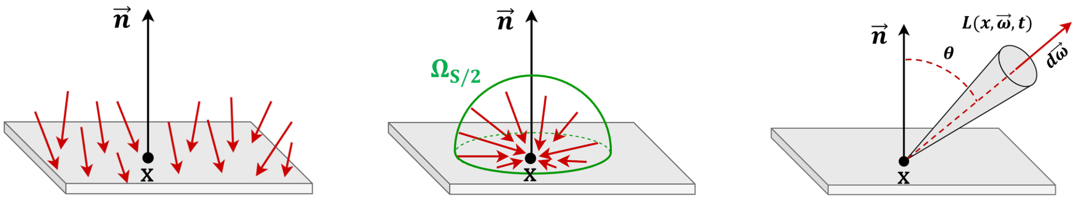
Radiometric quantities [2, 17], such as flux, irradiance and radiance, are fundamental concepts in the field of radiometry dealing with the measurement and characterization of electromagnetic radiation. They provide a way to measure and describe the behaviour of radiant energy in terms of light interactions with surfaces and medium. This section aims to establish a common terminology for the physical quantities and units employed in global illumination algorithms, according to the basic radiometric quantities illustrated in Figure 2.1. In reality, all of these radiometric quantities depend on the wavelength of light [2]. For the purposes of this study, they are implicitly defined for a given wavelength λ.

2.1.1 Flux

In radiometry, the most fundamental quantity is radiant power, or flux, denoted as Φ. It measures the amount of light that hits a surface over a finite area from all directions per unit time [2], as seen in Figure 2.1a. Notice that the term $\vec{n}$ in this figure is a unit vector orthogonal to the interaction surface at a single point x. For a given amount of energy Φ_e at a time duration t, the flux Φ can be computed as:

$$\Phi(t) = \frac{\mathrm{d}\Phi_e}{\mathrm{d}t}, \tag{2.1}$$

As Φ_e is expressed in Joules $[J]$, the unit of $\Phi(t)$ is watts $\left[W = J \cdot s^{-1}\right]$.



(a) Flux (b) Irradiance (c) Radiance

Figure 2.1: Radiometric quantities

2.1.2 Irradiance

A correlated measure, called irradiance (denoted as E), captures the integration over the entire hemisphere $\Omega_{S/2}$ of the light arriving at the surface $\mathrm{d}A$ centred onto a point x, as illustrated in Figure 2.1b. Essentially, it measures the amount of incident power striking a unit area per unit time, and is expressed in terms of flux as:

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

where the term $\mathrm{d}A$ is expressed in $[m^2]$. Consequently, the irradiance $E(x,t)$ is measured in the unit of $[W \cdot m^{-2}]$. It is worth noting that the relationship between flux and irradiance can be also expressed in the form of integration as follows:

$$\Phi(t) = \int_A E(x,t)\mathrm{d}A(x) \tag{2.3}$$

2.1.3 Radiance

Another very important radiometric quantity is radiance (denoted as L). It measures the amount of light arriving at the surface dA centred onto a point x from a differential solid angle $\mathrm{d}\overrightarrow{\omega}$, as seen in Figure 2.1c. The angle θ in this figure represents the angle between the surface normal $\overrightarrow{n}$ and the direction $\overrightarrow{\omega}$. Moreover, $\mathrm{d}A_{\perp}$ is the differential area corresponding to the projection of dA onto direction $\overrightarrow{\omega}$, expressed as follows:

$$\mathrm{d}A_{\perp} = \|\overrightarrow{\omega} \cdot \overrightarrow{n}\|\mathrm{d}A. \tag{2.4}$$

As such, the radiance can be expressed in terms of flux $\Phi(t)$ as:

$$L(x, \overrightarrow{\omega}, t) = \frac{\mathrm{d}^2\Phi(t)}{\mathrm{d}\overrightarrow{\omega}\mathrm{d}A_{\perp}} \tag{2.5}$$

Notice that the unit of solid angle is the steradian with its symbol sr, *i.e.* $\Omega_{S/2}$ with $2\pi[sr]$, and a full sphere Ω_S with $4\pi[sr]$. So the unit of $L(x, \overrightarrow{\omega}, t)$ is $[W \cdot m^{-2} \cdot sr^{-1}]$. In practice, the measurement of radiance is typically performed on an actual surface, rather than a surface perpendicular to the direction of light propagation, as follows:

$$L(x, \overrightarrow{\omega}, t) = \frac{\mathrm{d}^2\Phi(t)}{\|\overrightarrow{\omega} \cdot \overrightarrow{n}\|\mathrm{d}\overrightarrow{\omega}\mathrm{d}A} \tag{2.6}$$

According to this equation and Equation (2.2), this radiance can be also expressed in terms of irradiance $E(x,t)$ as:

$$L(x, \overrightarrow{\omega}, t) = \frac{\mathrm{d}^2 E(x,t)}{\|\overrightarrow{\omega} \cdot \overrightarrow{n}\|\mathrm{d}\overrightarrow{\omega}} \tag{2.7}$$

On this basis, we can deduce the following useful expression which explains the relationship between the irradiance and radiance in the integral form:

$$E(x,t) = \int_{\Omega_{S/2}} L(x, \overrightarrow{\omega}, t)\|\overrightarrow{\omega} \cdot \overrightarrow{n}\|\mathrm{d}\overrightarrow{\omega} \tag{2.8}$$

2.2 Sensor models

The sensors models play a vital role in the performance and behaviour of the light-based communication system. Their overall performance are determined by the basic crucial factors, including the geometry, area, and active surface of sensors. The geometry of a sensor refers to its physical shape, which can vary widely depending on the application for which it is intended. Common shapes include disc, rectangular, cylindrical, and spherical configurations, each with its unique advantages and drawbacks. The sensor's area represents the physical extent of its sensing capabilities, while the active surface is the part of the sensor that interacts with the environment and converts external stimuli into measurable signals. A larger area typically implies a greater sensing ability, but it also imposes challenges in terms of manufacturing, power consumption, and integration with other components. This section explores several sensor models in terms of the transmitter and the receiver.

2.2.1 Transmitters

In light-based communication systems, transmitters act as optical sources that emit photons into a scene. The angular distribution of emitted light depends on the transmitter's radiation pattern, specifying intensity variations with direction.

The most widely used model for simulating the optical wireless channel is the generalized Lambertian model [18], which simulates diffuse and directional emitters with uniaxial symmetry around the normal. It is characterized by the order m of its emission diagram, which controls its directivity, and is given by:

$$\mathscr{U}(\overrightarrow{\omega_{T_x}}, \overrightarrow{n_{T_x}}, m) = \frac{m+1}{2\pi} \|\overrightarrow{\omega_{T_x}} \cdot \overrightarrow{n_{T_x}}\|^m, \tag{2.9}$$

where:

- The order m defines the directivity of the radiation pattern, and depends on the transmitter's half power angle $\Phi_{1/2}$:

  $$m = \frac{-\ln 2}{\ln(\cos \Phi_{1/2})}. \tag{2.10}$$

  For instance, $m = 1$ implies the standard Lambertian model, corresponding to a half power angle of 60°, as seen in Figure 2.2.
- $\overrightarrow{n_{T_x}}$ is the normal of the transmitter, as seen in Figure 2.2.
- $\overrightarrow{\omega_{T_x}}$ is the unit emitting direction from the transmitter, as seen in Figure 2.2.

The emitter is often considered as punctual in the literature, *i.e.* without area. We propose more realistic models taking into account the physical shape and area of the sensor like (disc, rectangular and linear wire), in which the radiation pattern is normalized by the area (or length in case of wire). Figure 2.3 shows one such Lambertian surface emitter example: the disc-shaped emitter. Given the area A_{T_x} of

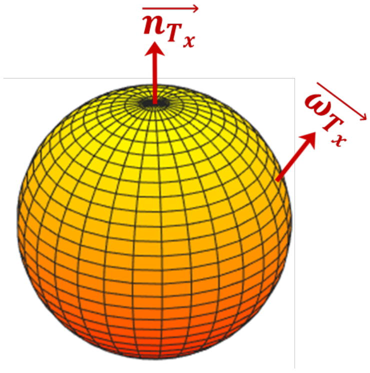


Figure 2.2: Lambertian model (m=1)

such an emitter, its angular distribution of the emitted radiance in the direction $\overrightarrow{\omega_{T_x}}$ is as follows:

$$\mathscr{U}_s(\overrightarrow{\omega_{T_x}}, \overrightarrow{n_{T_x}}, m) = \frac{m+1}{2\pi A_{T_x}} \| \overrightarrow{\omega_{T_x}} \cdot \overrightarrow{n_{T_x}} \|^m, \tag{2.11}$$

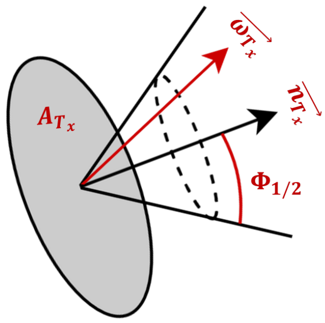


Figure 2.3: General characteristics of disc transmitter

The lineic emitter, on the other hand, has a linear shape corresponding to a wire, and is designed to emit radiation along its length. Different from the surface transmitter, its radiation is independent of the radiated direction inside the beam. Given the beam width w_0 and the maximum radiation angle $\theta_{\max}$, its corresponding angular distribution follows as:

$$\mathscr{U}_l(\overrightarrow{\omega_{T_x}}) = \frac{1}{2\pi(1-\cos\theta_{\max})w_0}. \tag{2.12}$$

2.2.2 Receivers

Receivers, also called optical detectors or photodetectors, are essential components of any communication system, responsible for capturing transmitted light [18]. The acceptance model of a receiver describes how the receiver interacts with incoming signals, helping predict the amount of signal captured by the receiver, as well as its sensitivity and directionality. It is usually defined by considering the receiver's shape, its surface area, *etc*.

The widely used receiver type is the disc receiver linked to a circular geometry, as seen in Figure 2.4. It consists of a flat, circular surface designed to capture incoming signals from a specific direction. The receiver's key characteristics include its Field of View (FOV) and its surface area A_{R_x}. A wider FOV allows the receiver to capture signals from a broader range of directions, increasing its sensitivity to incoming light. Similarly, a larger surface area provides a greater physical space for light collection, enabling the receiver to capture more light. Its acceptance with a strict FOV is given by:

$$\mathscr{A}\left(\overrightarrow{\omega_{R_x}}, \overrightarrow{n_{R_x}}\right) = A_{R_x} ||\overrightarrow{\omega_{R_x}} \cdot \overrightarrow{n_{R_x}}|| \mathrm{rect}\left(\frac{\theta_{R_x}}{\mathrm{FOV}}\right), \tag{2.13}$$

where $\overrightarrow{\omega_{R_x}}$ is the unit direction incoming to the receiver, $\overrightarrow{n_{R_x}}$ is the normal to the receiver's surface, $\theta_{R_x} = \arccos\left(||\overrightarrow{\omega_{R_x}} \cdot \overrightarrow{n_{R_x}}||\right)$ is the receiving angle at the receiver, and $\mathrm{rect}\left(\frac{\theta_{R_x}}{\mathrm{FOV}}\right)$ is the well-known rectangular function, being 1 if $\theta_{R_x} \leq FOV$, and 0 otherwise.

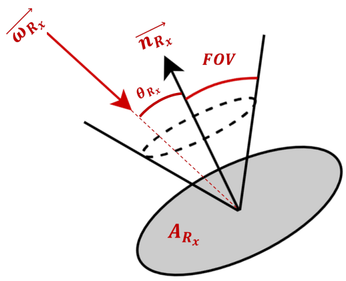


Figure 2.4: General characteristics of disc receiver

2.3 Surface reflection

This section delves into the intricate behaviour of light when it interacts with a surface.

2.3.1 BRDF

The light reflection properties of a given surface can be modelled by the Bidirectional Reflectance Distribution Function (BRDF) [19]. It expresses the relationship between the radiance L_o reflected at a point x along the outgoing direction $\overrightarrow{\omega}$, and the radiance L_i incoming from the direction $\overrightarrow{\omega}'$, as shown in Figure 2.5a. The reflectance distribution at the interaction point x is given by:

$$R(x, \overrightarrow{\omega}', \overrightarrow{\omega}) = \frac{\mathrm{d}L_o(x, \overrightarrow{\omega})}{L_i(x, \overrightarrow{\omega}')\|\overrightarrow{\omega}' \cdot \overrightarrow{n}\|\mathrm{d}\overrightarrow{\omega}'} \tag{2.14}$$

where $\overrightarrow{n}$ is the surface normal at x. Notably, a surface cannot reflect more light than it receives owing to energy conservation [2], leading to the following expression:

$$\int_{\Omega_{S/2}} R(x, \overrightarrow{\omega}', \overrightarrow{\omega})\|\overrightarrow{\omega} \cdot \overrightarrow{n}\|\mathrm{d}\overrightarrow{\omega} \leq 1 \tag{2.15}$$

The BRDF models depend on the material of surface, *i.e.* from ideal diffuse material as depicted in Figure 2.5b to perfectly specular material as seen in Figure 2.5d. In this section, three widely used BRDFs are discussed, including Lambertian BRDF, Blinn-Phong BRDF and specular BRDF.

Lambertian BRDF

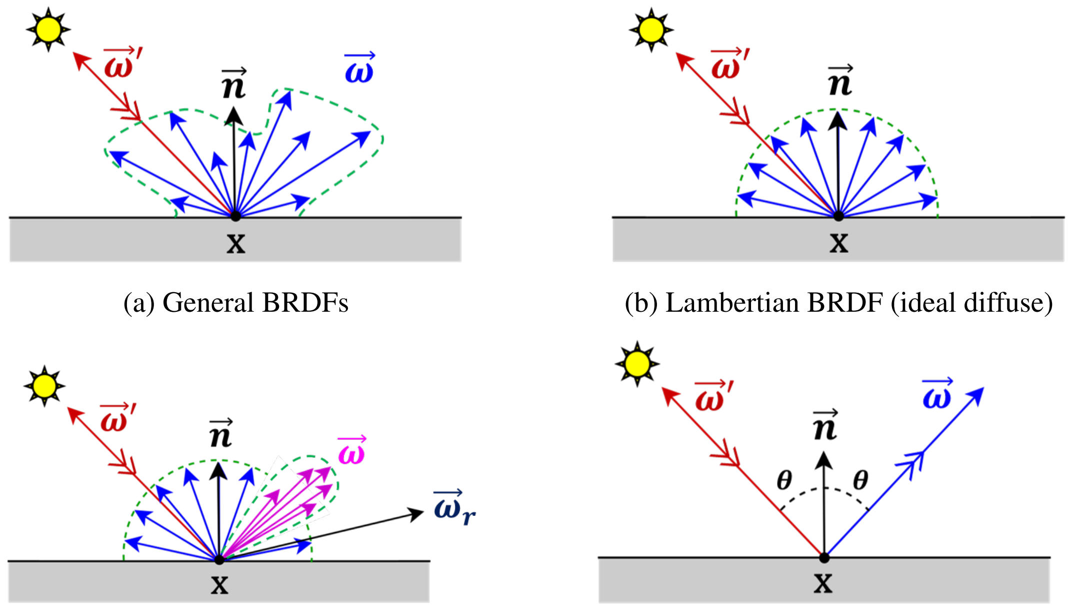
The Lambertian model was first introduced by Johann Heinrich Lambert in 1760 [19]. It represents a perfectly diffuse surface, where light is reflected uniformly in all directions. This model is simple and widely used for matte surfaces. Its expression is given by:

$$R(x, \overrightarrow{\omega}', \overrightarrow{\omega}) = \frac{\rho}{\pi}, \tag{2.16}$$

where ρ is the surface albedo (diffuse reflectance) of the material, ranging from 0 to 1.

Blinn-Phong BRDF

Blinn-Phong model was first introduced in 1977 [20], by using a linear interpolation between a Lambertian part and a directive one, as depicted in Figure 2.5c. It is widely



(a) General BRDFs

(b) Lambertian BRDF (ideal diffuse)

(c) Blinn-Phong BRDF

(d) Specular BRDF

Figure 2.5: BRDF models

used in real-time rendering due to its simplicity and efficiency. Its mathematical expression is given by:

$$R(x, \overrightarrow{\omega}', \overrightarrow{\omega}) = \rho \left(\frac{k_s}{\pi} + (1 - k_s) D(x, \overrightarrow{\omega}', \overrightarrow{\omega}) \right), \tag{2.17}$$

where k_s is the linear interpolation parameter, $D(x, \overrightarrow{\omega}', \overrightarrow{\omega})$ is the directive part corresponding to a perfect specular reflection on a virtual surface whose normal $\overrightarrow{\omega_r}$ following as:

$$\overrightarrow{\omega_r} = \frac{\overrightarrow{\omega} - \overrightarrow{\omega}'}{\|\overrightarrow{\omega} - \overrightarrow{\omega}'\|}$$

According to the Blinn distribution [20], $D(x, \overrightarrow{\omega}', \overrightarrow{\omega})$ is given by:

$$D(x, \overrightarrow{\omega}', \overrightarrow{\omega}) = \frac{\sigma + 2}{2\pi} |\overrightarrow{\omega_r} \cdot \overrightarrow{n}|^{\sigma}, \tag{2.18}$$

where $\sigma \geq 0$ defines the BRDF directivity.

Specular BRDF

The Specular BRDF represents a perfectly smooth and mirror-like surface, with light reflected only in the mirror direction [2, 21], as seen in Figure 2.5d. The outgoing direction $\overrightarrow{\omega}$ is symmetric to the incoming direction $\overrightarrow{\omega}'$ with respect to the surface normal $\overrightarrow{n}$.

The corresponding mathematical model is as follows:

$$R(x, \overrightarrow{\omega}', \overrightarrow{\omega}) = F_r\left(\overrightarrow{\omega}'\right) \frac{\delta\left(\overrightarrow{\omega} - \overrightarrow{\omega}'\right)}{|\cos\theta|} \tag{2.19}$$

where δ is the delta-Dirac function, and the function F_r is as follows:

$$F_r\left(\overrightarrow{\omega}'\right) = \frac{1}{2}\left(r_{\parallel}^2 + r_{\perp}^2\right), \tag{2.20}$$

where $r_{\parallel}$ and $r_{\perp}$ are the conventional Fresnel coefficients [28].

2.3.2 The rendering equation

Computing the lighting at a surface requires to solve the Rendering Equation (RE), which was first introduced by James in 1986 [22]. It mathematically expresses the outgoing radiance $L_o(x, \overrightarrow{\omega})$ of any point in a scene as the sum of emitted radiance $L_e(x, \overrightarrow{\omega})$ at that point and the reflected radiance L_r:

$$\begin{aligned} L_o(x, \overrightarrow{\omega}) &= L_e(x, \overrightarrow{\omega}) + L_r(x, \overrightarrow{\omega}) \\ &= L_e(x, \overrightarrow{\omega}) + \int\limits_{\Omega_{S/2}} R(x, \overrightarrow{\omega}', \overrightarrow{\omega}) L_i(x, \overrightarrow{\omega}') \left(\overrightarrow{\omega}' \cdot \overrightarrow{n_x}\right) \mathrm{d}\overrightarrow{\omega}', \end{aligned} \tag{2.21}$$

where L_i is the incident radiance at x from direction $\overrightarrow{\omega}'$. The equation is recursive in nature because the reflected radiance depends on L_i, determined by solving the same equation for all the other points in the scene seen from x. Solving the rendering equation accurately for every point in a scene is computationally expensive and requires approximations and numerical techniques.

In computer graphics, there exists some algorithms used to solve the rendering equation either fully or partially [2], generally categorized into two types: finite element methods and Monte Carlo Ray Tracing (MCRT) methods. In the computation of indirect illumination, the former first discretize the scene geometry into a collection of small surface patches, and then express the rendering equation as a system of linear equations describing the light exchange between these patches in the scene. Unfortunately, this solution cannot efficiently handle scenes with complex geometry and BRDFs owing to its corresponding expensive simulation cost. On the other hand, the latter relies on the general Monte Carlo solution of an integral problem, which consists in successively sampling each integration domain according to a random process. For instance, Monte Carlo Ray Shooting (MCRS) [19, 23], begins by generating a random path from the transmitter using sampling, and then performs successive reflections. At each ray-surface intersection point, a new direction is randomly chosen to recursively simulate an additional reflection. In this dissertation, we adopt this solution to deal with the light interaction with a surface.

2.4 Light propagation in participating media

The study of light propagation in either inhomogeneous or homogeneous media, such as water, smoke, fog, clouds, and biological tissue, has been a research subject since more than a century. In this context, Schuster conducted research on the absorption and scattering of light within such media, primarily focusing on half-forward and backward scattering [24]. Later, Preisendorfer contributed by introducing mathematical principles to elucidate the process of light scattering [25]. During the 1960s, Chandrasekhar and Preisendorfer authored two influential reference books on radiative transfer [26, 27]. These books serve as fundamental resources for contemporary works in this field, although the former may pose some difficulty of reading, with the emphasis on computational techniques. The latter recaps all the main physical quantities allowing to express the propagation of light in participating media.

On this basis, this section delves into the fundamental physical phenomena that occur when light interacts with such mediums, encompassing absorption, scattering, and transmittance. Furthermore, it explores the Radiative Transfer Equation (RTE) according to [27], which serves as a mathematical framework for calculating changes in radiance resulting from the interaction of light with the medium. Finally, a summary of prior research on light propagation in participating media is presented, consolidating the collective knowledge gained from previous investigations.

2.4.1 Physical phenomena and their models

To better describe the light interaction with participating media, let's model the participating media as a collection of randomly positioned microscopic particles according to the assumption made in [2], as seen in Figure 2.6. These particles are considered to be spaced far apart relative to the size of an individual particle. As a photon traverses through such a participating medium, it has two potential outcomes:

it may either pass through the medium unaffected bypassing all particles, or it may interact with some particles. Importantly, these interactions are considered statistically independent of the outcomes of subsequent interaction events [2]. When an interaction occurs, two physical phenomena may happen and lead to a change of radiance: the photon may be absorbed by the particle, resulting in its conversion to another form of energy such as heat, or the photon may be scattered in a different direction. To mathematically express this change of radiance, this section investigates these phenomena and their models.

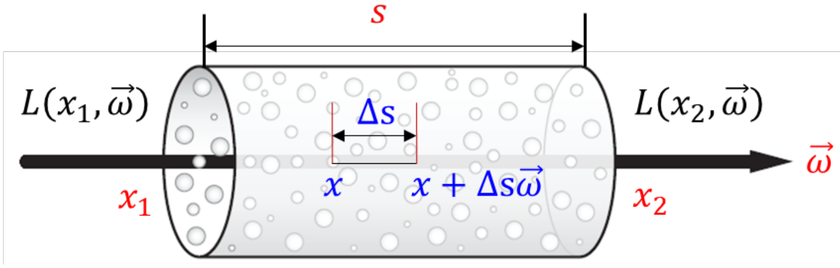


Figure 2.6: A change of radiance as the photons interact with the particles [2]

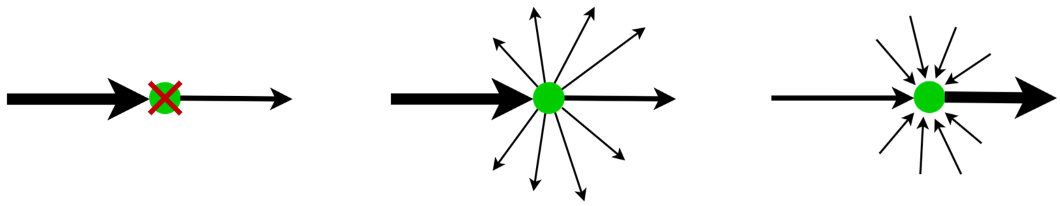
2.4.1.1 Absorption

Let's consider a thin beam of light travelling through a medium along the unit direction $\vec{\omega}$, as shown in Figure 2.6. The number of photons within this beam is proportional to the incident radiance $L(x_1, \vec{\omega})$ at the starting point x_1 of this beam. At each small step Δs along the beam, part of the photons may encounter some particles, and so may become absorbed, as seen in Figure 2.7a. When the photons move from x to a new point $(x + \Delta s \vec{\omega})$ at a distance Δs in the direction $\vec{\omega}$, the radiance $L(x, \vec{\omega})$ becomes $L(x + \Delta s \vec{\omega}, \vec{\omega})$. A change of radiance caused by this type of interaction is related to the absorption coefficient σ_a. It is the differential probability that light is absorbed per unit distance travelled in the medium, and so expressed in m^{-1}. Thus, σ_a can only take positive values, and it is not required to be between 0 and 1 [28]. The expression of σ_a at the point x can be expressed as:

$$\sigma_a(x) = \lim_{\Delta s \to 0} \left(\frac{L(x + \Delta s \vec{\omega}, \vec{\omega}) - L(x, \vec{\omega})}{L(x, \vec{\omega}) \Delta s} \right) = -\frac{\vec{\omega} \cdot \nabla_a L(x, \vec{\omega})}{L(x, \vec{\omega})}, \tag{2.22}$$

where:

- ∇_a represents the gradient due to absorption,
- $(\overrightarrow{\omega} \cdot \nabla_a)$ is the directional derivative along the direction $\overrightarrow{\omega}$.



(a) Absorption (b) Out scattering (c) In scattering

Figure 2.7: three physical phenomena as light travel through participating media

Notice that the value of σ_a depends on the medium characteristics. For instance, Coastal and Harbour are two types of oceans with different turbidities, with σ_a being 0.088 for the former and 0.295 for the latter [29].

2.4.1.2 Scattering

At each segment Δs along the beam, the incident radiance $L(x, \overrightarrow{\omega})$ may also be reduced due to photons being out-scattered into other different directions, as shown in Figure 2.7b. The reduction caused by this event is given by the scattering coefficient σ_s, which is the fraction of total light removed from a beam by scattering per unit distance, and so expressed in m^{-1}. Similar to the absorption coefficient σ_a, σ_s can be only positive values and not limited to the range $[0, 1]$. Thus, the expression of σ_s at x can be computed as:

$$\sigma_s(x) = \lim_{\Delta s \to 0} \left(\frac{L(x + \Delta s \overrightarrow{\omega}, \overrightarrow{\omega}) - L(x, \overrightarrow{\omega})}{L(x, \overrightarrow{\omega}) \Delta s} \right) = -\frac{\overrightarrow{\omega} \cdot \nabla_s L(x, \overrightarrow{\omega})}{L(x, \overrightarrow{\omega})}, \tag{2.23}$$

where:

- $L(x + \Delta s \overrightarrow{\omega}, \overrightarrow{\omega})$ represents the radiance at $(x + \Delta s \overrightarrow{\omega})$ in the direction $\overrightarrow{\omega}$ after scattering event (only considering out-scattering),
- ∇_s represents the gradient due to scattering,
- $(\overrightarrow{\omega} \cdot \nabla_s)$ is the directional derivative along the direction $\overrightarrow{\omega}$.

In addition to loss of radiance, the radiance may be increased as the photons travel through the media. The same process that leads to the reduction of radiance along the direction $\overrightarrow{\omega}$ due to out-scattering, may lead it to increase due to in-scattering from other directions, as see in Figure 2.7c. The increase in radiance due to this interaction is also related to the scattering coefficient σ_s as expressed in Equation (2.23) and the in-scattered radiance L_i, and given by:

$$\overrightarrow{\omega} \cdot \nabla_i L(x, \overrightarrow{\omega}) = \sigma_s(x) L_i(x, \overrightarrow{\omega}), \tag{2.24}$$

where $L_i(x, \overrightarrow{\omega})$ represents all the light from other different directions inside the whole sphere Ω_S in $\mathbb{R}^3$ (corresponding to unit sphere points), scattered into the thin beam along direction $\overrightarrow{\omega}$, and expressed as:

$$L_i(x, \overrightarrow{\omega}) = \int_{\Omega_S} \beta(x, \overrightarrow{\omega}', \overrightarrow{\omega}) L(x, \overrightarrow{\omega}') d\overrightarrow{\omega}', \tag{2.25}$$

where $\beta(x, \overrightarrow{\omega}', \overrightarrow{\omega})$ is the Volume Scattering Function (VSF). It describes the angular distribution of light scattered from $\overrightarrow{\omega}'$ to $\overrightarrow{\omega}$ at x in this medium. As such it can be seen as the probability of light being scattered from all direction into the scattered direction $\overrightarrow{\omega}$. Hence, it must be normalized, leading to the following expression:

$$\int_{\Omega_S} \beta(x, \overrightarrow{\omega}', \overrightarrow{\omega}) \mathrm{d}\overrightarrow{\omega}' = 1 \tag{2.26}$$

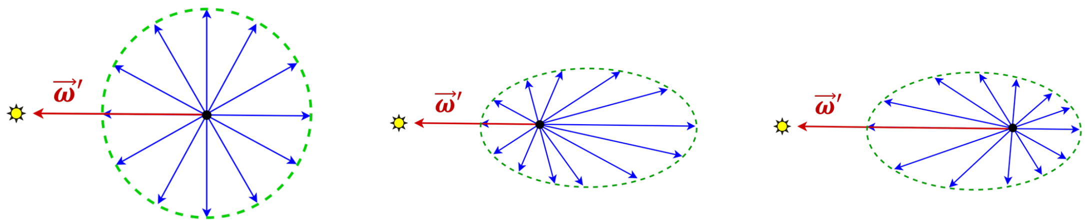
For instance, if light is scattered equally in all directions as shown in Figure 2.8a, then the normalization of this constant function leads to $\beta(x, \overrightarrow{\omega}', \overrightarrow{\omega}) = \frac{1}{4\pi}$. However, many natural media exhibit a preference for scattering light in either forward or backward direction [2], as depicted in Figures 2.8b and 2.8c, respectively. To account for such directional scattering, several non-isotropic VSFs are commonly used in the study of light propagation through participating media. Three examples of VSFs including the Henyey-Greenstein (HG), Two-Term Henyey-Greenstein (TTHG), and Rayleigh VSFs are discussed further below.

Henyey-Greenstein (HG)

HG VSF is widely used in modelling light scattering in a variety of materials such as oceans, clouds, and skin, owing to its simplicity and effectiveness [2]. It is defined as:

$$\beta_{HG}(\theta, g) = \frac{1 - g^2}{4\pi(1 + g^2 - 2g\cos\theta)^{\frac{3}{2}}} \tag{2.27}$$

where θ is the scattering angle between the incident and outgoing directions, g is the



(a) Isotropic scattering (b) Forward-scattering (c) Backward-scattering

Figure 2.8: Three types of scattering

HG asymmetry factor representing the mean cosine of θ considering all scattering directions:

$$g = \overline{\cos\theta}$$

As such, the g factor is required to be in $[-1, 1]$. It determines the degree of forward or backward scattering dominance in the medium. Hence, $g = 0$ corresponds to isotropic scattering, $g > 0$ indicates forward scattering while $g < 0$ means backward scattering. Notice this parameter hinges on the properties of medium. For instance, in terms of water medium turbidity, $g = 0.947$ and $g = 0.9199$ correspond to Coastal and Harbor [29], respectively. These different values of g give rise to distinct scattering distributions, which can be observed in Figure 2.9 (red and blue dash lines). For both Coastal and Harbor, the HG VSF decreases as the scattering angle θ grows. Besides, at very small scattering angles (less than 5°), the value of HG VSF corresponding to Coastal ($g = 0.947$) is more directive compared to that of Harbor ($g = 0.9199$). For bigger angles, we can observe the opposite with small differences.

Two Term Henyey-Greenstein (TTHG)

TTHG is based onto HG VSF [30–32]. It compensates the drawback that HG VSF is not accurate in modelling light scattering in underwater environment for small angle ($\leq 20°$) and large angles ($\geq 130°$) [32]. It is build as the linear interpolation of two HG functions: one in the incoming direction (backward scattering), the other in the opposite direction (forward scattering). The expression of TTHG is given by:

$$\beta_{TTHG}(\theta, g_F, g_B) = \alpha\beta_{HG}(\theta, g_F) + (1-\alpha)\beta_{HG}(\theta, -g_B), \tag{2.28}$$

where:

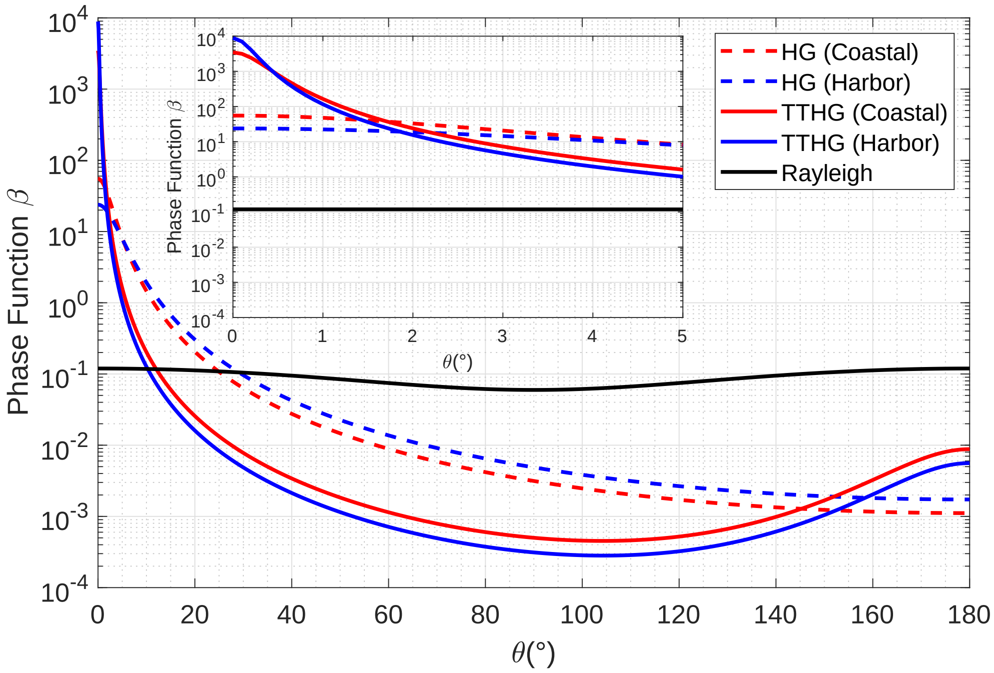


Figure 2.9: Different VSFs varying with the scattering angle θ

- g_F and g_B are respectively forward-directed and backward-directed HG asymmetry factors, respecting the following expression:

$$g_B = -0.3061446 + 1.000568 * g_F \\ - 0.01826338 * g_F^2 + 0.03643748 * g_F^3 \tag{2.29}$$

- α is the weight of the forward-directed HG VSF, expressed in terms of g_F and g_B as follows:

$$\alpha = \frac{g_B(1+g_B)}{(g_B+g_F)(1+g_B-g_F)} \tag{2.30}$$

From [32], α, g_F and g_B are linked through the following relation:

$$\overline{\cos\theta} = \alpha(g_F+g_B) - g_B, \tag{2.31}$$

This average mean $\overline{\cos\theta}$ also respects the following expression according to [33]:

$$\overline{\cos\theta} = 2\frac{1-2B}{2+B}, \tag{2.32}$$

where $B = \sigma_{sB}/\sigma_s$ represents the ratio of backward scattering coefficient σ_{sB} above the scattering coefficient σ_s. Both σ_{sB} and σ_s expressed in $[m^{-1}]$, depends on the medium characteristics, *i.e.* $\{\sigma_{sB} = 0.0014, \sigma_s = 0.216\}$ for Coastal,

$\{\sigma_{sB} = 0.0076, \sigma_s = 1.875\}$ for Harbour. For instance, the parameters $\overline{\cos\theta}$, g_F, g_B and α can be obtained by using the equations from (2.29) to (2.32) with the given σ_{sB} and σ_s, as seen in Table 2.1 for Coastal and Harbour media.

Table 2.1: Relative parameters of TTHG VSF

Water types	$\overline{\cos\theta}$	g_F	g_B	α
Coastal	0.9838	0.9933	0.7054	0.9945
Harbor	0.9899	0.9958	0.7081	0.9966

There are very small differences in these parameters considering Coastal and Harbour, resulting in very close TTHG VSFs, as seen in the red and blue solid lines of Figure 2.9. They decrease as the scattering angles grows regardless of water medium turbidities. Besides, comparing to distribution of HG VSF, it can be observed that TTHG VSF is more directive at very small angles (less than 0.5°) and large angles (more than 155°), and less directive when θ is between 0.5° and 155°. It is also interesting to observe that the TTHG associated with Coastal is more directive compared to Harbour for all scattering angles, except for angles smaller than or equal to 0.5°, which is the opposite case for HG VSF.

Rayleigh

The expression of Rayleigh VSF was first introduced in 1987 for the scattering of light by air molecules [2]. This is a very simple function of a single variable θ. It is derived from the study of a random distribution of very small spheres generating scattering. It is given by:

$$\beta_R(\theta) = \frac{3}{16\pi}(1+\cos^2(\theta)) \tag{2.33}$$

Its distribution of scattering angles is presented as the black solid line in Figure 2.9. The values of this function remain in the range of $[\frac{3}{16\pi}, \frac{6}{16\pi}]$ with a slight decrease from 0° to 90° and a slight increase from 90° to 180°. Comparing to HG or TTHG, this model is close to the isotropic scattering. However, its usage is limited to very small particles composing the medium: the particle's size is up to $\lambda/10$, where λ is the wavelength of the light. For optical communication, this corresponds particles smaller than 35nm for visible light.

2.4.1.3 Transmittance

The loss of radiance due to both absorption and out-scattering is defined as the extinction, and given by the extinction coefficient σ_t:

$$\begin{aligned} \lim_{\Delta s \to 0} \left(\frac{L(x + \Delta s \overrightarrow{\omega}, \overrightarrow{\omega}) - L(x, \overrightarrow{\omega})}{\Delta s} \right) &= \overrightarrow{\omega} \cdot \nabla_t L(x, \overrightarrow{\omega}) \\ &= -(\sigma_a + \sigma_s) L(x, \overrightarrow{\omega}) \\ &= -\sigma_t L(x, \overrightarrow{\omega}) \end{aligned} \quad (2.34)$$

where ∇_t represents the gradient due to absorption and out-scattering.

This differential equation can be solved by computing the transmittance $T(\Delta s)$ between two points at distance Δs [26, 27]. It is defined as the ratio of outgoing radiance $L(x + \Delta s \overrightarrow{\omega}, \overrightarrow{\omega})$ for an infinitesimal volume of length Δs above the incoming radiance $L(x, \overrightarrow{\omega})$:

$$T(\Delta s) = \frac{L(x + \Delta s \overrightarrow{\omega}, \overrightarrow{\omega})}{L(x, \overrightarrow{\omega})} \quad (2.35)$$

Besides, there are 3 important transmittance's properties stating as:

- Multiplication property: $T(\Delta s_1 + \Delta s_2) = T(\Delta s_1) T(\Delta s_2)$,
- Contraction property: $T(\Delta s) \leq 1$,
- Identity property: $T(0) = 1$.

Thus, the loss of radiance is $1 - T(\Delta s)$, leading to the following equation for σ_t:

$$\sigma_t = \lim_{\Delta s \to 0} \frac{1 - T(\Delta s)}{\Delta s}. \quad (2.36)$$

For a homogeneous participating medium, the transmittance is defined as follows:

$$T(\Delta s) = e^{-\sigma_t \Delta s}. \quad (2.37)$$

2.4.2 Radiative transfer equation

Ignoring the emitted radiance by participating media with today's notation, the total change of radiance at point x can be attributed to two factors: the reduction in radiance resulting from absorption and out-scattering, and the augmentation of radiance caused by in-scattering. Thus, the combination of Equation (2.34) and (2.24) yields the following expression:

$$\overrightarrow{\omega} \cdot \nabla L(x, \overrightarrow{\omega}) = -\sigma_t L(x, \overrightarrow{\omega}) + \sigma_s L_i(x, \overrightarrow{\omega}), \tag{2.38}$$

This equation is the well-know radiative transfer equation in the differential form [26, 27]. The term L_i is the incoming radiance at x and given in Equation (2.24). The corresponding integral form of the RTE considering the propagating distance s between the two points x_1 and x_2 as in Figure 2.6 is given by:

$$L(x_2, \overrightarrow{\omega}) = T(s)L(x_1, -\overrightarrow{\omega}) + \int_0^s T(l)\sigma_s L_i(x_l, -\overrightarrow{\omega})\mathrm{d}l, \tag{2.39}$$

where $x_2 = x_1 + s\overrightarrow{\omega}$ represents the point at distance s in direction $\overrightarrow{\omega}$, $x_l = x_1 + l\overrightarrow{\omega}$ are points on the half-line starting at x_1 and with direction $\overrightarrow{\omega}$, and $T(s)$ represents the transmittance between x_1 and x_2, $T(l)$ denotes the transmittance between x_1 and x_l.

2.4.3 Former works on light propagation in participating media

In most practical situations, it is almost impossible to directly solve these above equations which describe how light behaves in participating media. In the sixties, Twomey *et al.* employed the matrix methods to compute the reflection and diffuse transmission of radiation through multiple-scattering parallel layers [34, 35]. To our knowledge, using Monte Carlo Simulation [36, 37] to solve the RTE was first proposed by Collins and Wells during the same period. Considering the different layers of the atmosphere, the sea surface and clouds, they adopted the existing codes initially developed for neutron transport problems, trying to calculate the propagation of light after a nuclear explosion. Later, Plass and Kattawar applied Monte Carlo calculations to study light propagation in the atmosphere containing clouds, and subsequently from the atmosphere to the sea, including reflections on the seabed, taking into account uncollimated light source and laser, visible and infra-red light [38–45]. In all these previous works, it is remarkable that they attempted to establish connections between each scattering point and every receiver considered as a point (surface receiver being handled in another way). This common used technique is known as "Next Event Estimation" (NEE), allowing a significant reduction of the variance. Besides, by employing the differential form of the RTE, they naturally incorporated importance sampling in the selection of subsequent scattering points, utilizing the transmittance as the Probability Density Function (PDF). Additionally, in the choice of the new scattering direction, they also utilized importance sampling with taking the VSF as PDF, albeit they had employed tabulated form of the VSF and so the cumulative distribution to do the sampling. In these first works on Monte Carlo simulation of light propagation, all the roots of modern calculations were already set. Gordon *et al.* used a similar Monte Carlo algorithm to evaluate the light reflected by the sea [46], considering the sea bed [47] and two layers of ocean [48]. In contrast to previous solutions, they used a VSF in [49] obtained from measurement in the "Sargasso sea", and "Tongue of the ocean" (Bahamas island) from Petzold's measurement [50].

Later, Poole *et al.* have proposed to use a "semi-analytic" Monte Carlo method to calculate the propagation of light in ocean, and the flux received by an atmospheric Lidar system [51]. Owing to the low probability of a photon crossing the receiver surface, they adopted NEE to reduce the variance by explicitly connecting to it each scattering point. In fact, this works corresponds to the Collins' method, and the Plass and Kattawar with point receivers, but here with surface receivers. Furthermore, they explicitly used truncated distributions, in order to make sure that the photon stays in the ocean, and does not exit to the atmosphere.

Kirk made use of a Monte Carlo procedure similar to the Plass and Kattawar one [52, 53], with a VSF with respect to very turbid water obtained from the data measured in San Diego's harbour by Petzold [50]. Based on his simulation results, he derived an analytical simple law for light propagation in this kind of seawater. Afterwards, the same Monte Carlo method was utilized by Lermer and Summers with the Henyey-Greenstein VSF [54] for the first time to our knowledge. Taking into account the angle and the time of arrival of photons at the receiver, they drew some statistics, providing a fact that previous simple analytical propagation models were no more valid for such conditions.

The modern algorithm used in Underwater Wireless Optical Communication (UWOC) channel estimation was first proposed by Prahl *et al.* in [1], albeit it was for collimated light propagation in layered tissue. The Monte Carlo samples were considered as groups of photons instead of individual photons, with Russian roulette used to terminate the propagation of the photon packet, but not a threshold as in previous works. Besides, an organigram of such algorithm was provided and detailed in Figure C.2. This may explain why this work was used later as the starting point, albeit it resembles a lot to all the previous solutions starting from Collins's one.

2.5 Underwater wireless optical communication

In recent years, there has been a growing interest in UWOC as a promising solution to enable high-speed data transmission and communication in underwater environments. This section explores the basic background of UWOC and the previous underwater light propagation models, and provide a detailed implementation of the existing UWOC algorithm.

2.5.1 Motivations and applications

In an increasingly interconnected world, effective communication has become essential, even in the most challenging environments. Underwater Wireless Communication (UWC) has emerged as a pivotal field of study and innovation, driven by the need to bridge the communication gap in the depths of our oceans. They are present in a diverse range of applications with respect to both military and civil domains, such as facilitating submarine communications, revolutionizing the fishing industry, aiding ecological research, *etc.* [18]. Traditionally, UWC has relied on acoustic wireless communications and wireless radio communications. The former enables long-distance transmission but suffers from high latency and low throughput. On the other hand, the latter faces obstacles due to low data rates caused by water's high signal absorption properties. Fortunately, there exists a promising alternative technology allowing to compensate these drawbacks. It is known as underwater wireless optical communications, providing good properties such as high data rates, security due to small propagation range, and large bandwidth. Over the past few decades, there has been a significant amount of literature dedicated to UWOC, as evidenced by the numerous recent surveys exploring the various applications of UWOC [10–12, 18, 55–62].

The development and characterization of UWOC systems necessitate testing the developed communication protocols on various scenarios through a lot of simulations. For this purpose, accurate and fast simulation tools are required. To provide such tools, a well understanding of the propagation mechanisms involved in UWOC channels becomes crucial. This kind of channel is characterized by different physical phenomena as mentioned in Section 2.4.1, inherent to participating media:

- light absorption, owing to particles that typically absorb and convert the light flux into heat;
- light scattering, owing to particles that deflect the luminous flux in other directions, causing the dispersion of the light.

Particularly, the latter holds true for participating media like fog, clouds, and liquids such as good old Cognac or turbid harbour water.

Both absorption and scattering suggest that the UWOC channel can become extremely challenging to simulate. Two kinds of UWOC models have been proposed in the previous works:

- models based on Monte Carlo simulations, where photons are tracked in the scene and counted when they reach the receivers;
- statistical models fitting either measurements or Monte Carlo simulations.

In this work, we focus on the study of the first category. To the best of our knowledge, all previous solutions based on Monte Carlo historically came from the neutron transport problem, and thus from the photon transport. Their foundation of using the formalization of radiative transfer is the *differential form of the Radiative Transfer Equation* (RTE) [27], as detailed in Equation (2.38) of the previous section. In these studies, they utilized a Monte Carlo simulation performed with this differential form of the RTE, considering the propagation of a single photon or a group of photons. But there is a dual form of the RTE given in integral form, which expresses the radiance received at a given point by considering that emitted at another point [27]. In this thesis, we propose to start from scratch with this *integral form of the RTE*, allowing to express the UWOC channel in various ways using different integral equations. As such, it is possible to apply various variance reduction techniques to enhance simulation efficiency and accuracy.

2.5.2 Previous underwater light propagation models

There has been a substantial increase in research efforts focused on UWOC channel estimation in recent years. In [63], Jaruwatanadilok first introduced a reformulation of the differential form of the RTE, and then solved the incoming radiance L_i via a Fourier transformation, leading to a new integral expression. Subsequently, the Monte Carlo integration process was employed to solve this new expression. However, this approach first involves in discretizing the space of incoming directions to evaluate (L_i), which then necessitates balancing accuracy and computation time. The VSF used in this approach was derived from Mie scattering.

Since 2010, numerous papers [29, 32, 64–87] have utilized Monte Carlo simulation techniques to investigate the UWOC channel, by using algorithms very close to the Prahl's solution [1]. Some only considered the photons crossing the receiver's surface [29, 32, 64, 65, 67–83, 85–87], while a few adopted the semi-analytical principle and tried to connect each scattering photon to the receiver [66, 84]. Miramirkhani *et al.* incorporated some obstacles, including divers and underwater vehicles [79], into their study using a commercial tool. However, this work was done without providing either a formalization of such a channel or an algorithm. In this dissertation, we will include a visibility function to generalize our model for this kind of scenarios containing one or more obstacles.

More recently, Ghazy *et al.* studied the misalignment problem in UWOC system using MIMO link [88], with the simulations made by only considering a single scattering event. But for realistic water types, more than one scattering is needed for realistic water types. For instance, clear water needs 22 scatterings for a 12 meters link using Prahl's algorithm and $1E^{-6}$ as threshold value. In the same scenario but with coastal or harbour waters, 40 and 94 scatterings are needed respectively, as shown in Figure 3.7 of the next chapter.

In the study of modelling the UWOC channel, there are some works which tried to solve the RTE using finite difference scheme [89, 90]. Illi *et al.* simulated the light propagation in two dimensions for rectangular cross-section finite propagation domain, based on previous time dependent RTE (TD-RTE) solution [91]. The height of their studied space is only 0.2 meters, with the distance between transmitter and receiver ranging from 1 to 22 meters. In such a configuration, they showed that this finite difference scheme becomes faster than a Monte Carlo simulation algorithm. Nevertheless, the studied domain is too small for many water types and in ocean, as many significant scattered contributions come from outside. Besides, this finite difference scheme necessitated subdividing the propagation domain into both geometrical and direction spaces. In terms of an acceptable accuracy level and large 3-dimensional propagation domains containing obstacles like in [79], the complexity of the TD-RTE method will significantly increase, and its memory requirements will become prohibitive. In contrast, the Monte Carlo variance is related to the square root of the number of samples, irrespective of the dimensionality of the problem. Consequently, in realistic configurations, Monte Carlo simulations tend to become the best solutions for the same level of precision.

In the recent works [90, 92–95], some authors explored the possibility to express the propagation of photons as a probabilistic mechanism. They first investigated the single scattering effect before generalizing to multiple scattering and applying Monte Carlo integration techniques [93, 94]. They also considered multiple scatterings in 2D in [90] or in 3D for others [90, 92, 95]. The logic behind these articles follows the below steps:

- starting from the differential form of the RTE,
- the application of Monte Carlo Simulation leads to probabilities for the step size of photon's movement and the angles of deflection in a photon's trajectory,
- a combination of these probabilities leads to a sort of integral form of the RTE describing the photon arrival probability after n scattering events.

But it is worth noting that the integral form of the RTE was known for a long time, and it is far easier to begin by it instead of through these circumvolutions. That exactly aligns with our purpose:

- first to restart from the integral form of the RTE and to express the light propagation with 0 to n scatterings,
- then to retrieve the classical Prahl's algorithm and to apply some conventional variance reduction techniques resulting in a solution close to the semi-analytical old method, albeit without assuming a locally constant VSF.

Notice that any kind of VSF can be applied in our method.

2.5.3 Prahl's algorithm in details

Most of the previous works relies on Monte Carlo simulation consisting in simulating a physical process with taking some parts of its analytical description as a sampling function. In this context, a sampling function refers to the use of a PDF and a sampling mechanism based on such a density. Such a PDF $p(x)$ is a positive function with unit integral defined over its definition domain D:

$$\forall x \in D,\ p(x) \geq 0, \qquad \int_D p(x)\mathrm{d}x = 1. \tag{2.40}$$

In UWOC channels, the main two physical events are the scattering and the extinction, modelled by the VSF β and the transmittance T, respectively. These two phenomena are directly used into previous simulation algorithms. Unfortunately, Monte Carlo simulation does not provide a complete mathematical expression of the radiance at the receiver after n scattering events. This may explain why there have been no significant changes in the way participating environments are simulated since Prahl's algorithm in 1989. On the contrary, according to Monte Carlo integration method, we start from a mathematical expression of a given problem. This allows for the application of many techniques in reducing the estimator's variance, and so the number of samples for a given error threshold [96]. Importance sampling is one of such techniques, which is inherent to Monte Carlo simulation. Of course, with Monte Carlo integration method, the best variance reduction technique can be chosen, without being limited to the only importance sampling method.

In this section, we provide a detailed implementation used in the UWOC channel, based on previous works (see algorithm 1), more precisely based on [29], itself based on [1]. In this dissertation, this algorithm is named as *Prahl's algorithm*. The lines from 1 to 5 consist in some initialization, for a given path. Then, the loop builds a path from lines 6 to 25. In fact, this algorithm evaluates a single path, thus it should be called N times for a complete simulation. At line 8, this for-loop starts by setting the end of the ray, owing to a PDF $p(l) = \sigma_t T(l)$. Afterwards, lines 9 to 13 manage the ray that passes behind the receiver. In such a case, it is considered that the path cannot cross again the receiver, and so the loop exits at line 12. Before to exit, the intersection between the segment of line $[x_{n-1}, x_n]$ and the receiver R_x is handled at line 11: the function **tryReceiveLineSegment** does this test and if necessary, adds the path's weight **factor** to a global accumulator at specific time corresponding to the path's **length**.

On the lines from 14 to 19, the intersection of the path is tested if necessary (line 14) with the obstacles in the scene (line 15). The intersected object is returned (line 17) if such an intersection occurs. This allows to handle some objects in the participating media, like some authors are doing [79]. Then the next segment of path is built from lines 20 to 24. It extends the path length to the last point x_n on line 20, and updates the path's weight to x_n on line 21. On line 23, it samples a new direction thanks to the VSF β (*i.e.* the PDF is β). On line 24, the rays is reset so that it starts from the previous endpoint x_n and has the direction **newDir** sampled in line 23.

Algorithm 1 Prahl's Algorithm [1]

x_0 = samplePoint(T_x);
$\overrightarrow{\omega_0}$ = sampleDirection(T_x);
ray = createFromStartAndDir(x_0, $\overrightarrow{\omega_0}$);
factor = 1 / $\left(p(x_0)p(\overrightarrow{\omega_0})\right)$;
length = 0;
for n=1 maxScatteringPerLink **do**
    // Sample the length
    x_n =setEndOfRayWithTransmittance(ray, σ_t);
    if isCrossingAPlane(ray, endSurfaceSensor) **then**
        // Segment $[x_{n-1}, x_n]$ crosses R_x?
        tryReceiveLineSegment(ray, factor, length); **return** null;
    end if
    if useVisibility **then**
        hit = sceneIntersection(ray);
        if hit $\neq$ null **then return** hit;
        end if
    end if
    length = length + getDistance(ray);
    factor = factor * σ_s / σ_t;
    // Sample the direction
    newDir = sampleDirection(ray, σ_s);
    setDirectionAndStart(ray, newDir, ray.to);
end forreturn null;

Chapter 3

New Monte Carlo integration models for ULP

The design of efficient UWOC systems heavily relies on the availability of robust simulation tools. Current State-of-the-art simulation algorithms, widely adopted in UWOC research, are based on a paper written by Prahl in 1989 [1], originally developed to study the propagation of light through human tissue. This method relies on Monte Carlo Simulation, tracking the path of a photon through the participating media. However, the highly scattering nature of turbid water demands a significant number of photons to achieve reliable results with a low margin of error. Reducing the variance of a Monte Carlo estimator relies on known techniques, but requires formalizing the problem in the form of an integral. Unfortunately, this integral does not appear explicitly in Prahl's approach. Suggesting evolutions and optimizations for this approach can therefore prove difficult. Perhaps that's why Prahl's algorithm hasn't evolved since 1989...

In this chapter, we focus on a mathematical formalization of light propagation in water or any other participating medium in the form of an integral. This formalization opens up possibilities for future optimizations. A very simple and straightforward variance reduction technique is proposed as an example. Simulation results obtained in this study demonstrate the effectiveness of the proposed technique, which reduces sample variance and thus improves convergence speed compared with Prahl's method.

3.1 Mathematical formalization

In order to formalize the light propagation channel including participating media, this section uses first the volume formalism owing to its simplicity and intuitive representation of the media as a 3D volume (see §3.1.1). However, this formalism exhibits a large variance due to the high variability of both the transmittance function and the VSF. To overcome this issue, the directional formalism is then introduced by transforming the volume integral domain V to the direction's space Ω_S and distance's space L (see §3.1.2). As it separates the directional and distance dependencies of the propagation, more effective variance reduction techniques can be adopted. Finally, this section provides a rigorous mathematical model of the Prahl algorithm in §3.1.3, by using the directional formalism to simulate the propagation of light in participating media accurately. This formalization serves as a starting point to envisage the development of optimized Monte-Carlo algorithms in §3.2.2 and chapters 4 and 5.

3.1.1 Volume light propagation formalization

The characterization of the temporal dynamics of a channel is of great significance in understanding the behaviour of the optical communication system. The Channel Impulse Response (CIR) is a fundamental parameter that fully characterizes the optical channel. By considering only scattering events, it is given by:

$$h(T_x, R_x, t) = \frac{1}{P_t} \sum_{n=0}^{\infty} P_n(T_x, R_x, t), \tag{3.1}$$

where P_t is the total flux radiated by T_x, and n is the number of scattering events. Notice that the impulse response's unit is s^{-1}. According to the integral form of the RTE (expressed in Equation (2.39)), the flux $P_n(T_x, R_x, t)$ received at R_x at time t after n scattering events from T_x, can be expressed as:

$$P_n(T_x, R_x, t) = \int_{A_{T_x}} \overbrace{\int_V \dots \int_V}^{n} \int_{A_{R_x}} f_0^n(\bar{y}) \mathrm{d}x_0 \mathrm{d}x_1 \dots \mathrm{d}x_n \mathrm{d}x_r, \tag{3.2}$$

where A_{T_x} and A_{R_x} are the surfaces of respectively the emitter and the receiver, V is the volume of the participating media, and $\bar{y} = \{x_0, x_1, \dots, x_n, x_r\}$ is a propagating path of light starting from x_0 on T_x and reaching x_r on R_x after n scattering events at locations $x_i \in V$ for $i \in [1, \dots, n]$. The key parameter $f_0^n(\bar{y})$ is the contribution of the path $\bar{y}$. It is defined as:

$$\begin{aligned} f_0^n(\bar{y}) = {} & L_e\left(x_0, \overrightarrow{\omega_0}\right) \left|\overrightarrow{\omega_0} \cdot \overrightarrow{n_{T_x}}\right| \delta(t - \tau) \\ & \times \prod_{i=0}^{n-1} \frac{T\left(l_i\right) V_{is}\left(x_i, x_{i+1}\right) \sigma_s \beta\left(\overrightarrow{\omega_i}, \overrightarrow{\omega_{i+1}}\right)}{l_i^2} \\ & \times T\left(l_r\right) V_{is}\left(x_n, x_r\right) \frac{\left|\overrightarrow{\omega_n} \cdot \overrightarrow{n_{R_x}}\right|}{l_r^2}, \end{aligned} \tag{3.3}$$

where:

- x_0 is the emitting point onto the transmitter.
- x_r is the receiving point onto R_x.

- $l_i = \|\overrightarrow{x_i x_{i+1}}\|$ is the propagation distance between x_i and x_{i+1}.
- $l_r = l_n$ is the propagating distance from x_n to R_x.
- $\overrightarrow{\omega_0}$ is the unit emitting direction from x_0.
- $\overrightarrow{\omega_i} = \overrightarrow{x_i x_{i+1}}/l_i$ is the unit propagation direction at x_i.
- $\overrightarrow{\omega_n} = \overrightarrow{x_n x_r}/l_r$ is the unit propagation direction at x_n.
- $\overrightarrow{n_{T_x}}$ is the normal vector to transmitter's surface.
- $L_e\left(x_0, \overrightarrow{\omega_0}\right)$ is the emitted flux from x_0 in the direction $\overrightarrow{\omega_0}$ at time $t_0 = 0$.
- $\delta(t-\tau)$ is the delta-Dirac function.
- τ is the propagation delay of path $\bar{y}$.
- $T(l_i)$ is the transmittance over l_i.
- $V_{is}(x_i, x_{i+1})$ is the visibility function defined as:

$$V_{is}(x_i, x_{i+1}) = \begin{cases} 1, & \text{if } x_i \text{ and } x_{i+1} \text{ are visible} \\ 0, & \text{else.} \end{cases} \tag{3.4}$$

- σ_s is the scattering coefficient depending on the participating media.
- $\beta\left(\overrightarrow{\omega_i}, \overrightarrow{\omega_{i+1}}\right)$ is the volume scattering function between the incoming direction $\overrightarrow{\omega_i}$ and the outgoing direction $\overrightarrow{\omega_{i+1}}$.
- $\overrightarrow{n_{R_x}}$ is the normal vector to receiver's surface.

Through the utilization of Monte Carlo Integration [19], this integral can be calculated numerically. The corresponding estimator is given by:

$$\widehat{P_n}(T_x, R_x, t) = \frac{1}{N} \sum_{k=1}^{N} \frac{f_0^n(\overline{y_k})}{p(\overline{y_k})}, \tag{3.5}$$

where $p(\overline{y_k})$ is the probability density function (PDF) for a random sampled path $\overline{y_k}$, N is the number of samples $\overline{y_k}$. Given that each individual point x_i of $\overline{y_k}$ is independently sampled, p is expressed as:

$$p(\overline{y_k}) = p(x_0)p(x_1)\dots p(x_n)p(x_r). \tag{3.6}$$

This estimator provides a flexible and efficient way to approximate integrals of high dimension and complexity. It involves generating numerous random samples from the integration domain and using them to estimate the integral value. As $p(\overline{y_k})$ is contingent on the sampling approach employed, a naive approach is to evaluate the estimator by randomly choosing points leading a uniform PDF, on their respective domains, *i.e.* the transmitter surface for x_0, the receiver surface for x_r, and the participating medium volume for other points x_i. However, the variance $Var(\widehat{P_n})$ of

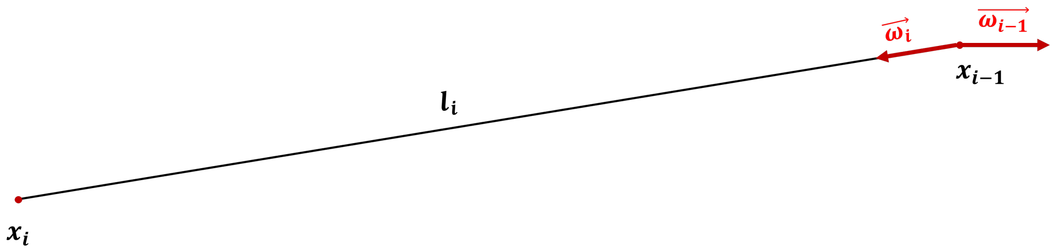


Figure 3.1: An example: a new scattering point very far from the previous scattering point and in backward direction

such an estimator is excessively large. This is primarily due to the high level of dynamic variation in both the transmittance and the VSF. As an example shown in Figure 3.1, a new scattering point very far from the previous one and in backward direction, has very weak values of both transmittance and VSF, leading to a contribution far from the integral value. As a result of this high variance, the estimator exhibits a very large sample error. According to the central limit theorem, the convergence of this method is in $\sqrt{Var(\widehat{P_n})/N}$.

One effective and conventional approach to reduce the variance and error of a Monte-Carlo estimator is to use importance sampling [19]. This technique involves utilizing parts of the integrated function as a PDF, in this case f_0^n. The selected parts of the PDF are chosen in such a way that they vanish in Equation (3.5). This fundamental concept is often utilized in Monte Carlo Simulation algorithms and has been used in previous UWOC simulation tools based on Prahl's algorithm. To employ importance sampling in the context of Prahl's algorithm, scattering points x_i for $i \in [1 \dots n]$ involved in the sample $\overline{y_k}$ are selected using a PDF based on β and T. To do that, each integral over the domain V must be transformed into a pair of integrals on: the space of directions Ω_S with respect to β and on the space of distances L with respect to T.

3.1.2 From volume formalism to directional formalism

It should be noted that when constructing the sample path $\overline{y_k}$, each volume point x_i is selected with knowledge of the previous path point x_{i-1}. The relationship between the volume point x_i and the previous point x_{i-1} is further described in Appendix A. For any given point x_i ($i \geq 1$), it is possible to construct a bijection between the space of points V and those of directions Ω_S and distances L. The transformations due to

Equation (A.2) in Appendix A are therefore as follows:

$$\int_V \mathrm{d}x = \int_{\Omega_S} \int_L l^2 \,\mathrm{d}l \,\mathrm{d}\overrightarrow{\omega}. \tag{3.7}$$

Applying this transformation to Equation (3.2), $P_n(T_x, R_x, t)$ becomes:

$$P_n(T_x,R_x,t) = \int\limits_{A_{T_x}} \overbrace{\int\limits_{\Omega_S}\int\limits_{L} \cdots \int\limits_{\Omega_S}\int\limits_{L}}^{n} \int\limits_{A_{R_x}} f_1^n(\bar{y}) \mathrm{d}x_0 \mathrm{d}\overrightarrow{\omega_0} \mathrm{d}l_0 \ldots \mathrm{d}\overrightarrow{\omega_{n-1}} \mathrm{d}l_{n-1} \mathrm{d}x_r, \tag{3.8}$$

The new parameter $f_1^n(\bar{y})$ deduced from $f_0^n(\bar{y})$ is given as:

$$f_1^n(\bar{y}) = f_0^n(\bar{y}) \prod_{i=0}^{n-1} l_i^2. \tag{3.9}$$

Referring to the expression of f_0^n presented in Equation (3.3), it is possible to express $f_1^n(\bar{y})$ as:

$$\begin{aligned} f_1^n(\bar{y}) = \; & L_e\left(x_0, \overrightarrow{\omega_0}\right) \left|\overrightarrow{\omega_0} \cdot \overrightarrow{n_{T_x}}\right| \delta(t-\tau) \\ & \times \prod_{i=0}^{n-1} T(l_i) V_{is}(x_i, x_{i+1}) \sigma_s \beta\left(\overrightarrow{\omega_i}, \overrightarrow{\omega_{i+1}}\right) \\ & \times T(l_r) V_{is}(x_n, x_r) \frac{\left|\overrightarrow{\omega_n} \cdot \overrightarrow{n_{R_x}}\right|}{l_r^2}, \end{aligned} \tag{3.10}$$

where each point x_i for $i \in [1 \ldots n]$ is obtained from previous one as:

$$x_i = x_{i-1} + l_{i-1} \overrightarrow{\omega_{i-1}} \tag{3.11}$$

3.1.3 A further step towards Prahl's algorithm

To add more importance sampling and so to reduce variance, the final integration domain over the receiver's surface as presented in Equation (3.8), can be replaced with an integral over the direction space Ω_{R_x}. This can be achieved by applying the Equation (A.1) described in Appendix A:

$$\int_{A_{R_x}} dx = \int_{\Omega_{R_x}} \frac{l_r^2}{\left|\overrightarrow{\omega_n} \cdot \overrightarrow{n_{R_x}}\right|} \,\mathrm{d}\overrightarrow{\omega}. \tag{3.12}$$

It should be noted that the integral over Ω_{R_x} represents the incoming radiation from all directions that can reach the receiver. Hence, the expression of $P_n(T_x, R_x, t)$ can be rewritten as:

$$P_n(T_x,R_x,t)=\int\limits_{A_{T_x}}\overbrace{\int\limits_{\Omega_S}\int\limits_{L}\dots\int\limits_{\Omega_S}\int\limits_{L}}^{n}\int\limits_{\Omega_{R_x}} f_2^n(\bar{y})\mathrm{d}x_0\mathrm{d}\overrightarrow{\omega_0}\dots\mathrm{d}l_{n-1}\mathrm{d}\overrightarrow{\omega_n} \tag{3.13}$$

The corresponding parameter $f_2^n(\bar{y})$ can be conceived from $f_1^n(\bar{y})$ as follows:

$$f_2^n(\bar{y})=f_1^n(\bar{y})\frac{l_r^2}{|\overrightarrow{\omega_n}\cdot\overrightarrow{n_{R_x}}|}. \tag{3.14}$$

Thus, the complete expression of f_2^n can be written as:

$$\begin{aligned} f_2^n(\bar{y})= &\ L_e\left(x_0,\overrightarrow{\omega_0}\right)\left|\overrightarrow{\omega_0}\cdot\overrightarrow{n_{T_x}}\right|\delta(t-\tau)\ G(x_n,\overrightarrow{\omega_n}) \\ &\times\prod_{i=0}^{n-1}T\left(l_i\right)V_{is}\left(x_i,x_{i+1}\right)\sigma_s\beta\left(\overrightarrow{\omega_i},\overrightarrow{\omega_{i+1}}\right), \end{aligned} \tag{3.15}$$

Notice that $G\left(x_n,\overrightarrow{\omega_n}\right)$ is introduced to describe the final connection to the receiver. It is defined as:

$$G\left(x_n,\overrightarrow{\omega_n}\right)=T\left(l_r\right)V_{is}\left(x_n,x_r\right), \tag{3.16}$$

where $x_r=A_{R_x}\cap[x_n,\overrightarrow{\omega_n})$ is the intersection point between the receiver surface A_{R_x} and the half-line starting at x_n with direction $\overrightarrow{\omega_n}$.

The last integration domain on the receiver's solid angle Ω_{R_x} can be replaced by the full direction domain Ω_S. However, to ensure the exactness of this new formulation, any additional directions $d\overrightarrow{\omega_n}$ that does not fall within the domain Ω_{R_x} must be multiplied by zero. This adjustment is necessary as the contribution of such directions would be null in the final connection to the receiver. To account for this, the function $H\left(x_n,\overrightarrow{\omega_n}\right)$ is introduced and defined as follows:

$$H\left(x_n,\overrightarrow{\omega_n}\right)=\begin{cases}1, & \text{if } [x_n,\overrightarrow{\omega_n}) \text{ intersects } A_{R_x},\\ 0, & \text{otherwise.}\end{cases} \tag{3.17}$$

For any direction not within Ω_{R_x}, $H\left(x_n,\overrightarrow{\omega_n}\right)$ evaluates to zero. This function ensures that only the relevant directions are included in the summation, and all unnecessary computations are removed. By expanding Ω_{R_x} to full direction space Ω_S, a new transformation can be derived as follows:

$$\int_{\Omega_{R_x}}G(x_n,\overrightarrow{\omega_n})\mathrm{d}\overrightarrow{\omega_n}=\int_{\Omega}G(x_n,\overrightarrow{\omega_n})H(x_n,\overrightarrow{\omega_n})\mathrm{d}\overrightarrow{\omega_n}. \tag{3.18}$$

On this basis, the new formalization of $P_n\left(T_x,R_x,t\right)$ becomes:

$$P_n(T_x, R_x, t) = \int_{A_{T_x}} \overbrace{\int_{\Omega_S} \int_L \dots \int_{\Omega_S} \int_L}^{n} \int_{\Omega_S} f_3^n(\bar{y}) \mathrm{d}x_0 \mathrm{d}\overrightarrow{\omega_0} \dots \mathrm{d}l_{n-1} \mathrm{d}\overrightarrow{\omega_n}, \tag{3.19}$$

where $f_3^n(\bar{y})$ is deduced as:

$$\begin{aligned} f_3^n(\bar{y}) = \; & L_e\left(x_0, \overrightarrow{\omega_0}\right) \left|\overrightarrow{\omega_0} \cdot \overrightarrow{n_{T_x}}\right| \delta(t-\tau) \\ & \times G(x_n, \overrightarrow{\omega_n}) H(x_n, \overrightarrow{\omega_n}) \\ & \times \prod_{i=0}^{n-1} T\left(l_i\right) V_{is}\left(x_i, x_{i+1}\right) \sigma_s \beta\left(\overrightarrow{\omega_i}, \overrightarrow{\omega_{i+1}}\right). \end{aligned} \tag{3.20}$$

3.1.4 Prahl's algorithm formalization

In the context of the previous Prahl's algorithm, importance sampling mechanism was employed for both the phase and transmittance functions to compute the next scattering point, even including the last segment of the path $\bar{y}$ from x_n to x_{n+1}. This implies that there are $n+1$ integrals on the double domain $\Omega_S \times L$, while Equation (3.19) involves $n+1$ integrals on Ω_S but only n integrals on L. To retrieve the integral corresponding to Prahl's algorithm, an additional integration domain on L must be introduced for the last segment.

More precisely, Prahl utilizes importance sampling based on the transmittance function to sample the length of the last segment of the path. This allows to find the point x_{n+1} on the half-line originating from x_n in the direction of $\overrightarrow{\omega_n}$. Consequently, Prahl has to verify whether the line segment $[x_n x_{n+1}]$ intersects with the receiver using the following function:

$$H'\left(x_n, x_{n+1}\right) = \begin{cases} 1, & \text{if } [x_n x_{n+1}] \text{ intersects } A_{R_x}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.21}$$

Then, the extra final integral corresponding to Prahl's algorithm is as follows:

$$\int_L g(l_n) \times H'\left(x_n, x_{n+1}\right) \, \mathrm{d}l_n = 1, \tag{3.22}$$

It is worth noting that $H'(x_n, x_{n+1}) = 1$ implies $H(x_n, \overrightarrow{\omega_n}) = 1$. However, the reverse may not be true. Additionally, g can be any function that satisfies the Equation (3.22).

Combining Equations (3.19) and (3.22), the formalization for Prahl's algorithm is given by:

$$P_n(T_x, R_x, t) = \int_{A_{T_x}} \overbrace{\int_{\Omega_S} \int_L \cdots \int_{\Omega_S} \int_L}^{n+1} f_4^n(\bar{y})\, \mathrm{d}x_0 \mathrm{d}\overrightarrow{\omega_0} \mathrm{d}l_0 \ldots \mathrm{d}\overrightarrow{\omega_n} \mathrm{d}l_n, \tag{3.23}$$

where the contribution parameter $f_4^n(\bar{y})$ is updated as:

$$\begin{aligned} f_4^n(\bar{y}) = \; & L_e\left(x_0, \overrightarrow{\omega_0}\right) \left|\overrightarrow{\omega_0} \cdot \overrightarrow{n_{T_x}}\right| \\ & \times \prod_{i=0}^{n-1} T(l_i) V_{is}(x_i, x_{i+1}) \sigma_s \beta\left(\overrightarrow{\omega_i}, \overrightarrow{\omega_{i+1}}\right) \\ & \times g(l_n) G\left(x_n, \overrightarrow{\omega_n}\right) H'(x_n, x_{n+1}). \end{aligned} \tag{3.24}$$

Referring to Equation (3.5), the Monte Carlo estimator with respect to Prahl's algorithm is deduced as:

$$\widehat{P_n}(T_x, R_x, t) = \frac{1}{N} \sum_{k=1}^{N} \frac{f_4^n(\overline{y_k})}{p(\overline{y_k})}. \tag{3.25}$$

In his work, Prahl uses the following probability $p(\overline{y_k})$:

$$\begin{aligned} p(\overline{y_k}) &= p(x_0)\, p\left(\overrightarrow{\omega_0}\right) p(l_0) \ldots p\left(\overrightarrow{\omega_n}\right) p(l_n) \\ &= p(x_0) \prod_{i=0}^{n} p\left(\overrightarrow{\omega_i}\right) p(l_i), \end{aligned} \tag{3.26}$$

where:

- $p(x_0)$ depends on the emitter area,
- $p\left(\overrightarrow{\omega_0}\right)$ depends on the radiation pattern of the emitter,
- $p\left(\overrightarrow{\omega_{i+1}}\right) = \beta\left(\overrightarrow{\omega_i}, \overrightarrow{\omega_{i+1}}\right)$ for $i \geq 0$,
- $p(l_i) = \sigma_t T(l_i)$.

To end the formalization of Prahl's previous work, $g(l_n)$ should be chosen to use importance sampling over the last integration domain L, according to $p(l_n)$. It leads to:

$$g(l_n) = c \cdot p(l_n) = c \cdot \sigma_t T(l_n), \tag{3.27}$$

where c is a normalization constant introduced to respect the Equation (3.22). This constant can be computed as follows:

$$
\begin{aligned}
1 =& \int_{L} g(l_n) \times H'(x_n, x_{n+1})\, \mathrm{d}l_n \\
1 =& \int_{0}^{l_r} g(l_n) \times 0 \mathrm{d}l_n + \int_{l_r}^{\infty} g(l_n) \times 1 \mathrm{d}l_n \\
1 =& \int_{l_r}^{\infty} g(l_n) \mathrm{d}l_n \\
1 =& c \int_{l_r}^{\infty} \sigma_t T(l_n) \mathrm{d}l_n \\
1 =& c\sigma_t \left| \frac{e^{-\sigma_t l_n}}{-\sigma_t} \right|_{l_r}^{\infty} \\
c =& \frac{1}{T(l_r)}.
\end{aligned}
\tag{3.28}
$$

Hence, the expression of $g(l_n)$ is:

$$
g(l_n) = \frac{\sigma_t T(l_n)}{T(l_r)}. \tag{3.29}
$$

The product $g \times G$ divided by the $p(l_n)$ simplifies as follows:

$$
g(l_n) \frac{G(x_n, \overrightarrow{\omega_n})}{p(l_n)} = V_{is}(x_n, x_r)\, \delta(t - \tau_k). \tag{3.30}
$$

Besides, using importance sampling into Equation (3.23) leads to a simplification as:

$$
\begin{aligned}
\prod_{i=0}^{n-1} \frac{T(l_i)\sigma_s \beta\left(\overrightarrow{\omega_i}, \overrightarrow{\omega_{i+1}}\right)}{p(l_i) p(\overrightarrow{\omega_{i+1}})} &= \prod_{i=0}^{n-1} \frac{T(l_i)\sigma_s \beta\left(\overrightarrow{\omega_i}, \overrightarrow{\omega_{i+1}}\right)}{\sigma_t T(l_i) \beta(\overrightarrow{\omega_i}, \overrightarrow{\omega_{i+1}})} \\
&= \prod_{i=0}^{n-1} \frac{\sigma_s}{\sigma_t} \\
&= \left(\frac{\sigma_s}{\sigma_t}\right)^n.
\end{aligned}
$$

At last, the MC estimator corresponding to Prahl's algorithm is simplified to the following expression:

$$
\begin{aligned}
\widehat{E}_n(T_x, R_x, t) = \frac{1}{N} \sum_{k=1}^{N} & \frac{L_e\left(x_0, \overrightarrow{\omega_0}\right) \left|\overrightarrow{\omega_0} \cdot \overrightarrow{n_{T_x}}\right|}{p(x_0) \cdot p\left(\overrightarrow{\omega_0}\right)} \\
& \times \left(\frac{\sigma_s}{\sigma_t}\right)^n \prod_{i=0}^{n} V_{is}(x_i, x_{i+1}) \\
& \times\ \delta(t - \tau_k) \mathrm{H'}(x_n, x_{n+1}).
\end{aligned}
\tag{3.31}
$$

3.2 Reducing the variance

This section presents a comprehensive analysis of variance reduction techniques pertaining to the proposed formalization of UWOC propagation channel. In addition, a novel algorithm designed to further decrease variance is proposed. The effectiveness of the new algorithm is demonstrated through simulation results, showcasing significant improvements in performance.

3.2.1 How to reduce the variance

Monte Carlo methods are widely used for solving complex mathematical problems through the generation of random samples. However, these methods are known to suffer from high variance, which can lead to inaccurate estimates and poor convergence rates. Several techniques have been developed to mitigate this issue, including stratified sampling, importance sampling, and control variates, as discussed in [96]. Since Prahl's algorithm is based on the importance sampling intrinsic to Monte Carlo Simulation, we only focus on importance sampling.

In light of our first formalization of the UWOC propagation channel represented by the function f_0^n, the application of importance sampling is subject to certain limitations. Indeed, the integration domains are A_{T_x}, V and A_{R_x}. Given these constraints, importance sampling may be applied for:

- choosing starting point x_0,
- choosing scattering point x_i in the volume,
- choosing the reception point x_r.

Notably, it is impossible to directly select any point on the path $\bar{y}$ with importance from f_0^n, since f_0^n does not involve explicitly the points of the domains. For instance, x_0 is used in L_0 which is a generic function, and then only uniform sampling can be done. On the basis of this formalization and with uniform sampling, we wrote a first algorithm that gave mediocre results, as expected. As presented in Equation (3.3), the more significant components of f_0^n are the VSF β, the transmittance T and the visibility function V_{is}. The latter is difficult to predict analytically, and since generally the UWOC channel does not include many obstacles if any, we mainly focus here on β and T. Thereby it is more interesting to choose the next point x_{i+1} from the point x_i using β and T. The sampling spaces are Ω_S for the former and L for the latter. To achieve this, the second formalization using f_1^n is required. Additionally, we can apply importance sampling to the last scattering point, but this excludes the use of f_1^n. Moreover, applying importance sampling according to β on f_2^n would require normalizing β on Ω_{R_x}, which seems to be very difficult in practice (albeit it is the subject of Chapter 4). At last and since Prahl had implicitly used the form f_4^n, we propose to define an algorithm based on f_3^n.

3.2.2 New UWOC algorithm Xiao 1

The expressions f_3^n and f_4^n are very similar, with the latter introducing a final integration domain on L. This additional integration domain requires the inclusion of an extra term $g(l_n)$, that satisfies Equation (3.22), along with a new version of the hit function H'. From an algorithmic perspective, the key implication of this difference is that the contribution is added after sampling the length l_n and only when $H' = 1$.

In contrast, using the expression f_3^n, the contribution should be added when the direction $\overrightarrow{\omega_n}$ is known, which occurs one step earlier than in Prahl's algorithm, as shown in line 8 of Algorithm Xiao 1. It is worth noting that the two algorithms differ only slightly. Specifically, instead of checking for intersection with the function H', we must check for intersection with the function H. The main difference is that the Prahl's algorithm uses a line segment $[x_n, x_{n+1}]$, while the new one uses a half-line $[x_n, \overrightarrow{\omega_n})$.

Using the f_3^n function increases the probability of generating a sampling path with a non-zero contribution compared to using the f_4^n function. This property has an interesting implication: the Prahl's algorithm adds at most one non-zero contribution per path, while our new algorithm has the potential to add up to $n+1$ non-zero contributions for a single path. Although one might argue that each of these contributions is not sampled using the transmittance for the very last segment, and thus intuitively the variance would increase compared to Prahl's algorithm, this concern can be mitigated. Even for clear water, many scattering events are required to reach the receiver, and thus the higher number of contributions can compensate for this aspect. The validity of this assertion is confirmed by the results presented in Section 3.3.

Note that this new algorithm proposes an early exit from the loop (lines 11 to 13), as does Prahl's algorithm. This involves checking whether the new scattering point x_n lies

Algorithm 2 Algorithm Xiao 1

```
x0 = samplePoint(Tx);
ω0 = sampleDirection(Tx);
ray = createFromStartAndDir(x0, ω0);
factor = 1 / (p(x0)p(ω0));
length = 0;
for n=1 maxScatteringPerLink do
    // Try to connect
    tryReceiveHalfLine(ray, factor, length);
    // Sample the length
    setEndOfRayWithTransmittance(ray, σt);
    if isCrossingAPlane(ray, endSurfaceSensor) then return null; // Early stop
    end if
    if useVisibility then
        hit = sceneIntersection(ray);
        if hit ≠ null then return hit;
        end if
    end if
    length = length + getDistance(ray);
    factor = factor * σs / σt;
    // Sample the direction
    newDir = sampleDirection(ray, σs);
    setDirectionAndStart(ray, newDir, ray.to);
end forreturn null;
```

behind the receiver plane. In such a case, it is assumed that the probability of returning in front of the receiver to cross it is very low.

3.3 Performance evaluation and discussions

This section explores some simulation results obtained from our implementation of two UWOC impulse response estimation algorithms. The first one corresponds to the Prahl's Algorithm [1]. The second one is our new approach, described in Algorithm Xiao 1 and based on Equation the (3.19).

To evaluate and compare the performance of our new algorithm with Prahl's one, we begin with a well-known configuration from the literature [29]. However, as a single configuration cannot provide conclusive results, we then use multiple configurations

Table 3.1: Initial Configuration.

Parameters	Description
T_x	Lineic transmitter Maximum radiation angle $\theta_{\max} = 5°$ Beam length $w = 3$ mm Located at $(0,0,0)$ Radiating direction $(1,0,0)$
R_x	Disc receiver Field Of View (FOV) set to 180° Radius set to 10 cm Located at $(12,0,0)$ Oriented to $(-1,0,0)$
Scattering phase function	Henyey-Greenstein function $g = 0.9199$
Threshold	$1E^{-6}$
Default water	Harbor water $\sigma_a = 0.295$ and $\sigma_s = 1.875$ $maxScatterings = 94$

by modifying one parameter at a time. Specifically, we change parameters such as the area and field of view of R_x, the distance between T_x and R_x, and the orientation and position of R_x relative to T_x. By systematically changing these parameters, we can analyze how the algorithms perform under different scenarios and gain a more comprehensive understanding of their respective strengths and weaknesses.

3.3.1 Methodology

Performance of the different considered algorithms are evaluated firstly using the initial configuration proposed in [29] and detailed in Table 3.1. This environment has no obstacle but only water, so the volume of water is considered as infinite. To stop the propagation of a photon, *i.e.* a ray, Prahl considers the notion of photon's weight which is equal to $\left(\frac{\sigma_s}{\sigma_t}\right)^n$ after n scatterings. A photon is stopped when its weight falls under a given threshold. So the corresponding maximum number of scattering events $maxScattering$ is deduced from the threshold T_h as:

$$maxScattering = \frac{\log T_h}{\log \sigma_s - \log \sigma_t} \tag{3.32}$$

Each simulation is conducted using 2 billion of Monte Carlo samples (paths), on a Precision 7920 Tower server with two Intel(R) Xeon(R) Gold 6240 running at 2.60 GHz and 92 Gb of RAM, and using 72 threads. These simulations generate discrete CIR with a time step of $\Delta t = 0.1$ nanoseconds. From these impulse responses, we can calculate the channel's gain C_g defined as follows:

$$C_g(T_x, R_x) = \int_0^\infty h(T_x, R_x, t)\mathrm{d}t. \tag{3.33}$$

The comparison of the two algorithms, *i.e.* Prahl and Xiao 1, involves evaluating their computation time, unbiased sample variance, and sample error. The unbiased sample variance is calculated using the samples from the population, which correspond to the paths used during the simulation. Let C_g^k be the contribution of the path $\overline{y_k}$ to the channel's gain $C_g(T_x, R_x)$, and let $\overline{C_g}$ be the average of these contributions. The unbiased sample variance of $C_g(T_x, R_x)$ is then as follows:

$$Var(C_g) = \frac{1}{N-1}\sum_{k=1}^{N}\left(C_g^k - \overline{C_g}\right)^2. \tag{3.34}$$

From this sample variance, the estimation error of $C_g(T_x, R_x)$ is computed as follows:

$$\mathscr{N}(C_g) = \sqrt{\frac{Var(C_g)}{N}}. \tag{3.35}$$

The following sections provide first comparison results between static gains, CIR, computation times, and unbiased sample variances between Prahl's and Xiao 1 algorithms from the initial Configuration environment. Then, they extend these study to different configurations.

3.3.2 Performance comparison with initial configuration

This section provides a direct comparison between the two algorithms using the initial configuration described above. Figure 3.2 displays the impulse responses generated by both algorithms with 2 billion samples. Although our new solution costs more time as shown in Table 3.2, the difference in computation time is very small, only 37 seconds. However, its unbiased sample variance has been reduced to 81.18% of Prahl's one, indicating more accurate results provided by this new solution. As a result, the sizes of the 95% confidence intervals of the two algorithms are $0.262E^{-7}$ and $0.236E^{-7}$ for Prahl's algorithm and the new algorithm, respectively.

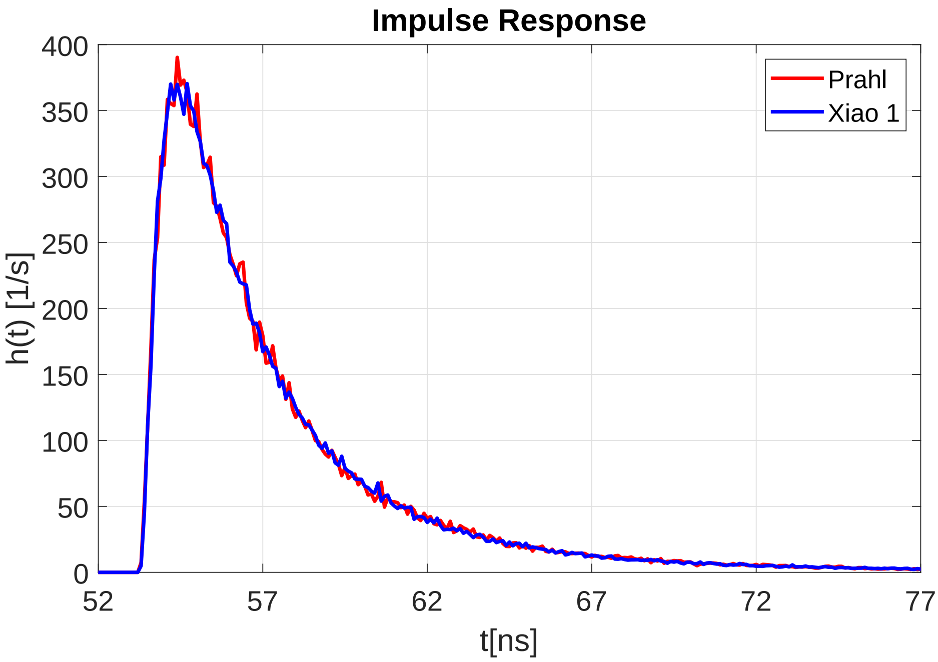


Figure 3.2: CIR obtained with initial configuration.

Table 3.2: Simulation Results for initial configuration using Prahl's Algorithm [1] and Algorithm Xiao 1 .

	Prahl's Algorithm [1]	Algorithm Xiao 1
Channel Gain	$1.6213E^{-6}$	$1.6176E^{-6}$
Unbiased sample variance	$8.9481E^{-8}$	$7.2641E^{-8}$
Sample error	$6.6888E^{-9}$	$6.0267E^{-9}$
Confidence interval with a 95% confidence level	$\left[1.6082E^{-6}, 1.6344E^{-6}\right]$	$\left[1.6058E^{-6}, 1.6294E^{-6}\right]$
Computation time	13 minutes and 15 seconds	13 minutes and 52 seconds

Figure 3.3 illustrates the local variance $C_g(T_x, R_x, i)$ of the two simulations, with respect to the unbiased sample variance of the channel's gain computed into each temporal bin of the CIR. The formula for this variance is given by:

$$C_g(T_x, R_x, i) = \int_{i\Delta t}^{(i+1)\Delta t} h(T_x, R_x, t)\mathrm{d}t. \tag{3.36}$$

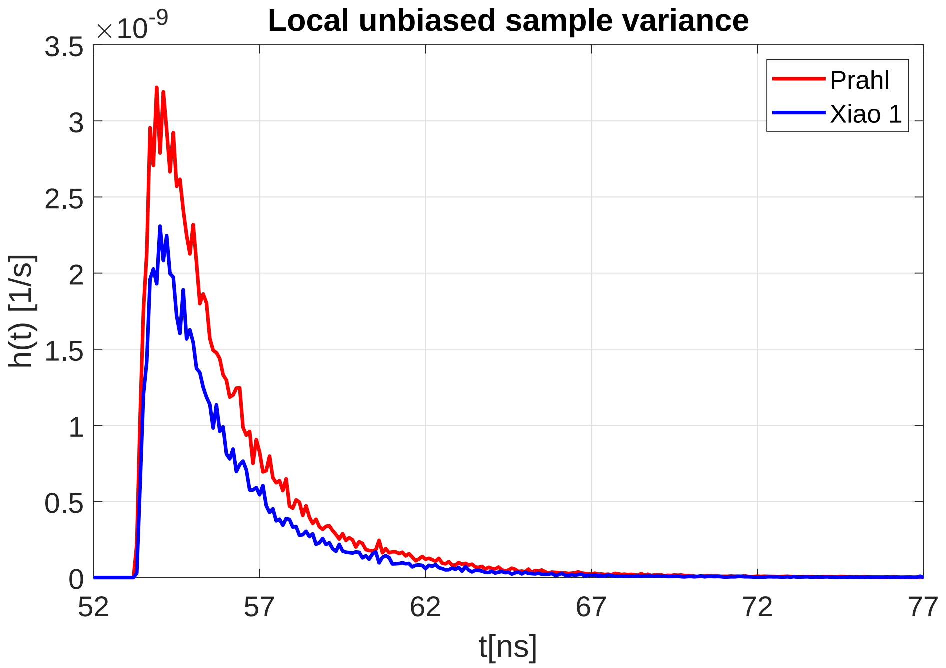


Figure 3.3: Local unbiased sample variances obtained with initial configuration.

This figure clearly shows that our algorithm reduces the sample variance. This observation supports the result obtained from Figure 3.2, which suggests that our new algorithm has better convergence. Furthermore, we observe a reduction in both the estimator's variance and error with only a slight increase in computation time. This is a significant point to note since increasing the number of paths for Prahl's algorithm is necessary to achieve the same level of error as the new algorithm. This increase in the number of paths will consequently lead to an increase in computation time. By denoting $Var(C_g^1)$ the sample variance and N_1 the number of paths for the first algorithm, and $Var(C_g^2)$ and N_2 the ones of the second algorithm, we can estimate this required number of paths as:

$$\begin{aligned} & \mathcal{N}(C_g^1) = \mathcal{N}(C_g^2) \\ \Longrightarrow \quad & \sqrt{\frac{Var(C_g^1)}{N_1}} = \sqrt{\frac{Var(C_g^2)}{N_2}} \\ \Longrightarrow \quad & N_1 = \frac{Var(C_g^1)}{Var(C_g^2)} N_2. \end{aligned} \tag{3.37}$$

To attain an error value equivalent to that of the new algorithm, Prahl's algorithm necessitates 2.4637 billion paths. This results in a computation time of 16 minutes and 21 seconds, in contrast to the 13 minutes and 52 seconds required by Xiao 1 algorithm

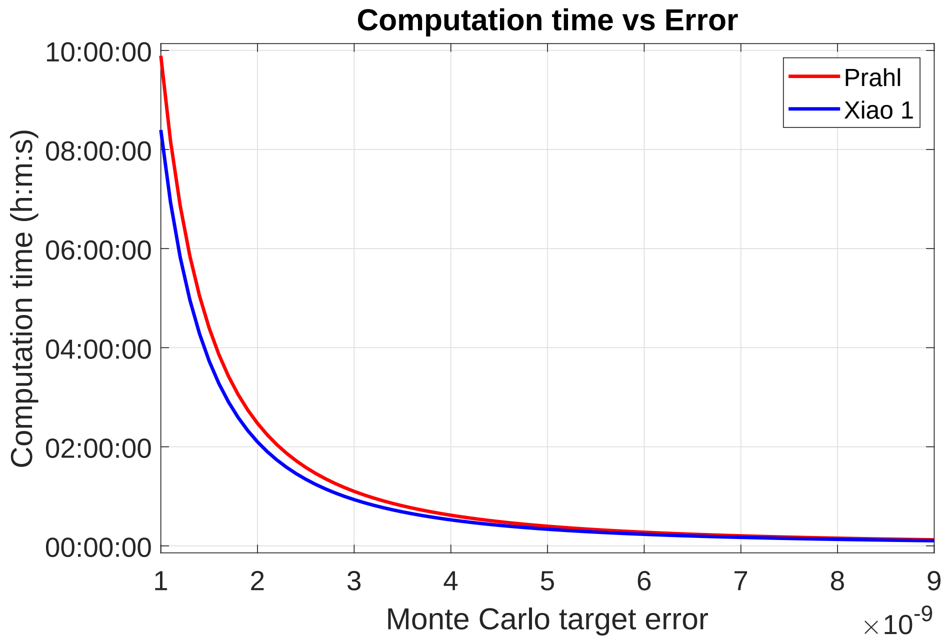


Figure 3.4: Expected computation time with initial configuration varying the target error.

to achieve the same error rate. This shows that the new algorithm is faster than the Prahl's one for a given target quality.

To enhance this result, Figure 3.4 illustrates the expected computation time for the two algorithms as the Monte Carlo target error varies within the range of $[1E^{-9}\ldots 9E^{-9}]$. At the minimum target error, Prahl's algorithm requires 9 hours and 53 minutes, while the new algorithm takes 8 hours and 35 minutes only. The corresponding CIR can be seen in Figure 3.5. Notably, for the same low target error, Prahl's algorithm necessitates 89.5 billion paths, whereas the new algorithm requires 72.6 billion paths only. On the other hand, considering the maximum target error depicted in Figure 3.4, the computation time decreases significantly to 7 minutes and 23 seconds for Prahl's algorithm, and 6 minutes and 13 seconds for the new algorithm. The corresponding CIR are displayed in Figure 3.6. It is worth mentioning that for a high target error, Prahl's algorithm uses 1.1 billion paths, while Xiao 1 algorithm uses only 897 million paths. Overall, the new algorithm provides better performance in terms of computation time for all target errors.

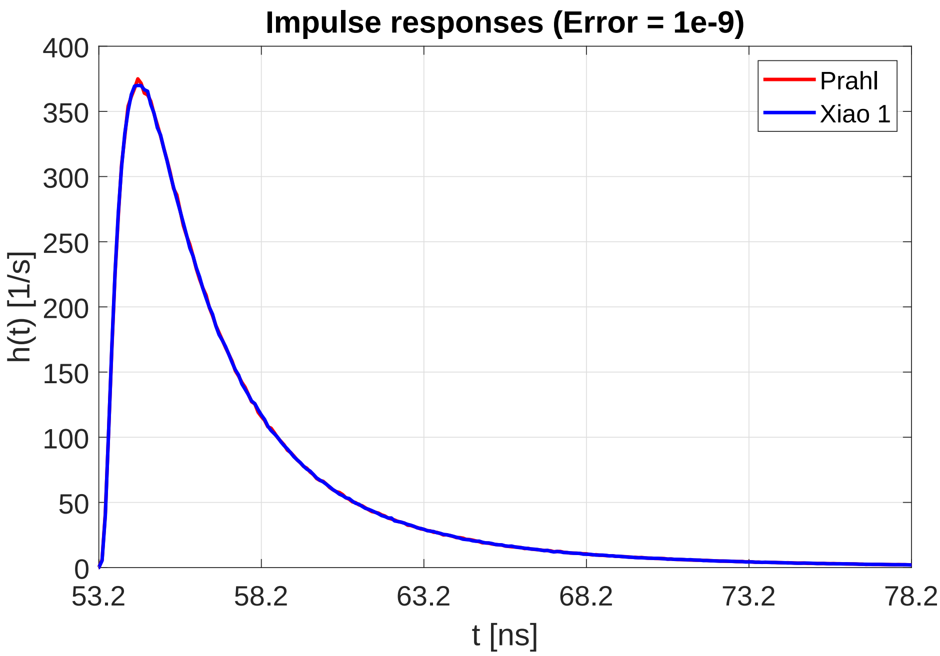


Figure 3.5: CIR with initial configuration and low target error.

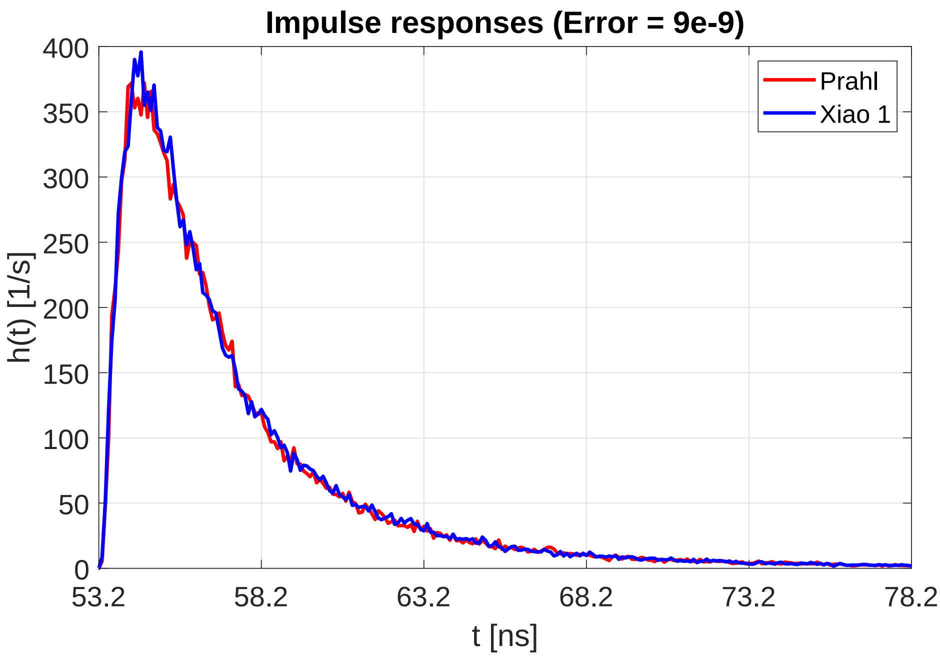


Figure 3.6: CIR with initial configuration and high target error.

Table 3.3: Parameters for different water type.

Water	σ_a	σ_s	g	maxScattering
Pure Sea	0.053	0.003	0.924	4
Clear Ocean	0.069	0.08	0.924	22
Coastal	0.088	0.216	0.947	40
Harbor	0.295	1.875	0.9199	94

3.3.3 Using different water types

UWOC systems can be deployed in different water types, so it's imperative to investigate their impacts on the variance of the simulation algorithms. Referring to [32], different parameters can be found with respect to 4 different kinds of water: Pure Sea, Clear Ocean, Coastal and Harbor. These parameters are summarized in Table 3.3. The right column indicates the maximum number of scatterings for the considered threshold, *i.e* $1E^{-6}$ as stated in the Initial configuration.

On this basis, this section investigates the impact of the different number of scattering events (denoted as $NbScatterings$) on the channel gain for different water types, as shown in Figure 3.7. Note that $NbScatterings$ being 0 indicates a line-of-sight (LOS) link. The channel gain tends to be higher for less diffuse water, as fewer light rays are scattered, resulting in a higher proportion of incident light being transmitted and arriving at the receiver. Specifically for Pure Sea, it remains approximately constant (around $-23.31dB$). This is due to the fact that the main contributions come from the LOS link, and the contributions involving 1 to 4 scattering events are less significant. In contrast, for Clear ocean, Coastal, and Harbor, there is an increase in channel gains as $NbScatterings$ grows until 2, 6, and 30. After these points, they start to stabilize at $-25.55dB$, $-28.16dB$, $-58.26dB$, respectively. However, there are still some small contributions before reaching the maximum scatterings for these three water types. To obtain accurate results, it is necessary to use a sufficient number of scatterings, namely the maximum number of scattering events, given in Table 3.3 for each water type considering the considered threshold.

The disparity between the parameters of the Harbor water characteristics and those of the other water types is significant, particularly when it comes to the scattering coefficient. This variation explains the difference in channel gain values with sufficient

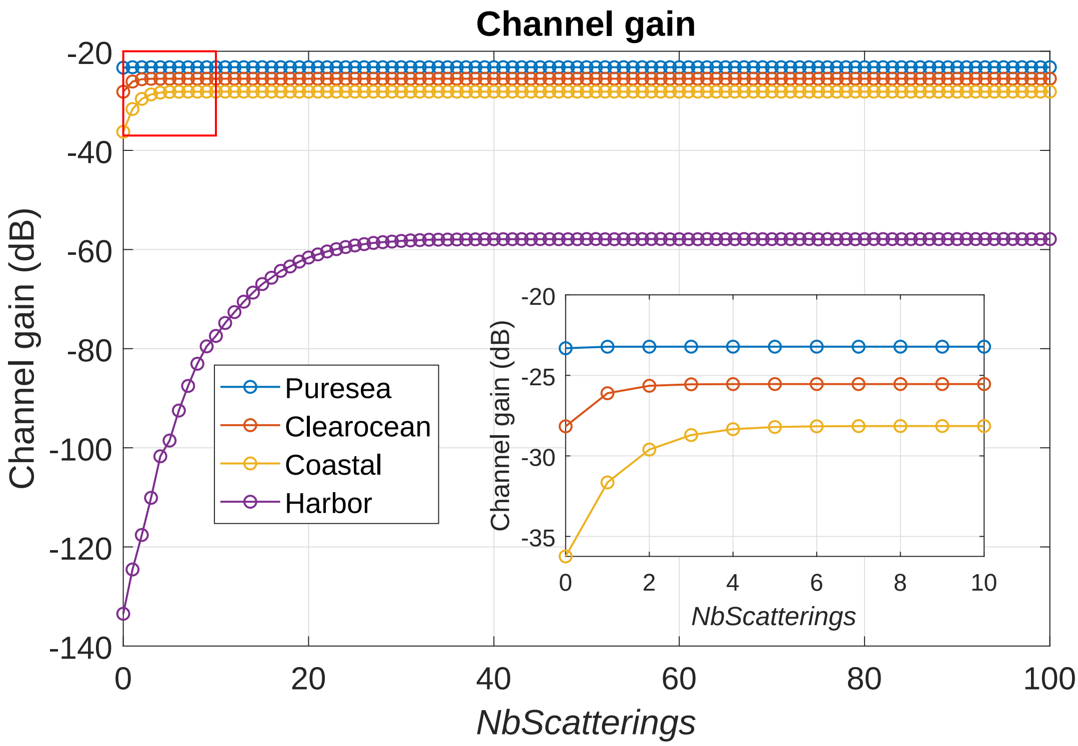


Figure 3.7: Channel gain obtained from Algorithm Xiao 1 with different numbers of scatterings

number of scattering events, which are detailed in Table 3.4. The ratio in the table is defined as:

$$Ratio = \frac{Var_{(Xiao1)}(C_g)}{Var_{(Prahl)}(C_g)} \tag{3.38}$$

where $Var_{(Xiao1)}(C_g)$ and $Var_{(Prahl)}(C_g)$ are the unbiased sample variance corresponding to Algorithm Xiao 1 and Prahl's Algorithm [1], computed using Equation (3.34). Clearly, the new solution reduces the variance regardless of water types, owing to ratio being always lower than 1. This difference is also illustrated by the Figures 3.8-3.10: most of the light flux is received in less than 1 ns for Pure Sea, Clear Ocean and Coastal waters, as compared to the 25 ns shown in Figure 3.2. These observations further emphasize the importance of taking water type into account when conducting simulations.

Table 3.4: Unbiased sample variance obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the water type.

Water type	Channel gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
Pure Sea	$4.7663E^{-3}$	$4.6429E^{-3}$	$2.3760E^{-3}$	51.17%
Clear Ocean	$2.7930E^{-3}$	$2.1041E^{-3}$	$6.6040E^{-4}$	31.39%
Coastal	$1.5327E^{-3}$	$8.9104E^{-4}$	$3.8410E^{-4}$	43.11%
Harbor	$1.6176E^{-6}$	$8.9481E^{-8}$	$7.2641E^{-8}$	81.18%

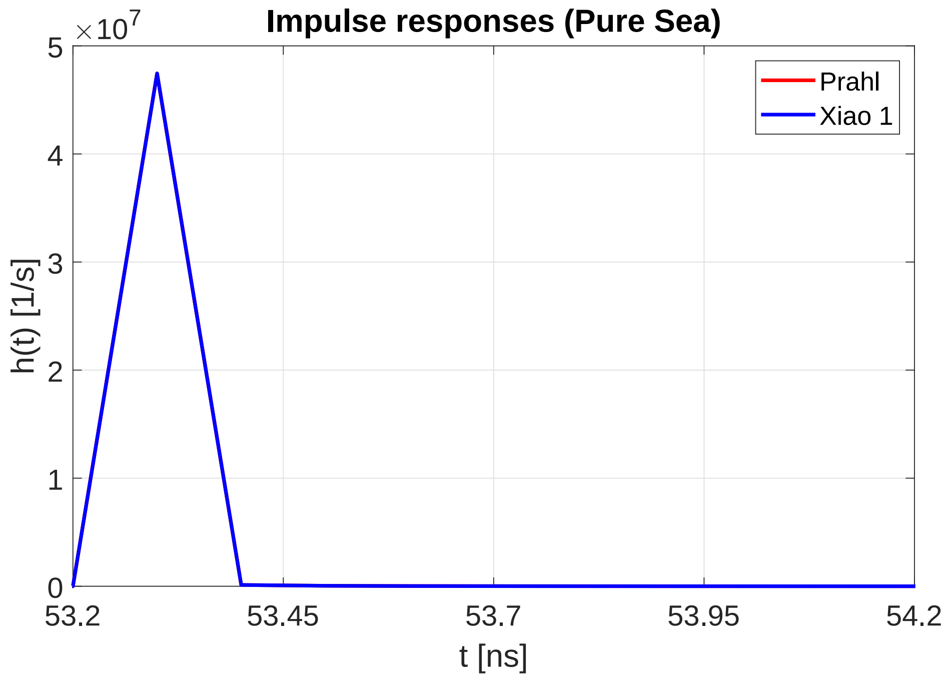


Figure 3.8: CIR obtained with Pure Sea.

3.3.4 Changing receiver's area and FOV

The performance of the Xiao 1 algorithm could be subject to various parameters associated with the simulated environment. One crucial factor likely to influence its performance is the area of the receiver. Note that the probability of a path $\bar{y}$ crossing the receiver is directly related to its size. Consequently, changes to the receiver's area could have a substantial effect on the algorithm's efficiency. Table 3.5 illustrates the impact of the receiver's area A_{R_x} on the unbiased sample variance. By varying the

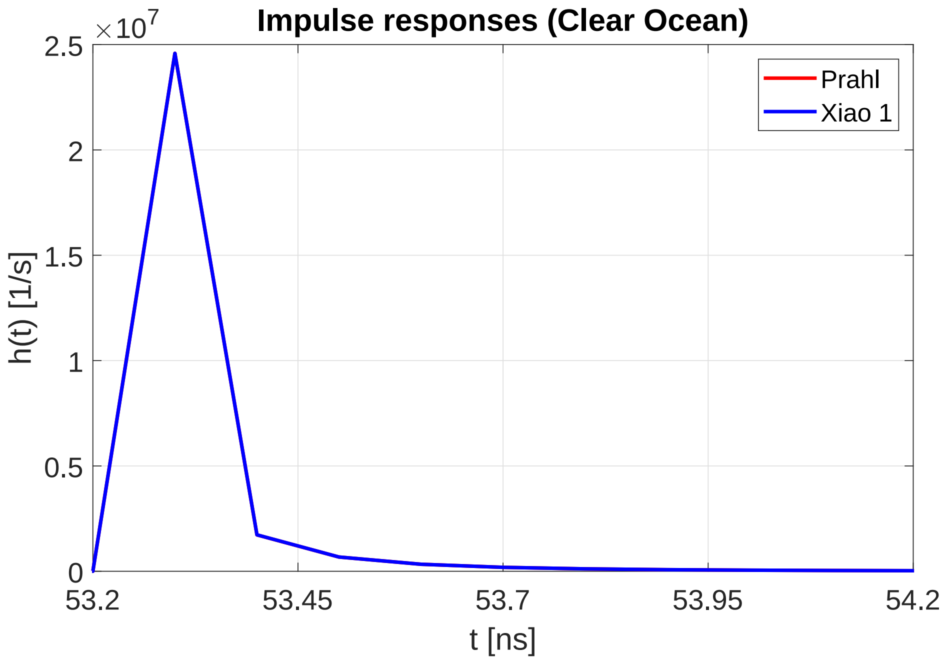


Figure 3.9: CIR obtained with Clear Ocean.

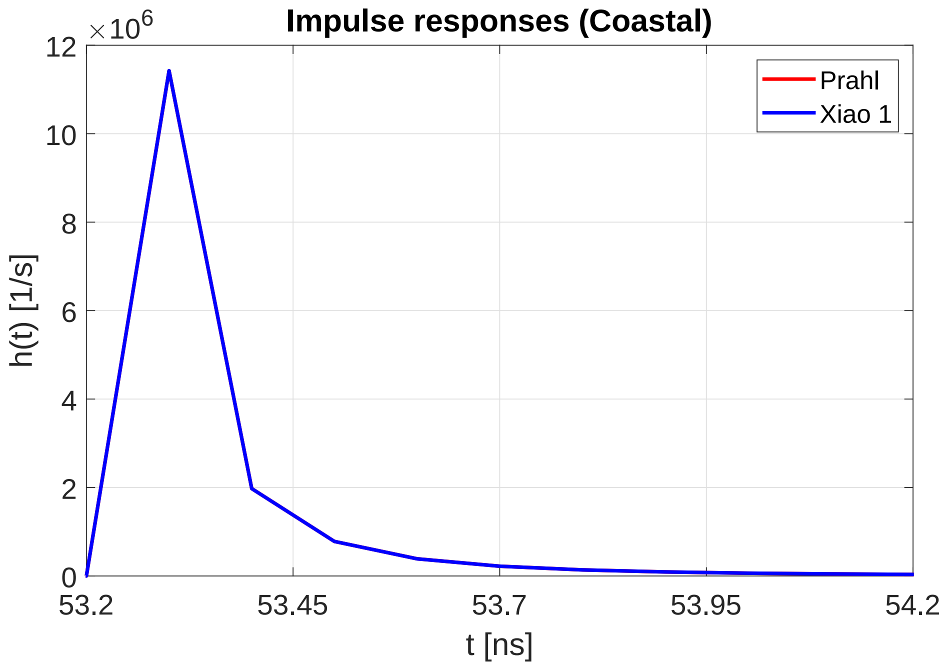


Figure 3.10: CIR obtained with Coastal.

receiver's radius from 0.5cm to 50cm from the initial configuration, we can observe that the channel gain increases due to more light arriving at the larger receiver. Upon close examination, it becomes clear that Xiao 1 algorithm operates with greater efficiency when used with smaller receiver. This can be attributed to the fact that the probability of a path crossing the receiver increases with the apparent size of the receiver, from either the transmitter or any scattering point. Consequently, the impact of Xiao 1 algorithm is reduced when used with larger receivers.

Table 3.5: Unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the receiver's radius (in cm).

Radius (cm)	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
0.5	$1.0003E^{-9}$	$5.5126E^{-11}$	$1.8959E^{-11}$	34.39%
2	$1.5820E^{-8}$	$8.8196E^{-10}$	$4.3945E^{-10}$	49.83%
5	$1.0068E^{-7}$	$5.5023E^{-9}$	$3.2826E^{-9}$	59.66%
8	$2.5842E^{-7}$	$1.4810E^{-8}$	$9.6992E^{-9}$	65.49%
20	$1.6176E^{-6}$	$8.9481E^{-8}$	$7.2641E^{-8}$	81.18%
50	$1.0085E^{-5}$	$5.5510E^{-7}$	$5.2861E^{-7}$	95.23%

This study was also conducted for the other water types, *i.e.* Pure Sea, Clear ocean and Coastal water. The corresponding results are given in Appendices D, E and F. Similar behaviour can be observed considering Clear Ocean and Coastal waters, as evidenced in Table E.1 and Table F.1. However, in the case of Pure Sea, the impact of receiver's area on the efficiency of Xiao 1 algorithm is very small, as shown in Table D.1. There are very small contributions from the multiple scattering links (at most 4 scatterings), so changing the receiver's area mainly affects the LOS link. The receiver's solid angle from T_x to R_x for LOS link for the two algorithms is always the same, indicating that they have the same probability that the last scattering direction $\overrightarrow{\omega_n}$ crosses the receiver regardless of the receiver's area.

The FOV is another parameter that could potentially affect the variance. The impact of FOV on the variance of the Monte Carlo estimator is presented in Table 3.6 for both Prahl and Xiao 1 algorithms. The behaviour is similar to the impact of the area until it reaches 90° of FOV, and then the effect stabilizes. This can be explained by the fact that the VSF β in Harbor water is highly directive, resulting in only a small portion of the radiated power being received beyond a 90° of FOV. Similar behaviour can be observed according to Table D.3, Table E.3 and Table F.3 of Appendices D, E and F, corresponding to pure sea, clear ocean and coastal.

Table 3.6: Unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the receiver's FOV.

FOV	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
20°	$2.1171E^{-7}$	$1.5671E^{-8}$	$8.6771E^{-9}$	55.37%
40°	$6.3446E^{-7}$	$4.2196E^{-8}$	$2.8572E^{-8}$	67.71%
90°	$1.3895E^{-6}$	$7.9408E^{-8}$	$6.5026E^{-8}$	81.89%
120°	$1.5468E^{-6}$	$8.6543E^{-8}$	$6.9997E^{-8}$	80.88%
150°	$1.6080E^{-6}$	$8.8877E^{-8}$	$7.2602E^{-8}$	81.69%
180°	$1.6176E^{-6}$	$8.9481E^{-8}$	$7.2641E^{-8}$	81.18%

3.3.5 Changing distance from T_x to R_x

The distance between the transmitter and the receiver may affect the simulation algorithms' variance. To investigate this, we launch a series of simulations by varying this distance by placing the receiver along the line starting at $(0,0,0)$ and with direction $(1,0,0)$. The corresponding results are presented in Table 3.7. It is worth mentioning that the channel's gain is subject to variation based on the distance. This variation is responsible for the changes observed in sample variance, which is defined to be proportional to the channel's gain, as described in Equation (3.34). It is important to note that these variances are estimate variances and, as a result, are subject to noise due to the Monte Carlo process used to evaluate them. Despite this, the ratio between the variance of Prahl's algorithm and the variance of Xiao 1 algorithm, appears to increase in a roughly linear fashion as the distance between the transmitter and receiver grows. For other 3 water types, we can also observe the similar behaviour, as evidenced in Table D.5, E.5 and F.5 of Appendix D, E and F, respectively.

Table 3.7: Unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the distance between T_x and R_x.

Distance (m)	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
5	$5.6388E^{-4}$	$2.1016E^{-4}$	$1.7154E^{-4}$	81.62%
8	$3.1920E^{-5}$	$5.3291E^{-6}$	$4.3894E^{-6}$	82.37%
12	$1.6176E^{-6}$	$8.9481E^{-8}$	$7.2641E^{-8}$	81.18%
16	$1.1928E^{-7}$	$2.2551E^{-9}$	$1.7600E^{-9}$	78.04%
20	$6.6329E^{-9}$	$5.2579E^{-11}$	$3.9957E^{-11}$	75.99%

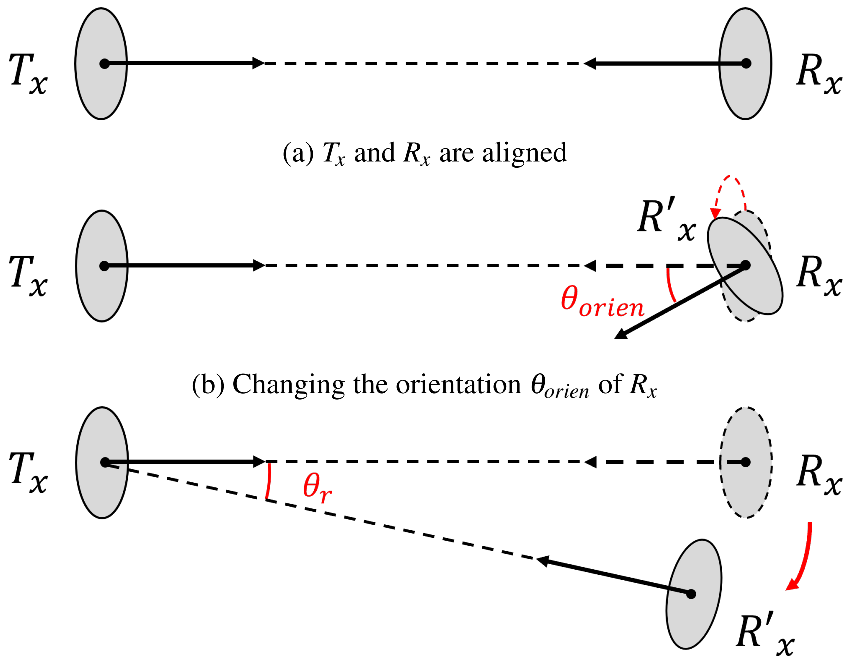
3.3.6 Changing receiver's orientation and relative position

Both the receiver's orientation and relative position may affect the simulation algorithms' variance. In the initial configuration defined in Table 3.1, T_x and R_x are aligned as shown in Figure 3.11a, and the receiver's orientation angle is 0°. Modifying the receiver's orientation θ_{orien} from 0° to 75° following as Figure 3.11b, leads to a reduction in received power due to the decreased receiver's solid angle, as evidenced by Table 3.8. Consequently, the unbiased sample variances are also reduced. These can be also confirmed with results from other 3 water types, as shown in Table D.7, E.7 and F.7 of Appendices D, E and F, respectively.

Table 3.8: Unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the receiver's orientation.

Orientation	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
0°	$1.6176E^{-6}$	$8.9481E^{-8}$	$7.2641E^{-8}$	81.18%
15°	$1.4558E^{-6}$	$8.0583E^{-8}$	$6.4471E^{-8}$	80.01%
30°	$1.4169E^{-6}$	$7.7603E^{-8}$	$6.1512E^{-8}$	79.27%
45°	$1.1846E^{-6}$	$6.4596E^{-8}$	$5.0580E^{-8}$	78.30%
60°	$9.1090E^{-7}$	$4.7250E^{-8}$	$3.8453E^{-8}$	81.38%
75°	$5.8624E^{-7}$	$2.9370E^{-8}$	$2.3132E^{-8}$	78.76%

Table 3.9 shows the results moving the receiver while maintaining a constant distance of 12 meters from the transmitter. Specifically, the receiver is rotated around the



(a) T_x and R_x are aligned

(b) Changing the orientation θ_{orien} of R_x

(c) Rotating R_x around T_x with the angle of θ_r

Figure 3.11: Changing Receiver's orientation and relative position to T_x

transmitter's location by an angle θ_r ranging from 0° to 75°, with its direction always pointing towards the transmitter, as seen from Figure 3.11c. As expected, the channel's gain decreases due to the transmitter emitting primarily in one direction (with 0° orientation) with a narrow beam of 10°. Additionally, it's worth noting the decrease in the Ratio of variances, Xiao 1 algorithm has better performance regardless of receiver's relative position to the transmitter. This conclusion also remains for the other water types, as evidenced in Table D.9, Table E.10 and Table F.10 of Appendices D, E and F, with respect to Pure Sea, Clear Ocean and Coastal.

3.3.7 Discussion

In most studied cases, it appears that Xiao 1 algorithm offers better results in terms of variance and confidence to the result given a fixed computation time. However, it is worth noting that there is one notable exception to this trend. When the receiver's area becomes quite large, the probability of a ray hitting the receiver increases. In such instances, the difference between the two algorithms in terms of sample variance is reduced. These observations underscore the potential benefits of using Xiao 1

Table 3.9: Unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the receiver's relative position.

Rotation angle	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
0°	$1.6176E^{-6}$	$8.9481E^{-8}$	$7.2641E^{-8}$	81.18%
15°	$1.2317E^{-6}$	$5.9962E^{-8}$	$4.8380E^{-8}$	80.68%
30°	$6.7381E^{-7}$	$2.6176E^{-8}$	$2.1308E^{-8}$	81.40%
45°	$3.3588E^{-7}$	$1.0773E^{-8}$	$8.6592E^{-9}$	80.38%
60°	$1.6624E^{-7}$	$4.3766E^{-9}$	$3.7598E^{-9}$	85.91%
75°	$8.6676E^{-8}$	$2.1586E^{-9}$	$1.5937E^{-9}$	73.83%

algorithm, particularly when working with smaller receiver areas or when time constraints are an important consideration.

Chapter 4

MC integration with more efficient IS for UWOC simulations

The mathematical formalization of UWOC channel proposed in Chapter 3 enables the application of variance reduction techniques to devise enhanced algorithms. An algorithm called Xiao 1 was proposed in Chapter 3, with superior performance in terms of sample variance and computation time, compared to the previous Prahl's algorithm. Building on this foundation, this chapter seeks to develop an innovative and highly efficient simulation model for the UWOC channel that further reduces variance and enhances accuracy. Specifically, we propose a new light propagation simulation algorithm, which incorporates efficient importance sampling starting from the integration Equation (3.13) based on the term f_2^n.

This chapter is structured as follows. Firstly, it provides a detailed explanation of the new algorithm and its underlying principles and model. Then it applies this model to 3 different VSFs: Henyey-Greenstein, Two-Term Henyey-Greenstein and Rayleigh. Finally, it conducts a comparative analysis between this new algorithm and Prahl's algorithm and Xiao 1, demonstrating that the proposed algorithm yields more efficient results.

4.1 Formalization of new MC model with more IS

This section puts forward a novel integration model that aims to mitigate variance in numerical simulations. This model is built upon the mathematical formalization of UWOC channel presented in Chapter 3.

In the MC integration with f_1^n presented in Equation (3.8), there exist some ways to apply importance sampling. The integration domains consist of A_{T_x}, n pairs of $\Omega_S \times L$ and A_{R_x}. Hence, importance sampling may be applied for:

- choosing the starting point x_0,

- choosing the emitting direction $\overrightarrow{\omega_0}$,
- choosing the scattering direction $\overrightarrow{\omega_i}$ from scattering point x_i,
- choosing the propagating distance l_i between x_i and x_{i+1}.
- choosing the reception point x_r onto the receiver.

Notably, the *PDF*s of selecting x_0, $\overrightarrow{\omega_0}$ and x_r depend on the transmitter's area, the transmitter's radiation pattern and receiver's area, respectively. Thereby, they can not be chosen using importance from f_1^n presented in Equation (3.10). For n pairs of $\Omega_S \times L$, only $n-1$ scattering directions $(\overrightarrow{\omega_1}, \ldots, \overrightarrow{\omega_{n-1}})$ and n propagating distances $(l_0, \ldots, l_{n-1})$ can be chosen using VSF β and the transmittance T. But the number of β and T consisting in f_1^n are n and $n+1$, indicating the VSF and the transmittance caused by last connection to receiver remain in the MC estimator. As a result, this model is prone to high variance. An algorithm named as Basic algorithm was developed in this chapter based on this model. Its results discussed in Section 4.3.1 are very poor as expected. To circumvent this issue, we eschew the use of this integration model.

Instead, we focus on utilizing the integration model having f_2^n as shown in Equation (3.15). The difference between the expression of f_1^n and f_2^n relies on the last integral: the latter replaces the last integration domain A_{R_x} with Ω_{R_x}. This refinement suggests we may apply importance for choosing the last scattering direction $\overrightarrow{\omega_n}$ emitting from x_n, rather than the reception point x_r.

However, applying importance sampling according to β on f_2^n would require normalizing β on Ω_{R_x} in order to obtain a *PDF*, which is the core problem for this model. This section explains how to solve the VSF normalization problem.

4.1.1 Apply Monte Carlo to f_2^n

Considering the MC integration with f_2^n presented in Equation (3.13), applying importance sampling to last scattering point involves using a part of the last integral over Ω_{R_x} as a PDF. This indicates that it requires transforming β as a PDF for the restricted domain Ω_{R_x}. To achieve this, it needs to normalize β on Ω_{R_x}. The normalization factor NF follows as:

$$NF = \int_{\Omega_{R_x}} \beta\left(\overrightarrow{\omega_{n-1}}, \overrightarrow{\omega_n}\right) \mathrm{d}\overrightarrow{\omega_n} \tag{4.1}$$

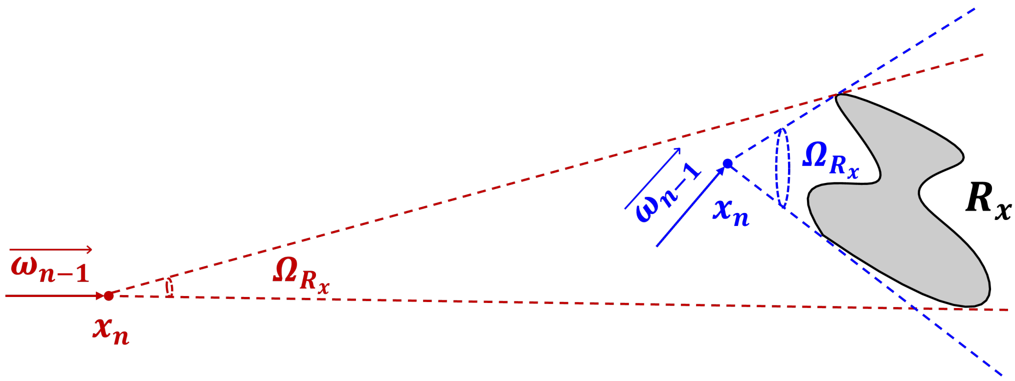


Figure 4.1: Receiver's solid angle Ω_{R_x}

Thus, the *PDF* of choosing the direction $\overrightarrow{\omega_n}$ is $\frac{\beta\left(\overrightarrow{\omega_{n-1}}, \overrightarrow{\omega_n}\right)}{NF}$. The computation of NF necessitates determining the receiver's solid angle Ω_{R_x} seeing from the last scattering point x_n. For example, let's consider a receiver R_x with an irregular shape at fixed position as shown in Figure 4.1. If the point x_n is in different relative positions to R_x, the value of Ω_{R_x} will differ. As such, the computation of the normalization factor poses a significant challenge in the absence of knowledge regarding Ω_{R_x}.

To simplify the problem and make the computations more tractable, we propose to approximate the solid angle of the receiver Ω_{R_x} to the one of its bounding sphere (BS). The bounding sphere of the receiver is defined as the smallest sphere completely enclosing the receiver, regardless of receiver's shape. From any receiver shape, its bounding sphere (denoted as $\mathcal{B}_{\mathcal{S}}$) can be determined by computing the maximum distance between the centre point O and any point on the receiver, and then using that distance as the radius of the bounding sphere, as shown in Figure 4.2.

The solid angle (denoted as $\mathcal{S}_{\mathcal{A}}$) subtended by this bounding sphere seen from the last scattering point x_n is determined as illustrated in Figure 4.3. It is defined by the cone which encloses the entire $\mathcal{B}_{\mathcal{S}}$. This cone extends from x_n to all the farthest points on the surface of the bounding sphere. Its opening angle α is defined as the angle between the line from x_n to O and the line from x_n to M or N, M and N being the couple of points corresponding to the intersection of $\mathcal{B}_{\mathcal{S}}$ and the plane containing $\overrightarrow{\omega_{n-1}}$ and $\overrightarrow{x_n O}$. The last scattering point x_n is viewed as the centre of the coordinate system. Its incoming direction $\overrightarrow{\omega_{n-1}}$ is viewed as the polar axis of the coordinate system. The polar angles of the points O, M, N are denoted as θ_0, θ^- and θ^+, respectively. The intersection

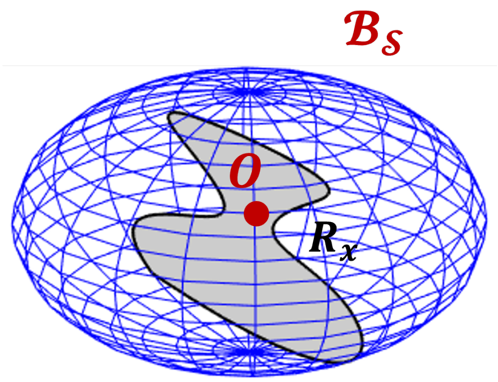


Figure 4.2: The bounding sphere of the receiver

between the solid angle $\mathcal{S}_{\mathcal{A}}$ and a unit sphere $\mathcal{S}$ centring on x_n is the grey cap $\mathcal{C}$ located on $\mathcal{S}$. This means that the final point of any unit propagating direction inside the solid angle $\mathcal{S}_{\mathcal{A}}$ will be located on this grey cap $\mathcal{C}$. In other words, this solid angle $\mathcal{S}_{\mathcal{A}}$ can be determined by computing the bounds of polar angle and azimuth angle of the points on this cap $\mathcal{C}$. In spherical coordinate system, the polar angle ranges from 0 to π. Given the polar angle θ_0 and the opening angle α, θ^- and θ^+ can be computed by the following expression:

$$\begin{aligned} \theta^- &= \max(0, \theta_0 - \alpha) \\ \theta^+ &= \max(\pi, \theta_0 + \alpha) \end{aligned} \tag{4.2}$$

Given the known bounds for the polar angle, our focus shifts to determining the bounds for the azimuth angle. For this purpose, we propose to align the centre line of $\mathcal{S}_{\mathcal{A}}$ with Cartesian axis x. By doing so, we can solely concentrate on computing the upper bound of the azimuth angle, considering that the lower bound is its opposite value. This upper bound can be obtained by computing the azimuth angle φ corresponding to an edge point on the grey cap $\mathcal{C}$ for a given polar angle θ. To achieve this, we proceed in two steps. First, we compute the Cartesian coordinates for this edge point, and subsequently, we derive its polar coordinates.

The intersected point C between the grey cap $\mathcal{C}$ and the line from x_n to O, is the centre point of $\mathcal{C}$, as shown in Figure 4.4. Since C is also situated on the unit sphere $\mathcal{S}$, the vector $\overrightarrow{x_nC}$ has a length of 1. By aligning the centre line from x_n to C with Cartesian axis x, the polar coordinates of the point C is $(\theta_0, 0)$, corresponding to the Cartesian coordinates $(x_0, y_0, z_0) = (\sin\theta_0, 0, \cos\theta_0)$. Knowing the vector $\overrightarrow{x_nC}$ and the unit

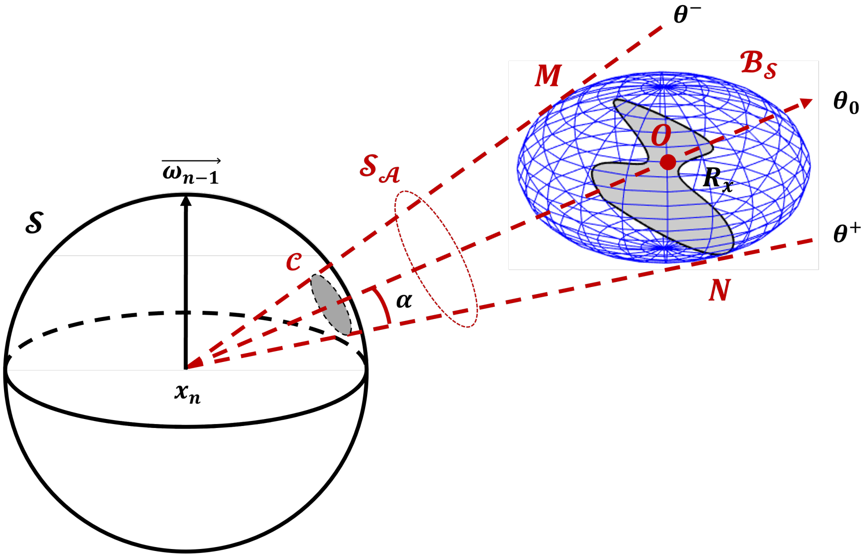


Figure 4.3: The solid angle from the bounding sphere of R_x seeing from x_n

direction $\overrightarrow{\omega_{n-1}}$, the unit direction $\overrightarrow{y}$ of y-axis can be computed by the following expression:

$$\overrightarrow{y} = \overrightarrow{\omega_{n-1}} \times \overrightarrow{x_n C}$$

Then the unit direction $\overrightarrow{x}$ of the x-axis can be computed as:

$$\overrightarrow{x} = \overrightarrow{y} \times \overrightarrow{\omega_{n-1}}$$

The x and y axes are established as shown in Figure 4.4. Let A be an edge point on $\mathcal{C}$ with polar angle θ, corresponding to Cartesian coordinate (x_A, y_A, z_A), as seen in Figure 4.4. Notice that this point is also on the unit sphere $\mathcal{S}$, so we can obtain $z_A = \cos\theta$. However, estimating the remaining two coordinates is not straightforward. In this situation, we can use the following scalar product:

$$\cos\alpha = \overrightarrow{x_n A} \cdot \overrightarrow{x_n C}$$

By using the Cartesian coordinates of A and C, it follows:

$$\cos\alpha = x_A \times \sin\theta_0 + y_A \times 0 + \cos\theta\cos\theta_0$$

The sole remaining unknown is x_A, so obviously it follows:

$$x_A = \frac{\cos\alpha - \cos\theta\cos\theta_0}{\sin\theta_0}.$$

If $\theta_0 = 0$ or $\theta_0 = \pi$, we will use $x_A = y_A = 0$.

The azimuthal angle φ of point A is defined by projecting A onto the plane $\overrightarrow{xx_n}\,\overrightarrow{y}$ (the projected point denoted as $A_\perp$), and then computing the angle between the vector $\overrightarrow{x_nA_\perp}$ with the unit vector $\overrightarrow{\mathbf{x}}$. Excepting for $\theta = 0$ or $\theta = \pi$, the azimuthal angle is defined thanks to the following cosine:

$$\cos\varphi = \frac{\overrightarrow{x_nA_\perp}\cdot\overrightarrow{\mathbf{x}}}{\left\|\overrightarrow{x_nA_\perp}\right\|\,\left\|\overrightarrow{\mathbf{x}}\right\|}$$

Since the Cartesian coordinates of vector $\overrightarrow{\mathbf{x}}$ are $(1,0,0)$, it leads directly to:

$$\cos\varphi = \frac{x_A}{\sqrt{{x_A}^2+{y_A}^2}}$$

This implies to compute y using unit sphere equation, and then to solve the above equation. Considering A is onto the unit sphere $\mathcal{S}$ at height $\cos\theta$, its Cartesian coordinates satisfy the following expression:

$$\begin{array}{rcl} 1 & = & {x_A}^2+{y_A}^2+z^2 \\ \Rightarrow\ {y_A}^2 & = & 1-{x_A}^2-\cos^2\theta \\ \Rightarrow\ {y_A}^2 & = & \sin^2\theta-{x_A}^2 \end{array}$$

Putting that into previous equation, it follows:

$$\begin{array}{rcl} \cos\varphi & = & \frac{x_A}{\sqrt{{x_A}^2+{y_A}^2}} \\ \Rightarrow\ \cos\varphi & = & \frac{x}{\sqrt{{x_A}^2+\sin^2\theta-{x_A}^2}} \\ \Rightarrow\ \cos\varphi & = & \frac{x_A}{|\sin\theta|} \\ \Rightarrow\ \cos\varphi & = & \frac{\cos\alpha-\cos\theta\cos\theta_0}{\sin\theta_0|\sin\theta|} \end{array}$$

This leads to the value of $\varphi \in [0,\pi]$:

$$\varphi(\theta) = \cos^{-1}\left(\frac{\cos\alpha-\cos\theta\cos\theta_0}{\sin\theta_0\,|\sin\theta|}\right) \tag{4.3}$$

Notice that the inverse of the cosine is defined only for $x_A \in [-1,1]$. This indicates that this condition should be verified first! Else, when the condition doesn't lie we should use $\varphi = \pi$. Notably, when $\sin\theta = 0$ occurs, the vector $\overrightarrow{x_nA_\perp}$ is nil. This leads to degeneracies at the poles (north and south). In this case, we consider that $\varphi = 0$. Since the centre of $\mathcal{S}_A$ is in the plane containing $\overrightarrow{\omega_{n-1}x_n}\,\overrightarrow{x}$, the bounds of φ for a given θ are $[-\varphi(\theta),+\varphi(\theta)]$. Notice that these bounds change varying the polar angle θ, *i.e.* $[-\varphi(\theta_1),\varphi(\theta_1)]$ for θ_1, $[-\varphi(\theta_2),\varphi(\theta_2)]$ for θ_2, as shown in Figure 4.4.

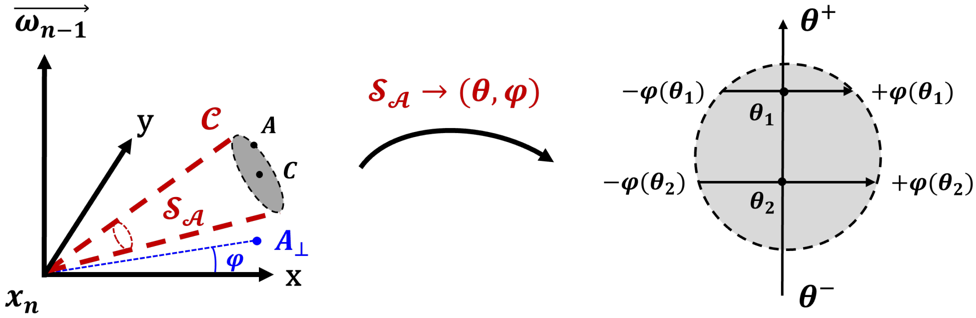


Figure 4.4: From solid angle $\mathcal{S}_{\mathcal{A}}$ to the bounds of polar angle θ and azimuth angle φ

4.1.2 VSF normalization on rectangular domain

Using importance sampling onto the reduced solid angle requires computing the normalization factor NF of the VSF onto this solid angle to obtain a PDF. It equals to the integral of β on the solid angle $\mathcal{S}_{\mathcal{A}}$. As known, the unit propagation direction $d\overrightarrow{\omega_n}$ is defined by the polar angle and azimuth angle. In this context, the differential direction $\overrightarrow{\omega_n}$ can be computed as follows:

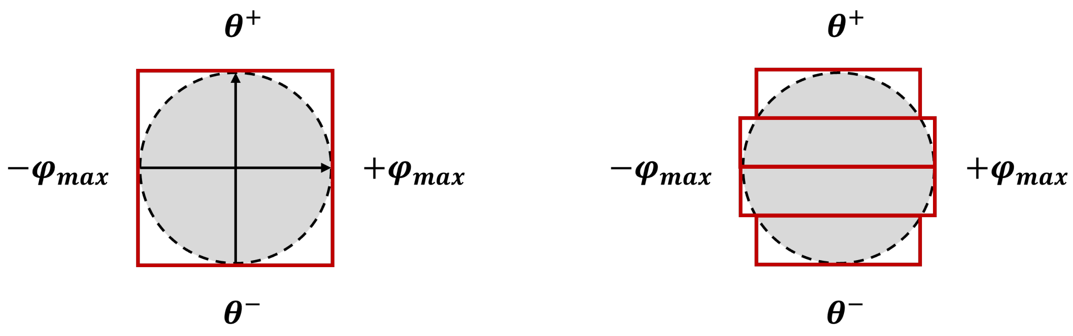
$$d\overrightarrow{\omega_n} = \sin\theta d\theta d\varphi \tag{4.4}$$

According to this, the integral over $\mathcal{S}_{\mathcal{A}}$ can be transformed to a pair of integrals over polar angle θ and azimuth angle φ. So NF can be rewritten as:

$$NF = \int_{\theta^{-}}^{\theta^{+}} \int_{-\varphi(\theta)}^{\varphi(\theta)} \beta(\theta) * \sin(\theta) d\theta d\varphi \tag{4.5}$$

Where θ^{-} and θ^{+} are the lower and upper bound of polar angle θ, referring to Equation (4.2), and the bounds of φ depends on θ with respect to Equation (4.3). In our attempts to compute the normalization factor, we explored the utilization of classic VSFs, such as the Henyey-Greenstein, Two-term Henyey-Greenstein, and Rayleigh models. Unfortunately, these attempts were met with formidable integrals that are difficult to solve, due to the bounds of φ depending on θ.

To get around this problem, an approximate scheme is employed to eliminate this dependency. It consists in modifying the integration domain from the grey cap $\mathcal{C}$ to the smallest rectangle including $\mathcal{C}$, as illustrated in Figure 4.5a. Hence, it removes the dependency to θ in the expression of φ by considering a constant integration domain for the azimuth angle. The height and width of this rectangular depends on θ^{-}, θ^{+}



(a) 1 stripe (b) 4 stripes

Figure 4.5: Stripes domain

and the maximum azimuth angle $\varphi_{\max}$ for over all θ. As θ^- and θ^+ are known, we only need to compute $\varphi_{\max}$. The upper bound of φ for a given θ can be computed by using Equation (4.3). It solely requires finding the maximum value among these upper bounds of φ changing θ from θ^- to θ^+.

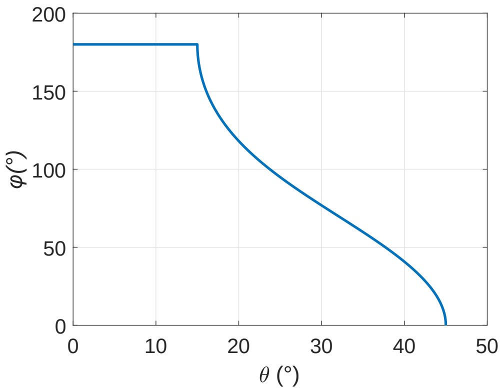
For this purpose, we observe that if the function $\varphi(\theta)$ detailed in Equation (4.3) is monotonic for an interval $[\theta^-,\theta^+]$, then $\varphi_{\max}$ is either $\varphi(\theta^-)$ or $\varphi(\theta^+)$. If $\varphi(\theta)$ is not monotonic, then the same reasoning can be made for each monotonic part resulting from the partition of $[\theta^-,\theta^+]$ with the roots of $\varphi'(\theta)$, the derivative of $\varphi(\theta)$. As verified in Appendix B, the monotonicity of $\varphi(\theta)$ depends on the ratio r, defined as:

$$r = \frac{\cos\theta_0}{\cos\alpha}. \tag{4.6}$$

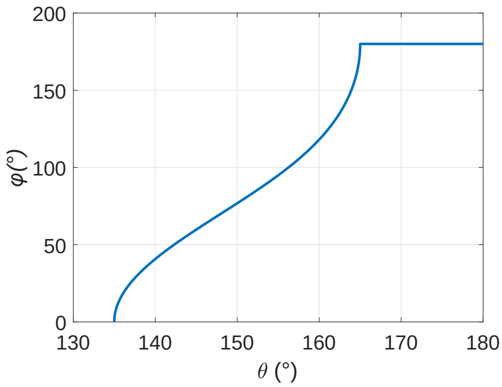
When $|r| > 1$, there is no root for $\varphi'(\theta)$, and its value can be only positive or negative. This indicates that $\varphi(\theta)$ is monotonic decreasing or increasing in the interval $[\theta^-,\theta^+]$, depending on θ_0 and α, as shown in Figure 4.6a and Figure 4.6b. As such, $\varphi_{\max}$ is defined as:

$$\varphi_{\max} = \max\left(\varphi(\theta^-),\varphi(\theta^+)\right) \tag{4.7}$$

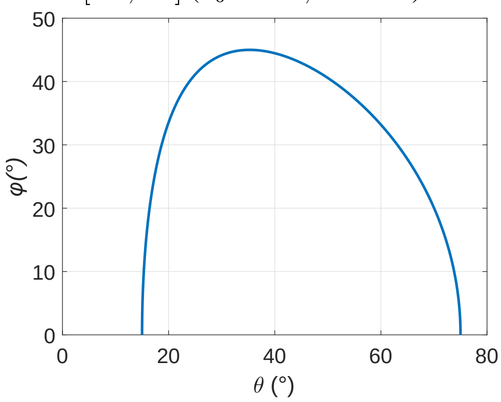
When $r \in [-1,1]$, there is a single root for $\varphi'(\theta)$, which is $\cos^{-1}(r)$. This indicates the upper bound of φ corresponding to this root ($\cos^{-1}(r)$) can be the maximum or minimum value if $\cos^{-1}(r) \in [\theta^-,\theta^+]$, depending on θ_0 and α, as shown in Figure 4.6c and Figure 4.6d. To verify it, we can use the following expression:



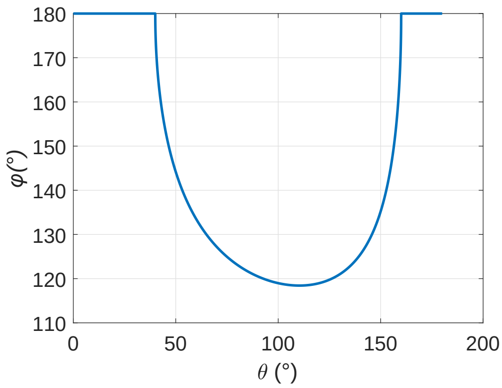
(a) $\varphi(\theta)$: monotonically decrease in $[\theta^-, \theta^+]$ ($\theta_0 = 15°, \alpha = 30°$)



(b) $\varphi(\theta)$: monotonically increase in $[\theta^-, \theta^+]$ ($\theta_0 = 165°$, $\alpha = 30°$)



(c) $\varphi(\theta)$: not monotonic in $[\theta^-, \theta^+]$ ($\theta_0 = 45°$, $\alpha = 30°$)



(d) $\varphi(\theta)$: not monotonic in $[\theta^-, \theta^+]$ ($\theta_0 = 80°$, $\alpha = 120°$)

Figure 4.6: The monotonicity of the function $\varphi(\theta)$ in $[\theta^-, \theta^+]$

$$\varphi_{\max} = \max\left(\varphi(\theta^-), \varphi(\theta^+), \varphi(\cos^{-1}(r))\right) \tag{4.8}$$

Using the rectangular domain, there may exist another problem. If $\theta_0 - \alpha \leq 0$ or $\theta_0 + \alpha \geq \pi$ occurs, it will result in $\theta^- = 0$ or $\theta^+ = \pi$, corresponding to the scattering direction being either directly forward ($\theta = 0$) or backward ($\theta = \pi$). At these pole directions (north and south), the sine term in the integral becomes zero, and the integral evaluates to zero. This leads to the boundary of integral problems. To fix this problem, the solution is to find the minimum of θ in the case of $\theta_0 - \alpha \leq 0$ as lower bound, and maximum of θ with respect to $\theta_0 + \alpha \geq \pi$ as upper bound, denoted as $\theta_{\min}$ and $\theta_{\max}$ respectively following as:

$$\begin{aligned} \theta_{\min} &= \alpha - \theta_0 \\ \theta_{\max} &= 2\pi - \theta_0 - \alpha \end{aligned} \tag{4.9}$$

Using the rectangular domain, the upper and lower bounds for θ and φ become constant, simplifying the computation of the normalization factor. It works for any VSF normalized onto the integration domain, *i.e.* the rectangle here.

Notably, the rectangular domain has a larger area compared to $\mathcal{S}_{\mathcal{A}}$, as evidenced by Figure 4.5a. Therefore, the random choice of the last scattering direction will sometimes lead to a ray which does not intersect the receiver area, corresponding to a nil contribution. This will ultimately result in an increase in the variance of the MC process.

4.1.3 VSF normalization on stripes

To minimize the increase of variance caused by the extended integration domain, the rectangular domain is split into multiple stripes, as shown in Figure 4.5b in the case of 4 stripes. This technique effectively reduces the extended integration domain, achieving an integration domain closer to the exact one, *i.e.* $\mathcal{S}_{\mathcal{A}}$, and so a reduction of the MC process' variance. To do so, first the polar angle interval $[\theta^-, \theta^+]$ is split into N stripes of same sizes, and then the stripes covered by the solid angle S_A are found. Finally, the bounds of θ and φ can be computed for each covered stripe.

We assume here the phase function is subdivided along the polar angle θ into N intervals using regular splitting, where the bounds of each interval are $[a_i, b_i] = [a_i, a_{i+1}]$ with $i \in [0, N-1]$. It leads to $N+1$ values, which are $\{a_0, a_1, \dots, a_N\}$. Instead of splitting the interval $[\theta^-, \theta^+]$, we propose to directly split the entire domain $[0, \pi]$. This solution involves dividing π by the stripe size N, so the a_i is defined as:

$$a_i = \frac{\pi \times i}{N} \tag{4.10}$$

Some stripes outside $\mathcal{S}_{\mathcal{A}}$ need to be disregarded, marked in yellow as seen in Figure 4.7, in order to reduce computation load of NF for a given VSF, and to improve the accuracy of the results. Hence, the stripes intersecting the solid angle $\mathcal{S}_{\mathcal{A}}$ have to be selected among N stripes, and then the maximum number of the selected stripes (denoted as $N_{\max}$) has to be determined. It is known that the bounds of θ corresponding to the rectangular domain are $[\theta^-, \theta^+]$. To find $N_{\max}$, we need to find the indices of the two stripes respectively intersected with θ^- and θ^+, denoted as $lowerIndex$ and $upperIndex$. Then $N_{\max}$ can be obtained as follows:

$$N_{\max} = upperIndex - lowerIndex + 1 \tag{4.11}$$

For this purpose, we can use a dichotomy with the ordered set of values $\{a_0, a_1, \ldots, a_N\}$ for N stripes. The dichotomy is a classic interval search algorithm detailed in Dichotomy. By iteratively dividing the search interval in half and comparing the middle element with the target value θ^- or θ^+, the algorithm narrows down the search space until it finds the desired element $lowerIndex$ or $upperIndex$. This indicates that the complexity of dichotomy algorithm increases as the number of sampling stripes grows. Nevertheless, this complexity is $O(\log_2^N)$ for N values. For instance, if there are 256 intervals, the search will need at most 8 iterations.

Algorithm 3 Dichotomy

```
function DICHOTOMY(start, end, {a_i}, value)
    while start + 1 < end do                    ▷ Search i such that a_i ≤ value < a_{i+1}
        middle ← (start + end)/2
        if a_middle ≤ value then
            start ← middle
        else
            end ← middle
        end if
    end while
    return start                                ▷ Here, start + 1 ≥ end
end function
```

With the knowledge of the indices $lowerIndex$ and $upperIndex$, the stripes covered by S_A are $\{lowerIndex, lowerIndex+1, \ldots, upperIndex-1, upperIndex\}$. As known, the bounds of θ corresponding to the rectangular domain are $[\theta^-, \theta^+]$. So the lower bound of θ for the $lowerIndex$ stripe should be updated as $a_{lowerIndex} = \theta^-$, and the upper bound of θ for the $upperIndex$ stripe should be updated as $b_{upperIndex} = \theta^+$. For the rest covered stripes from $(lowerIndex+1)$ to $(upperIndex-1)$, their bounds for θ remain unchanged. As such, the lower and upper bound of θ for the i-th stripe are updated as:

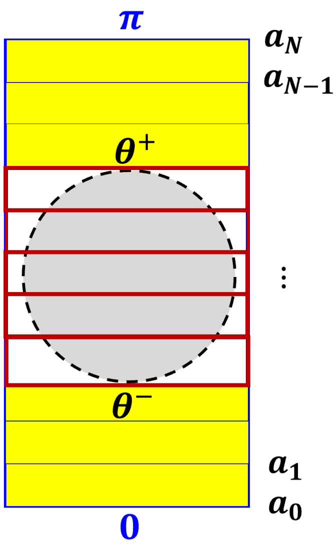


Figure 4.7: Dividing $[0, \pi]$ into N stripes

$$\begin{aligned} a_i &= \max(a_i, \theta^-) \\ b_i &= \min(a_{i+1}, \theta^+) \end{aligned} \tag{4.12}$$

The corresponding maximum upper bound of φ in the interval $[a_i, b_i]$, denoted as $\varphi_{\max_i}$, is computed similarly to $\varphi_{\max}$ for rectangular domain.

4.2 Application to VSFs

Having the bounds of θ and φ for all the stripes covered by the solid angle $\mathcal{S}_{\mathcal{A}}$, we can calculate the normalization factor for each individual stripe, starting from the *lowerIndex* and ending at the *upperIndex*. These individual normalization factors are then accumulated for all the stripes within $\mathcal{S}_{\mathcal{A}}$, to finally obtain the overall normalization factor (NF). This section explores this process using classic VSFs, including HG, TTHG and Rayleigh phase function.

4.2.1 Application to Henyey-Greenstein

Given the HG function as defined in Equation (2.27), the corresponding normalization factor $NF_{(HG_i)}$ for the i-th stripe follows as:

$$\begin{aligned}
NF_{(HG_i)}(g) &= \int_{a_i}^{b_i}\int_{-\varphi_{\max_i}}^{\varphi_{\max_i}} \beta_{HG}(\theta,g)*\sin\theta \mathrm{d}\theta\mathrm{d}\varphi \\
&= \int_{a_i}^{b_i}\int_{-\varphi_{\max_i}}^{\varphi_{\max_i}} \frac{(1-g^2)\sin\theta}{4\pi\left(1+g^2-2g\cos\theta\right)^{\frac{3}{2}}}\mathrm{d}\theta\mathrm{d}\varphi \\
&= \frac{(1-g^2)\varphi_{\max_i}}{2\pi}\int_{a_i}^{b_i} \frac{\sin\theta}{\left(1+g^2-2g\cos\theta\right)^{\frac{3}{2}}}\mathrm{d}\theta \\
&= \frac{(1-g^2)\varphi_{\max_i}}{2\pi g}\left\{\frac{1}{\sqrt{1+g^2-2g\cos(a_i)}}-\frac{1}{\sqrt{1+g^2-2g\cos(b_i)}}\right\}
\end{aligned} \tag{4.13}$$

where $\varphi_{\max_i} = \max\left(\varphi(a_i),\varphi(b_i),\varphi(\cos^{-1} r)\right)$, and r corresponds to Equation (4.6). Notice that $\varphi(a_i)$ $\varphi(b_i)$, $\varphi(\cos^{-1} r)$ can be obtained by Equation (4.3).

By accumulating the individual normalization factors $NF_{HG_i}(g)$ for each stripe, the overall normalization factor $NF_{(HG)}$ for all covered stripes is given by:

$$NF_{(HG)}(g) = \sum_{i=lowerIndex}^{upperIndex} NF_{(HG_i)}(g) \tag{4.14}$$

Knowing this normalization factor for HG phase function, the probability $p(\overrightarrow{\omega_n})$ for choosing the last scattering direction $\overrightarrow{\omega_n}$ inside $\mathcal{S}_{\mathcal{A}}$ can be computed. Notably, this $p(\overrightarrow{\omega_n})$ is the product of the probability of choosing a stripe and the probability of choosing the unit direction $\overrightarrow{\omega_n}$ into this stripe. As such, we first need to find a sampled stripe among $N_{\max}$ covered stripes, and then find the polar angle θ and the azimuth angle φ corresponding to the direction $\overrightarrow{\omega_n}$.

The probability of choosing a stripe covered by $\mathcal{S}_{\mathcal{A}}$ follows cumulative distribution, defined as:

$$p(j) = \frac{CNF_{(HG)}(j)}{NF_{(HG)}} \tag{4.15}$$

where j is the index of a chosen stripe among $N_{\max}$ covered stripes, ranging from 1 to $N_{\max}$, and $NF_{(HG)}$ is the overall normalization factor detailed in Equation (4.14). The cumulative normalization factor $CNF_{(HG)}(j)$ corresponding to this stripe, is computed by accumulating the individual normalization factor from $lowerIndex$ to $(lowerIndex + j - 1)$:

$$CNF_{(HG)}(j) = \sum_{i=lowerIndex}^{lowerIndex+j-1} NF_{(HG_i)}(g) \tag{4.16}$$

According to Equation (4.15), it can be observed that the selection of a stripe covered by $\mathcal{S}_{\mathcal{A}}$ is given with non-uniform density, thanks to the normalization factor importance. To sample a stripe with importance according to the normalization factor, we can use the inversion method applied to $p(j)$. For this purpose, we proceed in the 4 following steps:

1. Compute the $CNF_{(HG)}(j)$ ranging j from 1 to $N_{\max}$ by using Equation (4.16).
2. Set $CNF_{(HG)}(N_{\max}) = NF_{(HG)}$ as the overall normalization factor.
3. Choose a uniform random value $\xi_j \in [0,1]$ and scale it by the total normalization factor NF_{HG} to get the target value.
4. Use the discrete dichotomy search algorithm to find j such that $CNF_{HG}(j)$ is closest to the target value. This involves:
   - Setting the search range to $[1, N_{\max}]$,
   - Finding the midpoint $(middle)$ of the search range,
   - Comparing $CNF_{HG}(middle)$ to the target value $\xi_j NF_{(HG)}$:
     - If $CNF_{(HG)}(middle) \leq \xi_j NF_{(HG)}$, set the lower bound of the search range to $middle$,
     - If $CNF_{(HG)}(middle) > \xi_j NF_{(HG)}$, set the upper bound of the search range to $middle$,
   - Repeating step 4 until the search range contains only one element, which is the selected stripe index j.

Having the sampled stripe index j, we need to sample the last propagation direction $\overrightarrow{\omega_n}$ with importance, according to the polar and azimuthal angles. It consists of using the inversion method applied to the probability of θ and φ. As known, θ and φ are two independent parameters in the HG phase model. To compute the probability of θ corresponding to the chosen stripe, we first need to compute the marginal density function (MDF) of θ by integrating the HG phase function over the azimuth angle φ from $-\varphi_{\max_i}$ to $\varphi_{\max_i}$, following as:

$$\begin{aligned} MDF(\theta) &= \int_{-\varphi_{\max_i}}^{\varphi_{\max_i}} \beta_{HG}(\theta, g) * \sin(\theta)\,\mathrm{d}\varphi \\ &= \int_{-\varphi_{\max_i}}^{\varphi_{\max_i}} \frac{(1-g^2)\sin\theta}{4\pi\left(1+g^2-2g\cos\theta\right)^{\frac{3}{2}}}(d)\varphi \\ &= \frac{(1-g^2)\varphi_{\max_i}\sin(\theta)}{2\pi\left(1+g^2-2g\cos\theta\right)^{\frac{3}{2}}} \end{aligned} \tag{4.17}$$

Notice that i is the index of the stripe among N stripe and j is the index of the covered stripes among $N_{\max}$ covered stripes, such that $i = lowerIndex + j - 1$. Then we can compute the cumulative distribution function (CDF) of θ representing the probability of choosing θ onto this chosen stripe, which is given by:

$$\begin{aligned}
CDF(\theta_i) &= \int_{a_i}^{\theta_i} MDF(\theta)\mathrm{d}\theta \\
&= \int_{a_i}^{\theta_i} \frac{(1-g^2)\varphi_{\max_i}\sin(\theta)}{2\pi\left(1+g^2-2g\cos\theta\right)^{\frac{3}{2}}}\mathrm{d}\theta \\
&= \frac{(1-g^2)\varphi_{\max_i}}{2\pi g}\left\{\frac{1}{\sqrt{1+g^2-2g\cos(a_i)}}-\frac{1}{\sqrt{1+g^2-2g\cos(\theta_i)}}\right\}
\end{aligned} \tag{4.18}$$

Similarly, the MDF of φ for the j-th covered stripe (corresponding to the i-th stripe), can be computed by integrating the HG phase function over θ from a_i to b_i and dividing it by the NF_{HG_i}, following as:

$$\begin{aligned}
MDF(\varphi) &= \frac{\int_{a_i}^{b_i}\beta_{HG}(\theta,g)*\sin(\theta)\mathrm{d}\theta}{NF_{(HG_i)}} \\
&= \frac{\int_{a_i}^{b_i}\frac{(1-g^2)\sin\theta}{4\pi\left(1+g^2-2g\cos\theta\right)^{\frac{3}{2}}}\mathrm{d}\theta}{\frac{(1-g^2)\varphi_{\max_i}}{2\pi g}\left\{\frac{1}{\sqrt{1+g^2-2g\cos(a_i)}}-\frac{1}{\sqrt{1+g^2-2g\cos(b_i)}}\right\}} \\
&= \frac{\frac{(1-g^2)}{4\pi g}\left\{\frac{1}{\sqrt{1+g^2-2g\cos(a_i)}}-\frac{1}{\sqrt{1+g^2-2g\cos(b_i)}}\right\}}{\frac{(1-g^2)\varphi_{\max_i}}{2\pi g}\left\{\frac{1}{\sqrt{1+g^2-2g\cos(a_i)}}-\frac{1}{\sqrt{1+g^2-2g\cos(b_i)}}\right\}} \\
&= \frac{1}{2\varphi_{\max_i}}
\end{aligned} \tag{4.19}$$

The CDF of azimuth angle φ for the i-th stripe representing the probability of choosing φ onto this stripe is computed as:

$$\begin{aligned}
CDF(\varphi_i) &= \int_{-\varphi_{\max_i}}^{\varphi_i} MDF(\varphi)\mathrm{d}\varphi \\
&= \int_{-\varphi_{\max_i}}^{\varphi_i} \frac{1}{2\varphi_{\max_i}}\mathrm{d}\varphi \\
&= \frac{\varphi_i+\varphi_{x\max_i}}{2\varphi_{x\max_i}}
\end{aligned} \tag{4.20}$$

To generate the θ_i and φ_i with respect to the last scattering direction $\overrightarrow{\omega_n}$, we can use 2 random variables ε_{θ_i} and ε_{φ_i} uniformly distributed in the range of $[0,1]$, and then calculate the corresponding θ_i and φ_i by using $CDF(\theta_i)=\varepsilon_{\theta_i}$ and $CDF(\varphi_i)=\varepsilon_{\varphi_i}$. At last, θ_i and φ_i are given by:

$$\begin{aligned}
\theta_i = &\quad \cos^{-1}\left\{\frac{\left(1+g^2\right)\left(\frac{1}{\sqrt{1+g^2-2g\cos(a_i)}}-\frac{2\pi\varepsilon_{\theta_i}}{(1-g^2)\varphi_{\max_i}}\right)^2-1}{2g\left(\frac{1}{\sqrt{1+g^2-2g\cos(a_i)}}-\frac{2\pi\varepsilon_{\theta_i}}{(1-g^2)\varphi_{\max_i}}\right)^2}\right\} \\
\varphi_i = &\quad \left(2\varepsilon_{\varphi_i}-1\right)\varphi_{\max_i}
\end{aligned} \tag{4.21}$$

4.2.2 Application to two-term Henyey-Greenstein

This normalization process is not only suitable for HG phase function, but also applicable for other VSFs, such as Two-Term Henyey-Greenstein (TTHG) and Rayleigh phase functions. TTHG is a modified phase function based on the basis of HG phase function, which can compensate the drawbacks that HG is rather inaccurate for small angles ($< 20°$) and large angles ($> 130°$) [32].

The normalization calculation using TTHG is essentially the same as that for the HG function. Still, we need to compute the normalization factor for each stripe, and choose a stripe random variable ξ_i following uniform distribution in $[0,1]$. The corresponding normalization factor $NF_{(TTHG_i)}$ for i-th stripe follows as:

$$NF_{(TTHG_i)} = \alpha NF_{HG_i}(g_F) + (1-\alpha)NF_{(HG_i)}(-g_B) \tag{4.22}$$

Here, $NF_{(HG_i)}(g_F)$ and $NF_{(HG_i)}(-g_B)$ represent the normalization factors for the HG function with the forward scattering and backward scattering modes, respectively. α is the weight factor that determines the contribution of the forward scattering mode to the normalization factor.

Another random number is required to choose whether the scattering is forward or backward. It is denoted as ζ_i, following uniform distribution in $[0,1]$. To select forward scattering, ζ_i must satisfy a certain condition following as:

$$\zeta_i < \frac{\alpha NF_{(HG_i)}(g_F)}{\alpha NF_{(HG_i)}(g_F) + (1-\alpha)NF_{HG_i}(-g_B)} \tag{4.23}$$

4.2.3 Application to Rayleigh

Rayleigh scattering is an approximation of the light scattering behaviour valid in media composed of particles up to about a tenth of the wavelength [2]. The model is derived by assuming a random distribution of very small specular spheres. The process of normalization with stripes by applying Rayleigh phase function (detailed in Equation (2.33)) is similar to the case of using HG phase function. The only different is the normalization factor $NF_{(Rayleigh_i)}$ for i-th stripe given by:

$$NF_{(Rayleigh_i)} = \frac{\varphi_{\max_i}(15\cos(a_i) - 15\cos(b_i) + \cos(3a_i) - 15\cos(3b_i))}{32\pi} \tag{4.24}$$

4.3 Performance evaluation and discussions

To implement this new MC estimator using enhanced importance sampling for the last scattering direction, we developed new algorithm, denoted as Algorithm Xiao 2. Its implementation is same as Algorithm Xiao 1 except the last connection to the receiver. Algorithm Xiao 1 uses the hit function H to remove the contributions from all extra direction $\overrightarrow{\omega_n}$ not being into Ω_{R_x}, as its integration domain is full direction space Ω_S. As a new contribution, algorithm Xiao 2 directly samples with importance a solid angle higher but close to the receiver's one, aiming to reduce again the variance of the MC process. The objective of this section is to evaluate the performance of the newly developed solution (Algorithm Xiao 2) in comparison to both Prahl's method and Algorithm Xiao 1.

This section begins by evaluating the effectiveness of the new solution with different number of stripes, as compared to the two previous solutions using the Initial configuration. The subsequent part closely resembles that of §3.3 from §3.3.3 to §3.3.7.

4.3.1 Impact of the number of stripes

Algorithm Xiao 2 relies on a user parameter: the number of stripes $NbStrip$. This section aims at determining its effect onto the effectiveness of the Xiao 2 algorithm. To investigate this, we set $NbStrip$ as 2^i with $i \in [0,8]$, requiring at most i iterations to search the stripes intersected with $\mathcal{S}_{\mathcal{A}}$ thanks to the dichotomy algorithm. Table 4.1 presents the impact of $NbStripe$ on both unbiased sample variance and computation time when simulating the Initial configuration with 2 billion sample rays. It can be observed that the unbiased sample variance of MC estimator decreases as $nbStripe$ increases, and more computation time is needed when using more sampling stripes. Notably, Xiao 2 algorithm always provides more accurate results with lower variance regardless of the number of stripes, as compared to that of the other algorithms, including Prahl's algorithm, Basic and Xiao 1 algorithms.

However, the VSF normalization implies more computation time, and so Xiao 2 necessitates more time for a fixed number of samples. To enhance the result, Figure 4.8 presents the expected computation time with Initial configuration varying the target error from $1E^{-9}$ to $9E^{-9}$. Xiao 2 algorithm always cost less time to reach the same target error, especially for $NbStrip = 32$, which is from now considered as default number of stripes and is used for the next sections. Figure 4.9 compares the impulse response generated by Xiao 2 algorithm with 32 stripes to that of Prahl's algorithm and Xiao 1. Xiao 2 algorithm provides more precise results with less noise, as also confirmed by Figure 4.10, which shows the local unbiased variance corresponding to these 3 algorithms.

Table 4.1: Unbiased sample variance and computation time obtained from algorithm Xiao 2 varying $NbStrip$ using initial configuration.

Algorithm	$NbStrip$	Unbiased sample variance	Computation time (hh:mm:ss)
Algorithm Xiao 2	1	$5.6018E^{-8}$	00:14:02
	2	$5.5870E^{-8}$	00:14:04
	4	$5.5141E^{-8}$	00:14:26
	8	$5.4607E^{-8}$	00:14:35
	16	$5.3527E^{-8}$	00:14:51
	32	$5.1130E^{-8}$	00:15:09
	64	$4.9463E^{-8}$	00:16:34
	128	$4.9230E^{-8}$	00:16:38
	256	$4.8582E^{-8}$	00:16:31
Algorithm Basic		$8.9971E^{-8}$	00:13:32
Prahl's Algorithm [1]		$8.9481E^{-8}$	00:13:15
Algorithm Xiao 1		$7.2641E^{-8}$	00:13:52

4.3.2 Using different water types

This section explores the impact of the water type on the variance of Xiao 2 algorithm. Table 4.2 presents the unbiased sample variance considering 4 different water types. The Ratio 1 and Ratio 2 in this table are defined as:

$$\begin{aligned} Ratio1 &= \frac{Var_{(Xiao2)}(C_g)}{Var_{(Prahl)}(C_g)} \\ Ratio2 &= \frac{Var_{(Xiao2)}(C_g)}{Var_{(Xiao1)}(C_g)} \end{aligned} \tag{4.25}$$

where $Var_{(Xiao2)}(C_g)$, $Var_{(Prahl)}(C_g)$ and $Var_{(Xiao1)}(C_g)$ are the unbiased sample variance respectively corresponding to Xiao 2, Prahl and Xiao 1 algorithm, computed

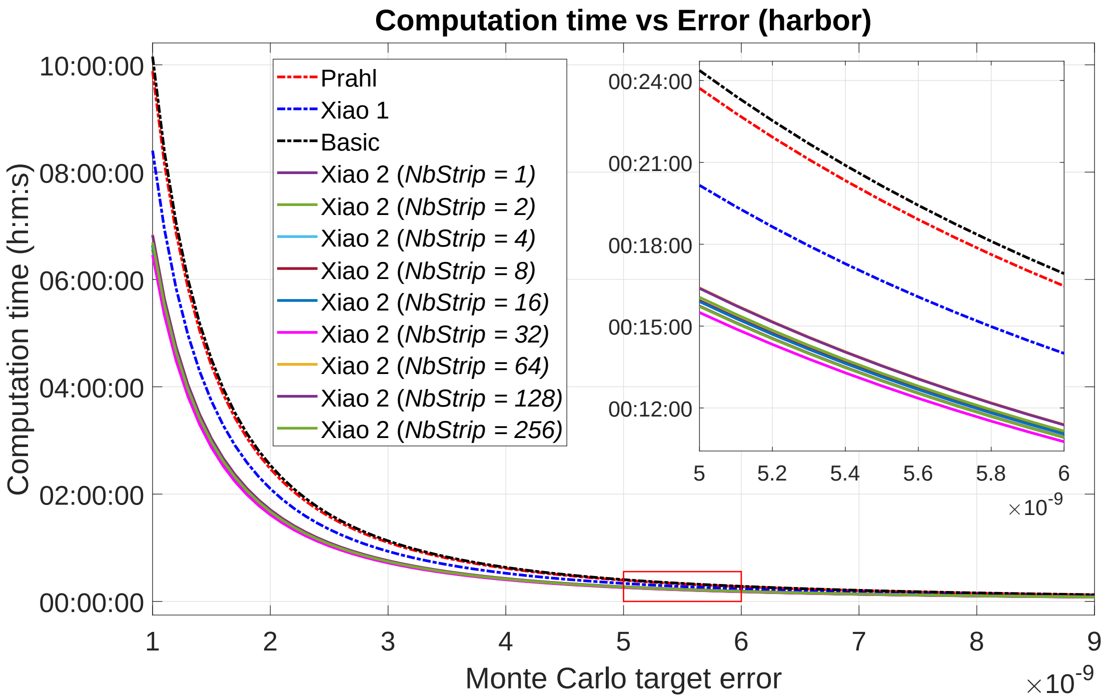


Figure 4.8: Expected computation times with initial configuration varying the target error.

by using Equation (3.34). There is a significant reduction in variance almost for all water types except Harbour water, suggesting the greater efficiency of our new solution. As shown in Figure 4.11, Xiao 2 algorithm needs less time to reach the same target error regardless of water types, and has greater efficiency for less diffuse water. This is especially true for the less diffuse waters with Prahl and Xiao 1 algorithm. This is due to the higher channel gain for less diffuse water as evidenced in Figure 3.7, leading to higher sample error. Hence, Xiao 2 algorithm outperforms the other two, except for Harbour water: less time are cost to reach the minimum target error for less diffuse water considering pure sea, clear ocean and coastal water. Xiao 2 algorithm uses next event estimation technique to connect the scattering points to the receiver, leading to more contributions from LOS link and so providing more accurate channel gain estimation for LOS link. Due to the contributions from LOS link being more significant for less diffuse water, the impact of multiple scatterings is reduced progressively, especially for pure sea.

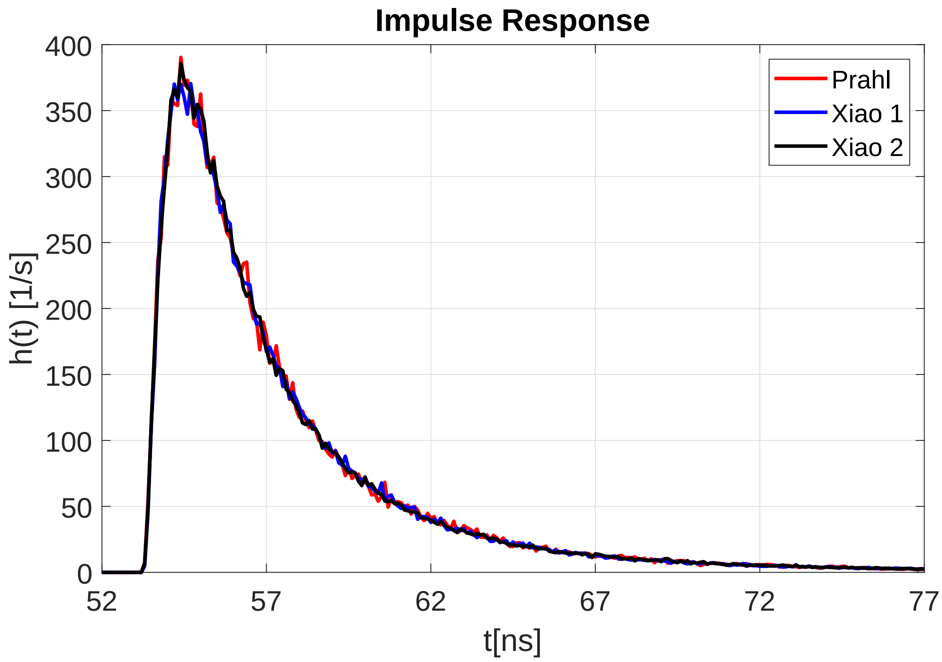


Figure 4.9: Impulse responses obtained with the initial configuration

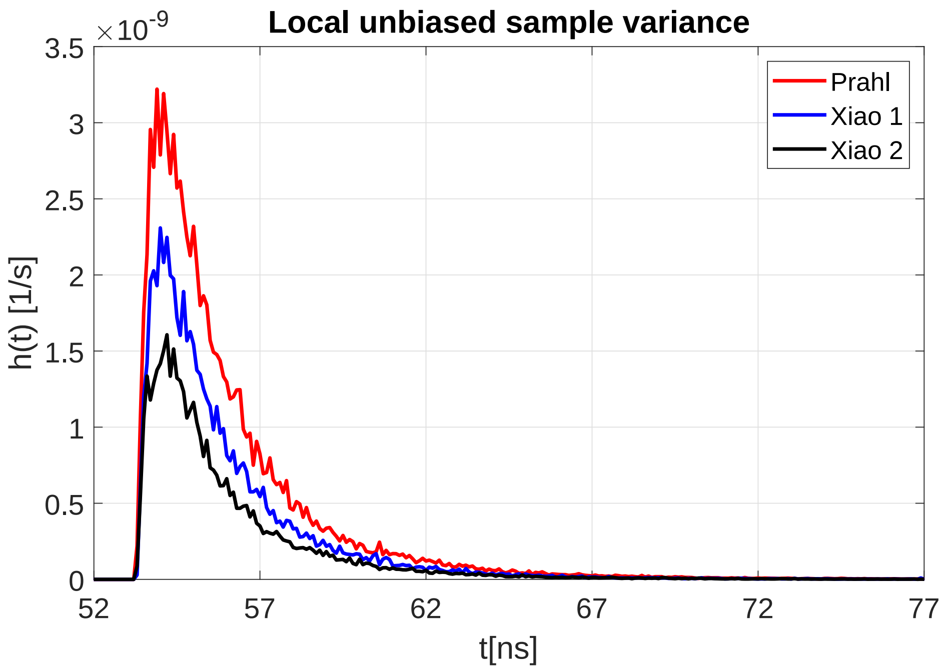
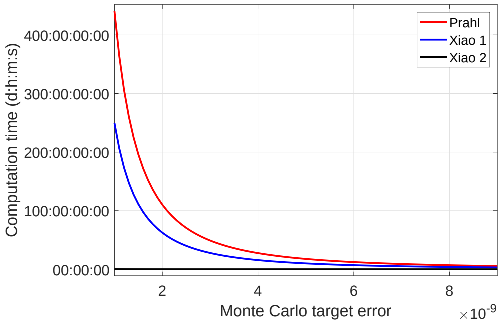


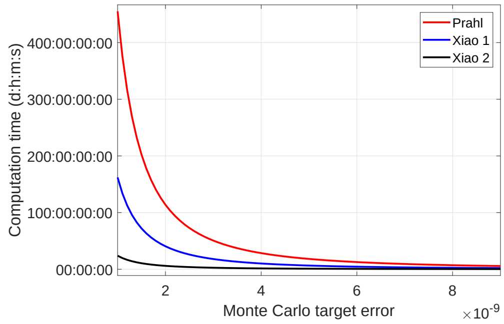
Figure 4.10: Local unbiased sample variance with the initial configuration

Table 4.2: Unbiased sample variance obtained from Algorithm Xiao 2 varying the water type.

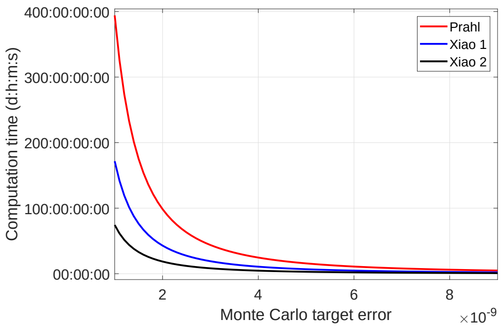
Water type	Algorithm Xiao 2	Ratio 1	Ratio 2
Pure Sea	$5.0532E^{-7}$	0.01%	0.02%
Clear Ocean	$6.1639E^{-5}$	2.93%	9.33%
Coastal	$1.2772E^{-4}$	14.33%	33.25%
Harbour	$5.1130E^{-8}$	57.14%	70.39%



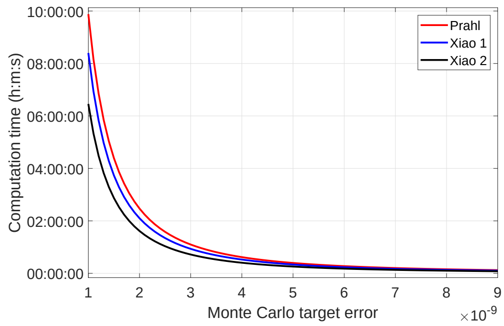
(a) Pure Sea



(b) Clear Ocean



(c) Coastal water



(d) Harbour water

Figure 4.11: Expected computation times varying the target error for different water types with initial configuration.

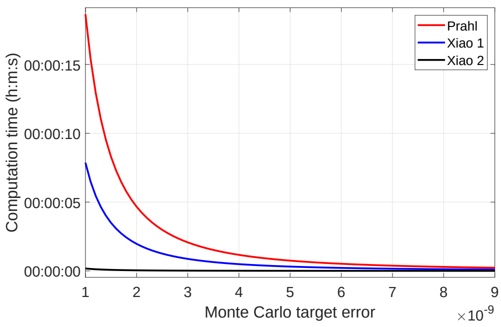
4.3.3 Changing receiver's area and FOV

As known, the probability that a propagating path starting from the transmitter crosses the receiver is dependent on the receiver's size. Table 4.3 showcases the unbiased sample variance obtained from Algorithm Xiao 2, as the receiver's radius varies from 0.5cm to 50cm based on the initial configuration. Clearly, as expected this algorithm exhibits greater accuracy with smaller receivers as compared to the 2 previous algorithms. In particular, for a receiver radius of 0.5cm, its unbiased sample variance has been reduced to 0.64% of Prahl's Algorithm [1] and 1.86% of Algorithm Xiao 1. It's worth noting that a larger receiver implies a larger solid angle Ω_{R_x} enclosing the receiver R_x. Consequently, the probability of a path intersecting with the receiver increases, whether from the transmitter or any scattering point. Therefore, the efficacy of Xiao 2 algorithm diminishes when used with larger receivers. As confirmed with Figure 4.12, Xiao 2 algorithm has greater efficiency with smaller receiver. Notably, the computation time for reaching the minimum target error ($1E^{-9}$) decrease as the receiver's area decreases. This can be explained by the 2 facts:

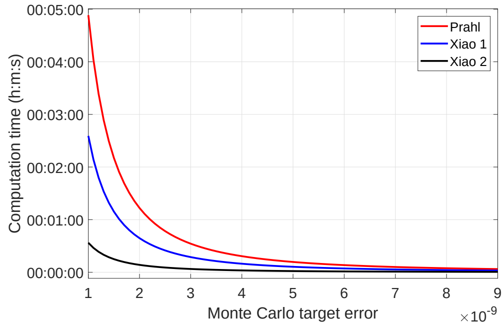
- Due to fewer rays arriving at the receiver with smaller area, the simulation costs less computation time. So the average simulation time per sample ray is shorter for smaller receiver.
- Due to sample error related to the channel gain as detailed in Equation (3.35), fewer rays arriving in the receiver with smaller area results in lower channel gain and so lower error level. Hence, to reach the same target error, the simulation needs less time for smaller receiver.

Besides harbour water, this study also considers other water types, *i.e.* Pure Sea, Clear Ocean and Coastal waters. Their results are presented in Appendix D, E and F. Similar behaviour can be observed for three these other water types, as shown in Tables D.2, E.2 and F.2, corresponding to pure sea, clear ocean and coastal water. Figures D.1, E.1 and F.1 present the expected computation time varying the target error with different receiver's area, respectively for pure sea, clear ocean and coastal water. Xiao 2 algorithm remains greater efficiency regardless of the receiver's area.

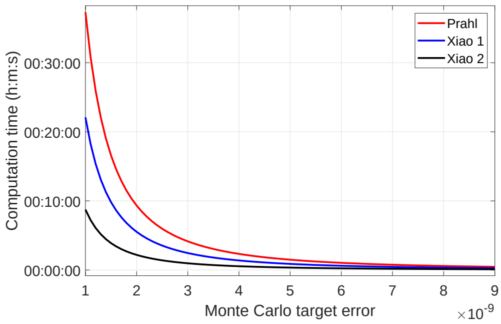
Table 4.4 shows the results obtained from our new solution varying the FOV from 20° to 180°. Interestingly, changing the FOV from 20° to 90°, Xiao 2 algorithm performs more accurately with smaller FOV as compared to Prahl's one. The opposite



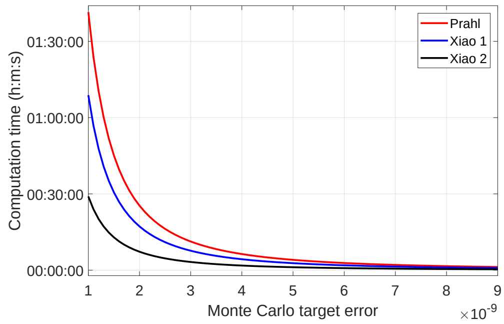
(a) Diameter = 0.5cm



(b) Diameter = 2cm

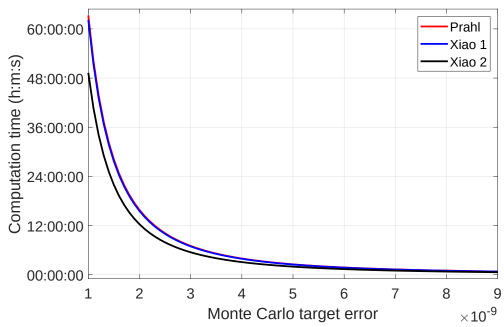


(c) Diameter = 5cm



(d) Diameter = 8cm

(e) Diameter = 20cm



(f) Diameter = 50cm

Figure 4.12: Expected computation times varying the target error for Harbour water with different receiver's area.

Table 4.3: Unbiased sample variances obtained from Algorithm Xiao 2 varying the receiver's radius (in cm).

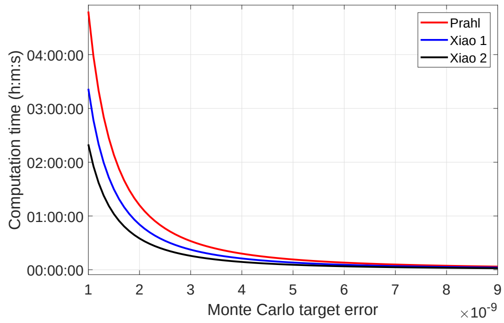
Radius (cm)	Algorithm Xiao 2	Ratio 1	Ratio 2
0.5	$3.5317E^{-13}$	0.64%	1.86%
2	$7.7280E^{-11}$	8.76%	17.59%
5	$1.2803E^{-9}$	23.27%	39.00%
8	$4.3800E^{-9}$	29.57%	45.16%
20	$5.1130E^{-8}$	57.14%	70.39%
50	$4.5445E^{-7}$	81.87%	85.97%

behaviour can be observed in comparison to that of Xiao 1 algorithm. This can be explained by the fact that Xiao 1 algorithm also exhibits better accuracy with smaller FOV, and that the difference between Xiao 1 and Xiao 2 algorithm is very small, as confirmed in Figure 4.13a. As the FOV increases beyond 90°, a stabilization effect can be observed. In fact, there is only a small part of the radiated power being received with a FOV beyond 90°, owing to highly direct HG phase function (β). Nevertheless, Xiao 2 always has greater efficiency comparing to that of the 2 other algorithms, as shown in Figure 4.14. Furthermore, more computation time is needed to reach the minimum target error ($1E^{-9}$) for larger FOV varying FOV from 20° to 90° regardless of simulation algorithms. This is attributed to the fact that more rays arriving in the receiver with larger FOV needs more computation time. Beyond 90○, the efficiency of the 3 algorithms remains stable. We can observe the similar behaviour for pure sea, clear ocean and coastal, as evidenced in Tables D.4, E.4 and F.4 of Appendix D, E and F. Their efficiency are verified in Figures D.2, E.2 and F.2 of Appendix D, E and F: Xiao 2 algorithm provides greater efficiency as compared to that of the 2 other algorithms.

4.3.4 Changing distance from T_x to R_x

The distance between the transmitter and the receiver has a direct impact on the gain of the channel and so affect the variance. Table 4.5 shows the impact of the distance between T_x and R_x on the variance of the Monte Carlo estimator with the new algorithm. It can be observed that Ratio 1 and Ratio 2 seem to increase slightly with an increase in distance. Despite this trend, our new algorithm still remains having better performance, as verified in Figure 4.14. To reach the minimum target error

(a) FOV = 20°



(b) FOV = 40°

(c) FOV = 90°

(d) FOV = 120°

(e) FOV = 150°

(f) FOV = 180°

Figure 4.13: Expected computation times varying the target error for Harbour water with different receiver's FOV.

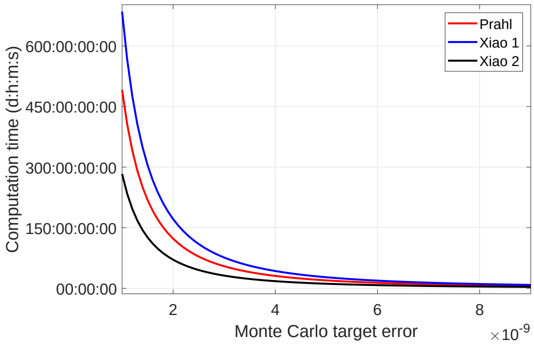
Table 4.4: unbiased sample variances obtained from ALgorithm Xiao 2 varying the receiver's FOV.

FOV	Algorithm Xiao 2	Ratio 1	Ratio 2
20°	$7.1327E^{-9}$	45.51%	82.20%
40°	$2.1503E^{-8}$	50.96%	75.26%
90°	$4.4432E^{-8}$	55.95%	68.33%
120°	$4.9425E^{-8}$	57.11%	70.61%
150°	$5.0394E^{-8}$	56.70%	69.41%
180°	$5.1130E^{-8}$	57.14%	70.39%

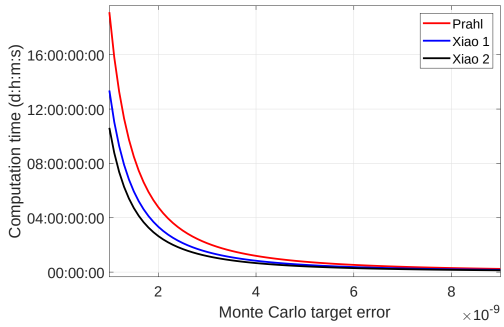
($1E^{-9}$), more computation time is needed for shorter distance. The shorter distance between T_x and R_x indicates less multiple scatterings and so more light reaching the receiver. Thereby, more computation time are needed and the channel gain for shorter distance is higher, leading to higher error level. Similar behaviour can be observed according to the unbiased sample obtained from Xiao 2 algorithm, as illustrated in Tables D.6, E.6 and F.6 of Appendix D, E and F, with respect to pure sea, clear ocean and coastal. The efficiency of the 3 algorithm is verified according to the expected computation time varying the target error with different distance between R_x and T_x for these 3 water types, as seen in Figures D.3, E.3 and F.3 of Appendix D, E and F. As expected, Xiao 2 algorithm exhibits better efficiency for all water types, especially for pure sea.

Table 4.5: unbiased sample variances obtained from ALgorithm Xiao 2 varying the distance between T_x and R_x.

Distance (m)	Algorithm Xiao 2	Ratio 1	Ratio 2
5	$1.2047E^{-4}$	57.32%	70.23%
8	$3.0631E^{-6}$	57.48%	69.78%
12	$5.1130E^{-8}$	57.14%	70.39%
16	$1.3247E^{-9}$	58.74%	75.27%
20	$2.9616E^{-11}$	56.33%	74.12%

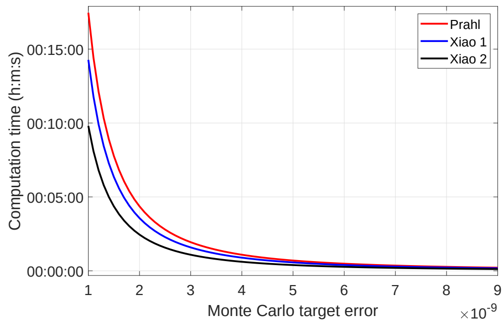


(a) Distance = 5m

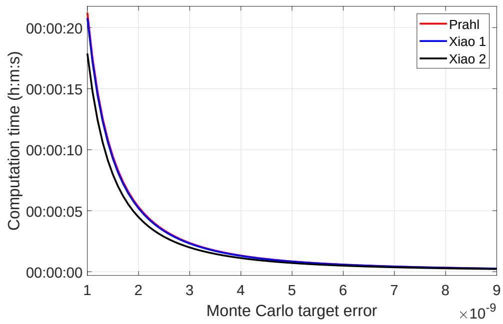


(b) Distance = 8m

(c) Distance = 12m



(d) Distance = 16m



(e) Distance = 20m

Figure 4.14: Expected computation times varying the target error for Harbour water with different distance between R_x and T_x.

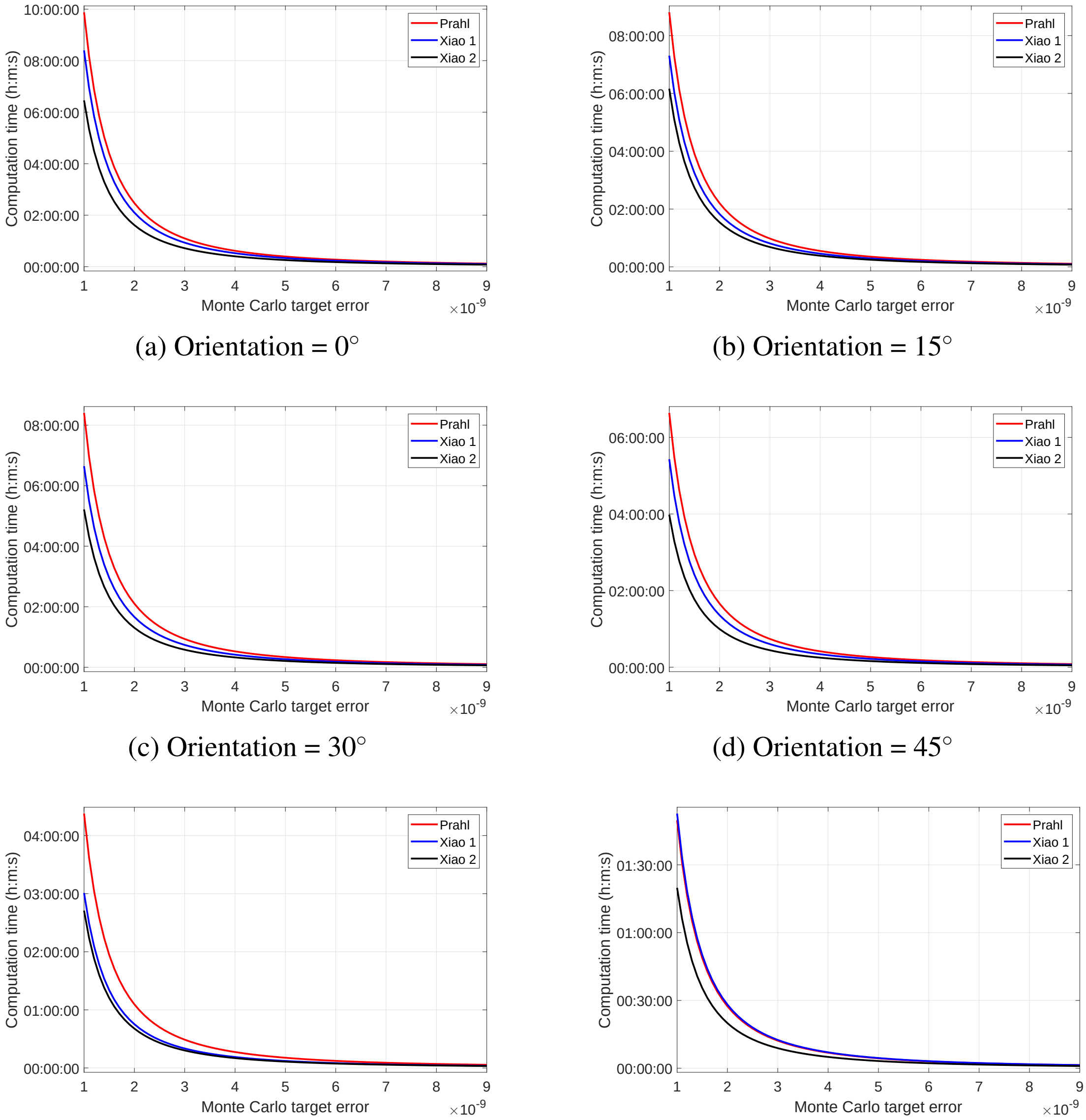
4.3.5 Changing receiver's orientation and relative position

This section studies the impact of receiver's orientation and relative position to T_x on the sample variance of Xiao 2 algorithm. Table 4.6 presents the results obtained from our new simulation algorithm by augmenting the orientation of R_x from 0° to 75°. As receiver's orientation increases, the unbiased sample variance decreases due to decreased received power caused by reduced solid angle of the receiver. Besides, Ratio 1 and Ratio 2 tends to decrease slightly. This indicates the impact of receiver's orientation has very little impact on the efficiency of our new algorithm, as verified in Figure 4.15. Notably, the simulation needs less computation time to reach the minimum target error ($1E^{-9}$) with bigger orientation angle. Orientating the receiver with larger angle has direct impact on the LOS link. In other words, there are less light from T_x directly arriving in the receiver with larger orientation. As a result, the needed computation time is reduced and the error level is lower due to lower received power. We can observe the similar behaviour using pure sea, clear ocean and coastal according to the unbiased sample variance obtained from Xiao 2 algorithm, as evidenced in Tables D.8, E.8 and F.8 of Appendix D, E and F. Their efficiency are verified according to Figures D.4, E.4 and E.4 of Appendix D, E and F. Nevertheless, Xiao 2 algorithm remains greater efficiency comparing to that of the 2 other algorithms.

Table 4.6: Unbiased sample variances obtained from Algorithm Xiao 2 varying the receiver's orientation.

Orientation	Algorithm Xiao 2	Ratio 1	Ratio 2
0°	$5.1130E^{-8}$	57.14%	70.39%
15°	$4.5133E^{-8}$	56.01%	70.01%
30°	$4.5084E^{-8}$	58.10%	73.29%
45°	$3.6649E^{-8}$	56.74%	72.46%
60°	$2.6108E^{-8}$	55.26%	67.90%
75°	$1.5826E^{-8}$	53.89%	68.42%

Table 4.7 shows the impact of receiver's relative position to T_x on the variance of Monte Carlo estimator with Xiao 2 algorithm. The unbiased sample variance decreases with



(a) Orientation = 0°

(b) Orientation = 15°

(c) Orientation = 30°

(d) Orientation = 45°

(e) Orientation = 60°

(f) Orientation = 75°

Figure 4.15: Expected computation times varying the target error for Harbour water with different receiver's orientation.

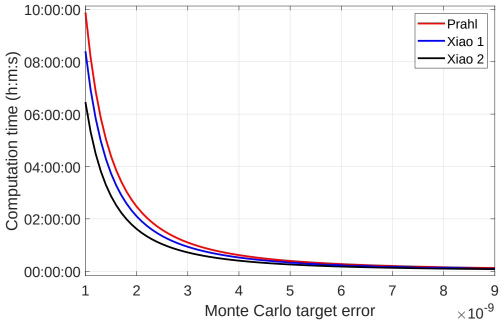
an increase of rotation angle, owing to a decrease in channel gain. Considering that the transmitter radiates mainly in one direction (with 0° orientation) with a small beam (10°), the increase of misalignment level between T_x and R_x leads to less light accepted at the reception. Despite this effect, there is a minor impact on the efficiency of our new algorithm, as evidenced in Figure 4.16. Similar behaviour can be observed as the impact of receiver's orientation. For other water types including pure sea, clear ocean and coastal, similar behaviour to that of harbour water can be observed according to the unbiased sample variance obtained from Xiao 2 algorithm, as illustrated Tables D.10, E.10 and F.10 of Appendix D, E and F. As verified in Figures D.5, E.5 and F.5 of Appendix D, E and F, the conclusion still stands that the Xiao 2 provides more accurate results for a given number of sample paths, and more efficiency for given target error.

Table 4.7: Unbiased sample variances obtained from Algorithm Xiao 2 varying the receiver's relative position for Harbour water.

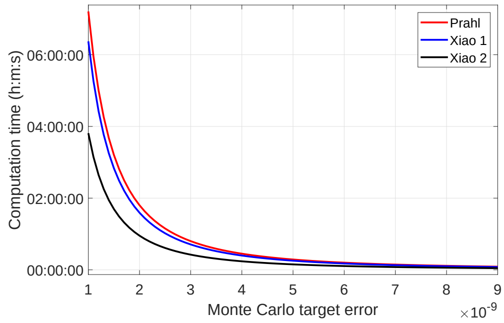
Rotation angle	Algorithm Xiao 2	Ratio 1	Ratio 2
0°	$5.1130E^{-8}$	57.14%	70.39%
15°	$3.4684E^{-8}$	57.84%	71.69%
30°	$1.5062E^{-8}$	57.54%	70.69%
45°	$5.9110E^{-9}$	54.87%	68.26%
60°	$2.5805E^{-9}$	58.96%	68.63%
75°	$1.1548E^{-9}$	53.50%	72.46%

4.3.6 Discussion

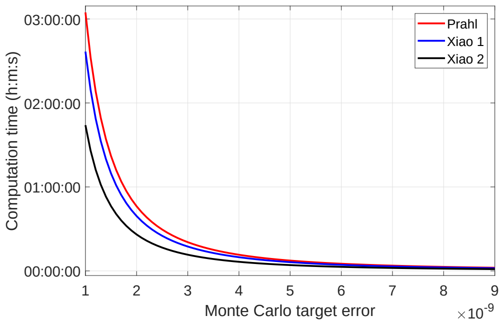
With regard to the effectiveness of the new solution we propose in this chapter, namely the Xiao 2 algorithm, some general conclusions can be drawn based on the results presented in this chapter. Xiao 2 algorithm provides better performance when $NbStrip$ is equal to 32, which is set as the default value for simulating other different configurations. For a given number of sample paths considering all configurations, Xiao 2 algorithm always provides more accurate results due to the lowest variance and so estimation errors, as compared to that of Prahl and Xiao 1 algorithms. At the same time, it exhibits higher efficiency to attain a given target error level, especially for smaller receiver in the environment composed of pure sea water.



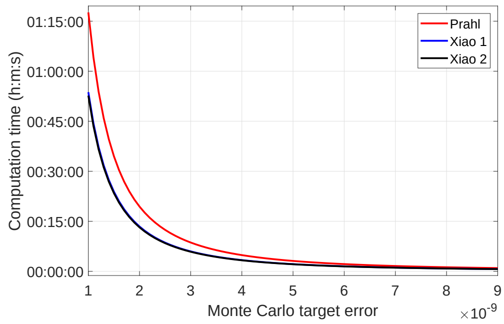
(a) Rotation angle = 0°



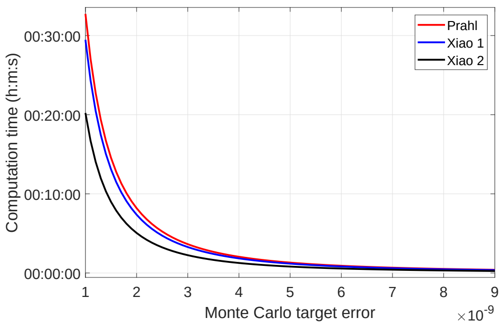
(b) Rotation angle = 15°



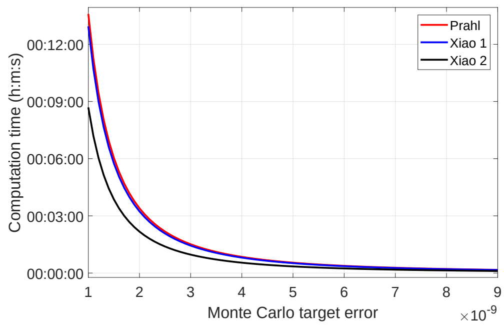
(c) Rotation angle = 30°



(d) Rotation angle = 45°



(e) Rotation angle = 60°



(f) Rotation angle = 75°

Figure 4.16: Expected computation times varying the target error for Harbour water with different receiver's relative position to T_x.

Chapter 5

UWOC algorithms including surface reflection

The previous two chapters have proposed two new UWOC algorithms for simulation of light propagation in participating media, respectively algorithm Xiao 1 and Xiao 2, with enhanced performance in terms of variance reduction and computation time as compared to Prahl's algorithm. They were developed based on the mathematical formalization of UWOC channel proposed in Chapter 3. However, it is important to notice that these algorithms were initially designed under the assumption of an underwater environment devoid of obstacles, made of an infinite volume of water. Consequently, they focus solely on modelling multiple scattering events. In the real underwater environment, *i.e.* undersea, light undergoes not only scatterings but also reflections due to the presence of the sea surface, sea ground and other obstacles as submarine. Therefore, it becomes crucial to consider the influence of surface reflections when designing UWOC simulation algorithms. This means that the formalization of UWOC channel considering an arbitrary sequence of reflection and scattering events is required.

For this purpose, this chapter proposes to generalize path space according to [28, 97], to offer an explicit expression for the UWOC channel as an integral over paths. On this basis, Prahl, Xiao 1 and Xiao 2 algorithms are supplemented with considering the effect of surface reflections, leading to three algorithms denoted as Prahl-R, Xiao 1-R and Xiao 2-R respectively. Finally, extensive simulations are conducted to evaluate the performance of these supplemented algorithms. The obtained simulation results demonstrate that Xiao 2-R algorithm exhibits superior performance compared to both Prahl-R and Xiao 1-R algorithms in terms of computation time for a target variance.

5.1 Generalized path space

This section aims at providing the complete mathematical model to deal with general UWOC channels including both scattering and surface reflections by generalizing path space. It first generalizes the path space within the formalism according to [28], and then derive the directional formalisms for Xiao 2-R, Xiao 1-R and Prahl-R.

5.1.1 Volumetric formalism

Considering both surface reflections and scatterings, a formal way of writing down the light propagation into participating media including reflections through a path integral needs a way to discriminate the different interaction domains: the reflection domain A (a set of surfaces), and the participating media V (a volume). To solve this problem, we introduce a generalized domain noted D_i which is determined by the nature of the point x_i on the path of light propagation, and whose expression is as follows:

$$D_i = \begin{cases} A, & \text{if the interaction point } x_i \text{ is a surface point (reflection event);} \\ V, & \text{if the interaction point } x_i \text{ is a volume point (scattering event).} \end{cases} \tag{5.1}$$

Hence, for n interactions a path $\bar{y}$ is built with points into the domain product $D_1 D_2 \ldots D_n$. To simplify the notation, for each point x_i we introduce a bit c_i defined as follows:

- $c_i = 0$ if the interaction point x_i is a surface point (reflection event);
- $c_i = 1$ if the interaction point x_i is a volume point (scattering event).

These bits can be arranged to form a *path configuration number* $j = (c_n \ldots c_1)_2$ corresponding to a path containing n interactions points, formed with n bits (the subscript 2 indicates a base 2 representation number; moreover we adopt the classical convention of number writing: from right to left). Obviously, j is in $[0 \ldots 2^n - 1]$. Notice that 2^n corresponds to the total number of possible sequences for n interaction points. For instance, when $n = 2$, there are 4 distinct sequences for the 2 interaction points:

- if $j = 0 = (00)_2$, the two interaction points (x_1 and x_2) are surface points;
- if $j = 1 = (01)_2$, the first interaction point (x_1) is volume point and the second one (x_2) is surface point;
- if $j = 2 = (10)_2$, the first interaction point (x_1) is surface point and the second one (x_2) is volume point;
- if $j = 3 = (11)_2$, the two interaction points (x_1 and x_2) are volume points.

From this path configuration number j, the product P_j^n is used to denote a specific sequence of integration domains $D_1 D_2 \ldots D_n$ corresponding to the n successive interactions described in the n bits c_i of j.

Moreover, for a path configuration number j we defined the associated differential form as follows:

$$\mathrm{d}\mu_j^n(x_1,\ldots,x_n) = \prod_{i=1}^{n} \begin{cases} dA\left(x_i\right), & \text{if } c_i = 0 \\ dV\left(x_i\right), & \text{if } c_i = 1 \end{cases} \tag{5.2}$$

Notice that the domain P_j^n corresponds to a specific number j, and so to a specific combination of integration domains A and V. Moreover, considering the number $j = 2^n - 1$, P_j^n is the product of volume domains only; for such a case, our formalism must be equivalent to the one previously proposed in Chapter 3.

Thanks to the Rendering Equation (expressed in Equation (2.21) of Chapter 2) and the Radiative Transfer Equation (expressed in Equation (2.39) of Chapter 2), the received flux $P_n(T_x,R_x,t)$ at R_x and at time t, coming from T_x after n interaction events is modelled as follows:

$$P_n(T_x,R_x,t) = \sum_{j=0}^{2^n-1} \int_{A_{T_x}} \int_{P_j^n} \int_{A_{R_x}} g_0^n(\overline{y^j}) \mathrm{d}x_0 \mathrm{d}\mu_j^n\left(x_1,\ldots,x_n\right) \mathrm{d}x_r \tag{5.3}$$

where A_{T_x} and A_{R_x} are the surfaces of the transmitter T_x and the receiver R_x respectively, and $\overline{y^j} = \{x_0,x_1,\ldots,x_n,x_r\}$ is a propagating path of light starting from x_0 on T_x and reaching x_r on R_x after n interaction events following the path configuration number j. The function $g_0^n(\overline{y^j})$ is the contribution of the path $\overline{y^j}$, including all the propagation phenomena from the transmitter to the receiver, including the radiation, the reception, the reflections and the scatterings. It is defined as:

$$\begin{aligned} g_0^n(\overline{y^j}) = & \frac{L_e\left(x_0,\overrightarrow{\omega_0}\right)\delta\left(t-\tau\right)B_{x_n}\left(x_r\right)V_{is}\left(x_n,x_r\right)T(l_r)B_{x_r}\left(x_n\right)}{l_r^2} \\ & \times \prod_{i=1}^{n} \frac{B_{x_{i-1}}\left(x_i\right)V_{is}\left(x_{i-1},x_i\right)T(l_{i-1})B_{x_i}\left(x_{i-1}\right)f\left(\overrightarrow{\omega_{i-1}},\overrightarrow{\omega_i}\right)}{l_{i-1}^2}, \end{aligned} \tag{5.4}$$

where:

- x_0 is the emitting point onto the transmitter.
- $\overrightarrow{\omega_0}$ is the unit emitting direction from x_0.
- $L_e\left(x_0,\overrightarrow{\omega_0}\right)$ is the emitted flux from x_0 in the direction $\overrightarrow{\omega_0}$.
- $\delta(t-\tau)$ is the delta-Dirac function.
- τ is the propagation delay of path $\overline{y^j}$.

- $B_x(x')$ represents the energy conservation for the light arriving to a point x, depending on its nature c_i, *i.e.* surface or volume; hence, it is given by:

$$B_x(x') = \begin{cases} \left| \overrightarrow{n_x} \cdot \frac{\overrightarrow{xx'}}{\|\overrightarrow{xx'}\|} \right| & \text{if } x \text{ is a surface vertex,} \\ 1 & \text{otherwise.} \end{cases} \tag{5.5}$$

- $\overrightarrow{n_{x_0}} = \overrightarrow{n_{T_x}}$ is the normal vector to transmitter's surface.
- $\overrightarrow{n_{x_i}}$ is the normal vector to the surface interaction point x_i.
- $V_{is}(x_{i-1}, x_i)$ is the visibility function that takes into account the occlusion of the light path by any obstacle (so any surface). It is defined as:

$$V_{is}(x_{i-1}, x_i) = \begin{cases} 1 & \text{if } x_{i-1} \text{ and } x_i \text{ are visible,} \\ 0 & \text{else.} \end{cases} \tag{5.6}$$

- $l_{i-1} = \|\overrightarrow{x_{i-1}x_i}\|$ is the propagation distance between x_{i-1} and x_i.
- $T(l_{i-1})$ is the transmittance over the distance l_{i-1}.
- $\overrightarrow{\omega_i} = \overrightarrow{x_i x_{i+1}}/l_i$ is the unit propagation direction at x_i.
- $f\left(\overrightarrow{\omega_{i-1}}, \overrightarrow{\omega_i}\right)$ is the generalized distribution function, depending on the interaction point x_i; it is given as follows:

$$f\left(\overrightarrow{\omega_{i-1}}, \overrightarrow{\omega_i}\right) = \begin{cases} R\left(\overrightarrow{\omega_{i-1}}, \overrightarrow{\omega_i}\right), & \text{if } c_i = 0 \\ \sigma_s \beta\left(\overrightarrow{\omega_{i-1}}, \overrightarrow{\omega_i}\right), & \text{if } c_i = 1 \end{cases} \tag{5.7}$$

- $R\left(\overrightarrow{\omega_{i-1}}, \overrightarrow{\omega_i}\right)$ is the bidirectional reflectance distribution function between the incoming direction $\overrightarrow{\omega_{i-1}}$ and the outgoing direction $\overrightarrow{\omega_i}$.
- σ_s is the scattering coefficient depending on the participating media.
- $\beta\left(\overrightarrow{\omega_{i-1}}, \overrightarrow{\omega_i}\right)$ is the volume scattering function between the incoming direction $\overrightarrow{\omega_{i-1}}$ and the outgoing direction $\overrightarrow{\omega_i}$.
- x_r is the receiving point onto R_x.
- $l_r = l_n$ is the propagating distance from x_n to x_r on R_x.
- $\overrightarrow{\omega_n} = \overrightarrow{x_n x_r}/l_r$ is the unit propagation direction at x_n.
- $\overrightarrow{n_{x_r}} = \overrightarrow{n_{R_x}}$ is the normal vector at receiver's surface.

This sum of integrals can be numerically computed using Monte Carlo integration [19], leading to the following MC estimator:

$$\widehat{P}_n(T_x, R_x, t) = \sum_{j=0}^{2^n-1} \left\{ \frac{1}{N} \sum_{k=1}^{N} \frac{g_0^n(\overline{y_k^j})}{p(\overline{y_k^j})} \right\}, \tag{5.8}$$

where $\overline{y_k^j}$ is the random variable with respect to a sample path made of n interaction events following the path configuration number j, generated according to the PDF $p(\overline{y_k^j})$, N being the number of samples $\overline{y_k^j}$. Considering that each individual point x_i of $\overline{y_k^j}$ is independently sampled, p is given by:

$$p(\overline{y_k^j}) = p(x_0)p(x_1)\dots p(x_n)p(x_r). \tag{5.9}$$

This estimator can be evaluated by sampling the points on their respective domains:

- the transmitter's surface area for x_0;
- the surface space A or the volume space V for the interaction points from x_1 to x_n following the path configuration number j;
- the receiver's surface area for x_r.

Similarly to the formalization without reflections corresponding to f_0^n described in Chapter 3, it is very difficult else impossible to sample efficiently the path $\overline{y_k^j}$ without rewriting the integral. Then the terms with very high dynamics will remain in the estimator (5.8), more precisely the transmittance, the BRDF and the VSF. Consequently, their high dynamic variations lead to high variance of the MC process, and so to a very poor estimator and to high estimation error. Likewise, this issue can be solved by applying importance sampling onto the transmittance, BRDF and VSF. As such, it requires to transform each surface integral (over the domain A) or each volume integral (over the domain V) into a pair of integrals over the half / full directions' space $\Omega_{S/2}$ (for R) or Ω_S (for β) and the distances' space L.

5.1.2 Directional formalism

To apply importance sampling on direction and distance, each domain D_i appearing into the equation (5.3) needs to be replaced by a couple of domains made of the directions and the distances, like it was done without reflection in Chapter 3. More precisely, each surface integration domain A can be replaced by the half directions' space $\Omega_{S/2}$ and the distance's space L, and each volume integration domain V can be replaced by the full directions' space Ω_S and the same distance's space L. These replacements correspond to the following transformations according to Equations (A.1) and (A.2) of Appendix A:

$$\begin{aligned}
\int_A \mathrm{d}\mu\left(x_i\right) &= \int_{\Omega_{S_{1/2}}} \int_L \frac{l_{i-1}^2}{\left|\overrightarrow{\omega_{i-1}} \cdot \overrightarrow{n_{x_i}}\right|} \mathrm{d}\overrightarrow{\omega_{i-1}} \mathrm{d}l_{i-1} \\
&= \int_{\Omega_{S_{1/2}}} \int_L \frac{l_{i-1}^2}{B_{x_i}\left(x_{i-1}\right)} \mathrm{d}\overrightarrow{\omega_{i-1}} \mathrm{d}l_{i-1}, \qquad (5.10) \\
\int_V \mathrm{d}\mu\left(x_i\right) &= \int_{\Omega_S} \int_L l_{i-1}^2 \mathrm{d}\overrightarrow{\omega_{i-1}} \mathrm{d}l_{i-1},
\end{aligned}$$

where $\overrightarrow{\omega_{i-1}}$ denotes the direction at point x_{i-1} to point x_i, l_{i-1} is the distance between the points x_{i-1} and x_i, and the vector $\overrightarrow{n_{x_i}}$ is the normal vector at the (surface) point x_i. Notice that these two equations replace a domain related to the point x_i by a domain starting from the previous point x_{i-1}.

Hence, each domain D_i is now defined by the following domain:

$$(\Omega L)_{(i-1)} = \begin{cases} \Omega_{S_{1/2}} L & \text{if } c_{i-1} = 0, \\ \Omega_S L & \text{if } c_{i-1} = 1. \end{cases} \qquad (5.11)$$

Notice the specific case for the domain D_1: it is undefined since the bit c_0 is not defined. But x_0 is a point located onto the transmitter, *i.e.* a point on the radiating surface. Hence, the definition of the bits c_i is generalized such that $c_i = 0$ for each $i \leq 1$. Finally, notice also that the bit c_n is not used when the integration problem is rewritten with directions and distances integration spaces.

This new domain can be extended to a product of n domains as $(\Omega L)_j^n = (\Omega L)_1 \dots (\Omega L)_n$. The associated differential form is denoted as $\mathrm{d}\mu_j^n\left(\overrightarrow{\omega_0}, l_0, \dots, \overrightarrow{\omega_{n-1}}, l_{n-1}\right)$. Hence, Equation (5.3) can be rewritten as:

$$P_n(T_x, R_x, t) = \sum_{j=0}^{2^n-1} \int_{A_{T_x}} \int_{(\Omega L)_j^n} \int_{A_{R_x}} g_1^n(\overline{y^j}) \mathrm{d}x_0 \mathrm{d}\mu_j^n\left(\overrightarrow{\omega_0}, l_0, \dots, \overrightarrow{\omega_{n-1}}, l_{n-1}\right) \mathrm{d}x_r, \qquad (5.12)$$

where

$$g_1^n(\overline{y^j}) = g_0^n(\overline{y^j}) \prod_{i=1}^n \frac{l_{i-1}^2}{B_{x_i}(x_{i-1})}. \qquad (5.13)$$

Notice that the products $\frac{l_{i-1}^2}{B_{x_i}(x_{i-1})}$ come from the transformation of the integral domains V to $\Omega_S \times L$ or A to $\Omega_{S/2} \times L$. Hence, g_1^n can be rewritten as follows:

$$g_1^n(\overline{y^j}) = \frac{L_e\left(x_0, \overrightarrow{\omega_0}\right)\delta\left(t-\tau\right)B_{x_n}\left(x_r\right)V_{is}\left(x_n, x_r\right)T\left(l_r\right)B_{x_r}\left(x_n\right)}{l_r^2} \times \prod_{i=1}^{n} B_{x_{i-1}}\left(x_i\right)V_{is}\left(x_{i-1}, x_i\right)T\left(l_{i-1}\right)f\left(\overrightarrow{\omega_{i-1}}, \overrightarrow{\omega_i}\right), \tag{5.14}$$

where each point x_i is derived from the previous one according to Equation (3.11) of Chapter 3.

5.1.3 Formalization of Xiao 2-R algorithm

To further reduce Monte Carlo estimator's variance, the last integration domain A_{R_x} from Equation (5.12) can be replaced by the receiver's solid angle Ω_{R_x} according to the Equation (3.12) of Chapter 3:

$$\int_{A_{R_x}} dx = \int_{\Omega_{R_x}} \frac{l_r^2}{|\overrightarrow{\omega_n} \cdot \overrightarrow{n_{R_x}}|}\,\mathrm{d}\overrightarrow{\omega} = \int_{\Omega_{R_x}} \frac{l_r^2}{B_{x_r}(x_n)}\,\mathrm{d}\overrightarrow{\omega}. \tag{5.15}$$

Hence, the expression of $P_n(T_x, R_x, t)$ can be rewritten as:

$$P_n(T_x, R_x, t) = \sum_{j=0}^{2^n-1} \int_{A_{T_x}} \int_{(\Omega L)_j^n} \int_{\Omega_{R_x}} g_2^n(\overline{y^j})\mathrm{d}x_0\mathrm{d}\mu_j^n\left(\overrightarrow{\omega_0}, l_0, \ldots, \overrightarrow{\omega_{n-1}}, l_{n-1}\right)\mathrm{d}\overrightarrow{\omega_n}, \tag{5.16}$$

where:

$$g_2^n(\overline{y^j}) = g_1^n(\overline{y^j})\frac{l_r^2}{B_{x_r}(x_n)}. \tag{5.17}$$

The function $g_2^n(\overline{y})$ can be simplified as:

$$g_2^n(\overline{y^j}) = L_e\left(x_0, \overrightarrow{\omega_0}\right)\delta\left(t-\tau\right)B_{x_n}\left(x_r\right)V_{is}\left(x_n, x_r\right)T\left(l_r\right) \times \prod_{i=1}^{n} B_{x_{i-1}}\left(x_i\right)V_{is}\left(x_{i-1}, x_i\right)T\left(l_{i-1}\right)f\left(\overrightarrow{\omega_{i-1}}, \overrightarrow{\omega_i}\right). \tag{5.18}$$

Notably, this formalization corresponds to Xiao 2 algorithm supplemented with taking into account the impact of reflections, namely Xiao 2-R algorithm. Its corresponding MC estimator is given by:

$$\widehat{P_n}(T_x, R_x, t) = \frac{1}{N}\sum_{j=0}^{2^n-1}\sum_{k=1}^{N_j}\frac{g_2^n(\overline{y_k^j})}{p(\overline{y_k^j})}, \tag{5.19}$$

where $p(\overline{y_k^j})$ is the probability of the sample path $\overline{y_k^j}$, N_j is the number of samples for the path configuration number j such that $\sum_{j=0}^{2^n-1} N_j = N$. In our algorithm, the probability is chosen as follows:

$$
\begin{aligned}
p\left(\overline{y_k^j}\right) &= p\left(x_0\right) p\left(\overrightarrow{\omega_0}\right) p\left(l_0\right) \ldots p\left(\overrightarrow{\omega_n}\right) \\
&= p\left(x_0\right) p\left(\overrightarrow{\omega_0}\right) \prod_{i=1}^{n} p\left(l_i\right) p\left(\overrightarrow{\omega_i}\right),
\end{aligned}
\tag{5.20}
$$

where:

- $p\left(x_0\right)$ depends on the emitter area,
- $p\left(\overrightarrow{\omega_0}\right)$ depends on the radiation pattern of the emitter,
- $p\left(\overrightarrow{\omega_i}\right)$ depends on the point x_i for $i \in [1, n-1]$:

$$
p(\overrightarrow{\omega_i}) = \begin{cases} \dfrac{R(\overrightarrow{\omega_{i-1}}, \overrightarrow{\omega_i})}{\int_{\Omega_{S/2}} R(\overrightarrow{\omega_{i-1}}, \overrightarrow{\omega_i}) \mathrm{d}\overrightarrow{\omega_i}}, & \text{if } c_i = 0 \ (x_i \in A) \\ \beta(\overrightarrow{\omega_{i-1}}, \overrightarrow{\omega_i}), & \text{if } c_i = 1 \ (x_i \in V) \end{cases}
\tag{5.21}
$$

- $p\left(l_i\right) = \sigma_t T\left(l_i\right)$.

This estimator relies on the choice of number of samples N_j per path configuration number j, which becomes a problem when the number of interactions n is big, says $n > 40$. Obviously, the number of configurations, being equal to 2^n is too huge to be managed in a reasonable amount of computation time. To overcome this problem, the estimator is modified such that the path configuration number j is computed from the sample path $\overline{y_k^j}$, and not the reverse.

It can be observed that the probability to obtain a reflection point x_{i+1} from the probability $p(l_i)$ is 0 in general. So from the starting point x_i, the direction $\overrightarrow{\omega_i}$ and the length l_i, the algorithm builds the point x_{i+1} and then checks the visibility between x_i and x_{i+1}.

- If the two points are visible, then the algorithm considers that x_{i+1} is a scattering point and sets $c_{i+1} = 1$.
- In the other case $V_{is}(x_{i-1}, x_i) = 0$ and so g_2; the algorithm moves x_{i+1} to the closest surface from x_i in the direction $\overrightarrow{\omega_i}$, such that x_{i+1} is a reflection point, and sets $c_{i+1} = 0$.

Let L_i^s be the minimum distance before to span a surface from x_i. In the direction $\overrightarrow{\omega_i}$ from the point x_i the probability to obtain a surface point is $T(L_i^s)$ (it is the probability to sample a point between the distance L_i and ∞). Hence, thanks to the visibility term $V_{is}(x_{i-1}, x_i)$ it is easy to demonstrate that the expected value of the following estimator equals to the Equation (5.16):

$$\widehat{P}_n(T_x, R_x, t) = \frac{1}{N} \sum_{k=1}^{N} \frac{g_2^n(\overline{y_k})}{p(\overline{y_k})}, \tag{5.22}$$

where for each path $\overline{y_k}$ the path configuration number j is now deduced from the nature of the points x_i, $i \in [1 \dots n]$, and not the reverse. In other words, the probability $p(l_i)$ of choosing the distance l_i fully determines the number j.

It is worth noting that, when only considering multiple scatterings Xiao 2 algorithm which corresponds to f_2^n, uses VSF normalization. Actually, this normalization depends on the last scattering point x_n. But when considering both reflections and scatterings in Xiao 2-R algorithm, the nature of the last point x_n can be a reflection point if $c_n = 0$ or a scattering point if $c_n = 1$. Hence, the normalization factor NF must be rewritten as:

$$NF = \begin{cases} \int_{\Omega_{R_x}} R\left(\overrightarrow{\omega_{n-1}}, \overrightarrow{\omega_n}\right) \mathrm{d}\overrightarrow{\omega_n}, & \text{if } c_n = 0 \ (x_n \in A) \\ \int_{\Omega_{R_x}} \beta\left(\overrightarrow{\omega_{n-1}}, \overrightarrow{\omega_n}\right) \mathrm{d}\overrightarrow{\omega_n}, & \text{if } c_n = 1 \ (x_n \in V) \end{cases} \tag{5.23}$$

Notice that both NF and the probability $p(\overrightarrow{\omega_n})$ of choosing last propagation direction $\overrightarrow{\omega_n}$, can be computed similarly as described in Chapter 4.

Assuming there are m surface reflections and $n-m$ scattering events, the MC estimator corresponding to Xiao 2-R algorithm is simplified as follows:

$$\begin{aligned} \widehat{P}_n(T_x, R_x, t) = \frac{1}{N} \sum_{k=1}^{N} & \frac{L_e\left(x_0, \overrightarrow{\omega_0}\right)}{p\left(x_0\right) p\left(\overrightarrow{\omega_0}\right)} \delta(t - \tau_k) T(l_r) NF \\ & \times \prod_{i=1}^{n+1} \left\{ B_{x_{i-1}}(x_i) V_{is}\left(x_{i-1}, x_i\right) \right\} \\ & \times \left(\frac{\sigma_s}{\sigma_t}\right)^{n-m} \prod_{i=1}^{m} \int_{\Omega_{S/2}} R(\overrightarrow{\omega_i'}, \overrightarrow{\omega_i}) \mathrm{d}\overrightarrow{\omega_i}, \end{aligned} \tag{5.24}$$

where the direction $\overrightarrow{\omega_i'}$ and $\overrightarrow{\omega_i}$ are respectively the incoming and outgoing direction for the i^{th} reflection point x_i. The implementation of Xiao 2-R algorithm follows the steps presented in Figure C.1 of Appendix C.

5.1.4 Formalization of Xiao 1-R algorithm

As done in Chapter 3, it is possible to replace the last integration domain Ω_{R_x} by Ω_{c_n}, *i.e.* $\Omega_{S/2}$ or Ω_S depending on the last bit c_n. Then, all the directions $\mathrm{d}\overrightarrow{\omega_n}$ from Ω_{c_n} not being into Ω_{R_x} should be removed. For this purpose, we reuse the function $H(x_n, \overrightarrow{\omega_n})$

previously defined and recalled here:

$$H\left(x_n, \overrightarrow{\omega_n}\right) = \begin{cases} 1, & \text{if } [x_n, \overrightarrow{\omega_n}) \text{ intersects } A_{R_x}, \\ 0, & \text{otherwise,} \end{cases} \tag{5.25}$$

where $[x_n, \overrightarrow{\omega_n})$ denotes the half-line starting at x_n with direction $\overrightarrow{\omega_n}$. The new expression of $P_n(T_x, R_x, t)$ is derived as:

$$P_n(T_x, R_x, t) = \sum_{j=0}^{2^n-1} \int_{A_{T_x}} \int_{(\Omega L)_j^n} \int_{\Omega_{cn}} g_3^n(\overline{y^j}) \mathrm{d}x_0 \mathrm{d}\mu_j^n \left(\overrightarrow{\omega_0}, l_0, \ldots, \overrightarrow{\omega_{n-1}}, l_{n-1}\right) \mathrm{d}\overrightarrow{\omega_n}, \tag{5.26}$$

where $g_3^n(\overline{y^j})$ is defined as follows:

$$\begin{aligned} g_3^n(\overline{y^j}) = {} & L_e\left(x_0, \overrightarrow{\omega_0}\right) \delta\left(t-\tau\right) B_{x_n}\left(x_r\right) V_{is}\left(x_n, x_r\right) T\left(l_r\right) H(x_n, \overrightarrow{\omega_n}) \\ & \times \prod_{i=1}^{n} B_{x_{i-1}}\left(x_i\right) V_{is}\left(x_{i-1}, x_i\right) T\left(l_{i-1}\right) f\left(\overrightarrow{\omega_{i-1}}, \overrightarrow{\omega_i}\right) \end{aligned} \tag{5.27}$$

This formalization corresponds to Xiao 1 algorithm supplemented with considering the impact of reflections, namely Xiao 1-R algorithm. Using a similar sampling strategy than Xiao 2-R (meaning the direction sampling generates the bits of the path configuration number j), the corresponding MC estimator is as follows:

$$\widehat{P_n}(T_x, R_x, t) = \frac{1}{N} \sum_{k=1}^{N} \frac{g_3^n(\overline{y_k})}{p(\overline{y_k})}, \tag{5.28}$$

where $p(\overline{y_k})$ follows as Equation (5.20). The probabilities including $p(x_0)$, $p(\omega_i)$ and $p(l_i)$ for $i \in [0, n-1]$ are computed in the same way as those in the Xiao 2-R algorithm. The only distinction lies in $p(\overrightarrow{\omega_n})$, which is computed according to Equation (5.21) taking into account the binary factor c_n. The MC estimator of algorithm Xiao 1-R (assuming m reflections and $n-m$ scatterings) simplifies as follows:

$$\begin{aligned} \widehat{P_n}(T_x, R_x, t) = \frac{1}{N} \sum_{k=1}^{N} & \frac{L_e\left(x_0, \overrightarrow{\omega_0}\right)}{p\left(x_0\right) \cdot p\left(\overrightarrow{\omega_0}\right)} \delta(t-\tau_k) T\left(l_r\right) H(x_n, \overrightarrow{\omega_n}) \\ & \times \prod_{i=1}^{n+1} \left\{B_{x_{i-1}}(x_i) V_{is}\left(x_{i-1}, x_i\right)\right\} \\ & \times \left(\frac{\sigma_s}{\sigma_t}\right)^{n-m} \prod_{i=1}^{m} \int_{\Omega_{S/2}} R(\overrightarrow{\omega_i'}, \overrightarrow{\omega_i}) \mathrm{d}\overrightarrow{\omega_i}. \end{aligned} \tag{5.29}$$

The implementation of Xiao 1-R algorithm also follow the flowchart as illustrated in Figure C.1 of Appendix C.

5.1.5 Formalization of Prahl-R algorithm

In [1], Prahl used importance sampling in MC simulation for the transmittance, VSF and also BRDF when taking into account the reflections, appearing in Equation (5.27), including for the last segment of $\overline{y^j}$ from x_n to x_{n+1}. The formalization of his former work necessitates $n+1$ integrals over the double domain: the directions' space ($\Omega_{S/2}$ or Ω_S, depending on the interaction point) and the distances' space L. Equation (5.26) is close to this formalization. It only requires to add one more domain L for the distance to obtain this formalism, that is named Prahl-R algorithm in the rest of this chapter.

Prahl uses importance sampling according to the transmittance to sample a distance, producing the point x_{n+1} along the half line starting from x_n in the direction $\overrightarrow{\omega_n}$. Then, he tests whether the line segment $[x_n, x_{n+1}]$ intersects with the receiver. This can be formalized with the following integral:

$$\int_L g(l_n) \times H'(x_n, x_{n+1})\ \mathrm{d}l_n = 1, \tag{5.30}$$

where $g(l_n)$ can be any function that respects the above equation, the hit function $H'(x_n, x_{n+1})$ checks the intersection of the line segment $[x_n, x_{n+1}]$ as follows:

$$H'(x_n, x_{n+1}) = \begin{cases} 1, & \text{if } [x_n, x_{n+1}] \text{ intersects } A_{R_x}, \\ 0, & \text{otherwise.} \end{cases} \tag{5.31}$$

To shorten the formalism, the last two integration domains can be merged to a couple $(\Omega L)_n$ leading the following integration expression of $P_n(T_x, R_x, t)$:

$$P_n(T_x, R_x, t) = \sum_{j=0}^{2^n-1} \int_{A_{T_x}} \int_{(\Omega L)_j^{n+1}} g_4^n(\overline{y^j}) \mathrm{d}x_0 \mathrm{d}\mu_j^{n+1}\left(\overrightarrow{\omega_0}, l_0, \ldots, \overrightarrow{\omega_n}, l_n\right), \tag{5.32}$$

where the function $g_4^n(\overline{y^j})$ is as follows:

$$\begin{aligned} g_4^n(\overline{y^j}) = & L_e\left(x_0, \overrightarrow{\omega_0}\right) \delta\left(t-\tau\right) B_{x_n}\left(x_r\right) V_{is}\left(x_n, x_r\right) T\left(l_r\right) g(l_n) H'(x_n, x_{n+1}) \\ & \times \prod_{i=1}^{n} B_{x_{i-1}}\left(x_i\right) V_{is}\left(x_{i-1}, x_i\right) T\left(l_{i-1}\right) f\left(\overrightarrow{\omega_{i-1}}, \overrightarrow{\omega_i}\right) \end{aligned} \tag{5.33}$$

Then, the MC estimator corresponding to Prahl-R algorithm is derived as:

$$\widehat{P_n}(T_x, R_x, t) = \frac{1}{N} \sum_{k=1}^{N} \frac{g_4^n(\overline{y_k})}{p(\overline{y_k})}, \tag{5.34}$$

The probability $p(\overline{y_k})$ is given by:

$$\begin{aligned} p\left(\overline{y_k}\right) &= p\left(x_0\right) p\left(\overrightarrow{\omega_0}\right) p\left(l_0\right) \ldots p\left(\overrightarrow{\omega_n}\right) p\left(l_n\right) \\ &= p\left(x_0\right) \prod_{i=0}^{n} p\left(\overrightarrow{\omega_i}\right) p\left(l_i\right), \end{aligned} \tag{5.35}$$

The probabilities $p(x_0)$, $p(\overrightarrow{\omega_i})$ and $p(l_i)$ for $i \in [0,n]$ are computed as in Xiao 2-R algorithm, except $p(\overrightarrow{\omega_n})$ which is computed using Equation (5.21).

To complete the formalization of Prahl-R algorithm, $g(l_n)$ should be chosen by using importance sampling according to $p(l_n)$ to the last integration domain L, leading to:

$$g(l_n) = c \cdot p(l_n) = c \cdot \sigma_t T\left(l_n\right), \tag{5.36}$$

where c is the normalization constant introduced to respect $\int_L g(l_n) \times H'\left(x_n, x_{n+1}\right) \mathrm{d}l_n = 1$, computed as in Chapter 3:

$$c = \frac{1}{T\left(l_r\right)}. \tag{5.37}$$

Hence, the expression of $g(l_n)$ is obtained as:

$$g(l_n) = \frac{\sigma_t T\left(l_n\right)}{T\left(l_r\right)}. \tag{5.38}$$

Putting all together, Prahl-R MC estimator simplifies to the following expression:

$$\begin{aligned} \widehat{P_n}(T_x, R_x, t) = \frac{1}{N} \sum_{k=1}^{N} & \frac{L_e\left(x_0, \overrightarrow{\omega_0}\right)}{p\left(x_0\right) \cdot p\left(\overrightarrow{\omega_0}\right)} \delta(t - \tau_k) H'(x_n, x_{n+1}) \\ & \times \prod_{i=1}^{n+1} \left\{ B_{x_{i-1}}(x_i) V_{is}\left(x_{i-1}, x_i\right) \right\} \\ & \times \left(\frac{\sigma_s}{\sigma_t}\right)^{n-m} \prod_{i=1}^{m} \int_{\Omega_{S/2}} R(\overrightarrow{\omega_i}', \overrightarrow{\omega_i}) \mathrm{d}\overrightarrow{\omega_i}. \end{aligned} \tag{5.39}$$

According to [1], Prahl did not utilize next event estimation (NEE) technique for MC simulation of light propagation. This means this algorithm directly checks if it is the last photon after each photon arriving in the receiver, as described in Figure C.2 of Appendix C.

5.2 Performance evaluation and discussions

This section compares the three algorithms formalized in the previous section and considering both scattering and reflections: Prahl-R, Xiao 1-R and Xiao 2-R. More specifically, this section investigates the impact of the reflections due to the presence of surfaces. We start with the Initial configuration, corresponding to sea water with sensors located close to the sea ground, to evaluate the performance of these 3 algorithms with one surface. But a single configuration is not sufficient to draw general conclusions. Furthermore, it's known that the impact of the reflection is contingent upon various factors, including water types, the albedo of the sea ground, and the distance between sensors and the sea ground. So this section studies different configurations by modifying one of these parameters at a time, to further evaluate the effectiveness of these 3 algorithms. Each simulation is conducted using 2 billion sample rays in general.

5.2.1 Performance comparison with Initial configuration including sea ground

Based on the Initial configuration detailed in Table 3.1, we add a sea ground and replace the Lineic transmitter with a surface transmitter, as seen in Figure 5.1. The sensors are positioned above the sea ground at a distance $d = 1$m in order to allow light to reflect the sea ground significantly. The sea ground albedo ρ is set to 0.3 according to [98]. The surface transmitter radius is 10cm and its half power angle $\Phi_{1/2} = 60°$.

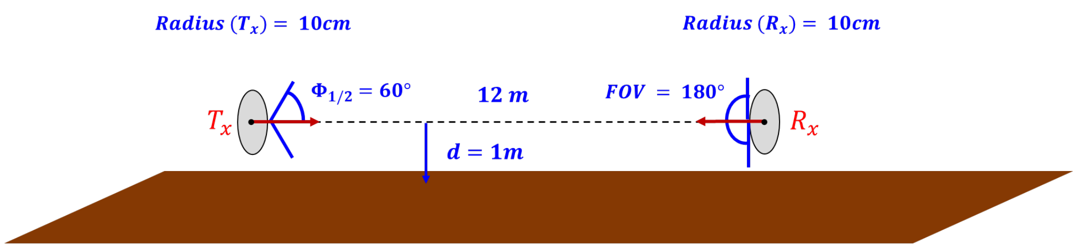


Figure 5.1: Initial configuration with sea ground

To ensure meaningful contributions involving multiple reflections while avoiding unnecessary computational load, it is essential to set an appropriate maximum number of reflections, referred to as $NbReflections$ in the remainder of this chapter. For this purpose, this section first investigates the channel gain with different number of reflections, as shown in Figure 5.2. Notice that setting $NbReflections = 0$ implies that no reflections occurs. It can be observed that the channel gain increases as $NbReflections$ increases until to stabilize after 3 reflections. Besides, there is a sharp

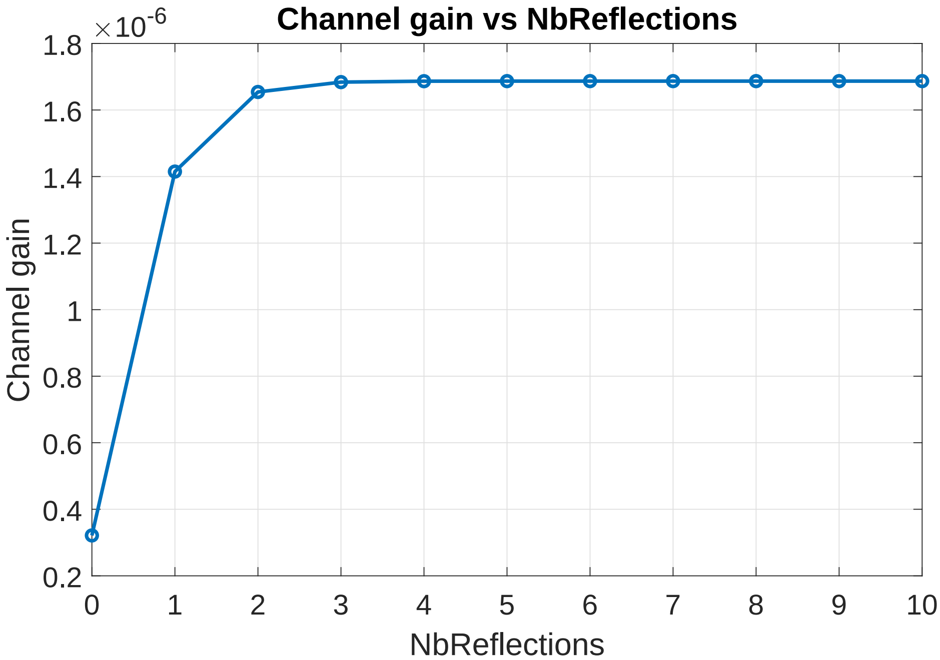


Figure 5.2: Channel gain of Harbor water varying $NbReflections$.

increase from 0 to 1 reflection, indicating the contributions having at least 1 reflection are significant for overall channel gain, as also confirmed by Figure 5.3, which presents the CIR with different $NbReflections$ values. After 3 reflections, there is still very slight increase in channel gain until reaching 5 reflections. To not miss any contributions having any reflections with all the others configurations, $NbReflections$ is set to 10.

Figure 5.4 presents the impulse response obtained by Prahl-R, Xiao 1-R and Xiao 2-R algorithms with Initial configuration including sea ground. Owing to the albedo of sea ground being less than 1, each reflection leads to lose a portion of light, *i.e.* 70% for $\rho = 0.3$; hence, the peak value of CIR decreases. The presence of the sea ground leads to the reflection of all light directed towards it, resulting in less light getting scattered and thereby increasing the overall channel gain. Besides, extended tail of CIR can be observed after 77ns. This can be explained by the fact that when light is reflected, its corresponding path may get extended, resulting in a longer propagation delay. It is important to note that regardless of the used algorithms, more noises are generated due to MC process, indicating more MC sample rays are required in order to achieve more accurate results. This confirms the formalism of the CIR that relies onto the path

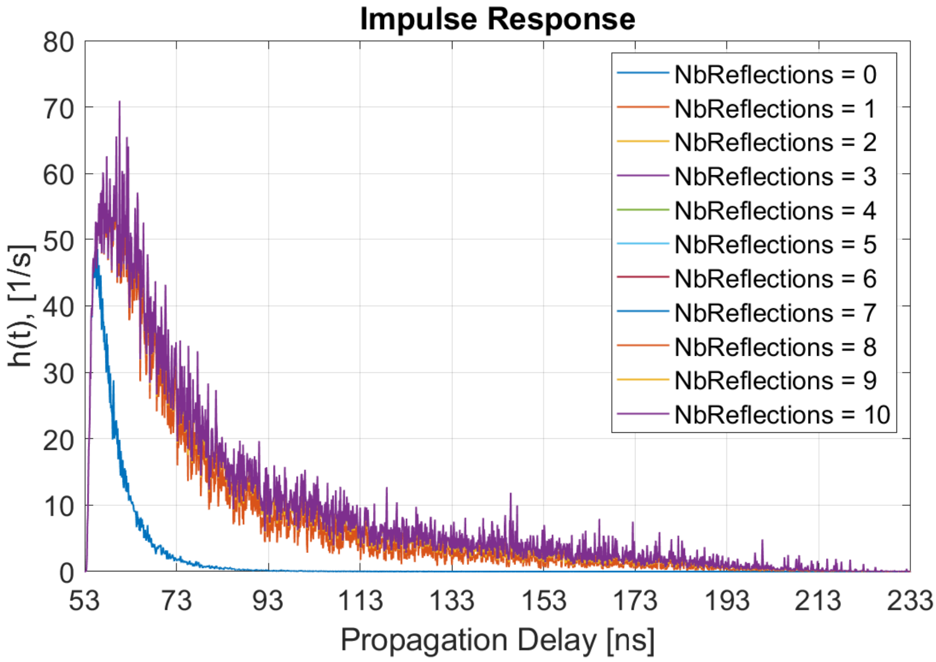


Figure 5.3: CIR with different $NbReflections$.

configuration number j made of n bits: with taking into account reflection, the CIR is formalized as a sum of 2^n integrals.

It can be observed that Xiao 2-R has less noises in comparison to that of the other two algorithms, as also confirmed as in Figure 5.5, which presents the local unbiased sample variance. Clearly, both Xiao 1-R and Xiao 2-R reduce the variance comparing to Prahl-R algorithm, as also evidenced in Table 5.1. As a result, they have lower MC sample errors, resulting in smaller sizes of 95% confidence intervals, which are respectively $4.96E^{-10}$ and $4.02E^{-10}$, as compared to Prahl's one which is $5.50E^{-10}$. However, an increase of around 6 minutes in computation time for Xiao 2-R can be observed. To enhance this result, we present the expected computation time with Initial configuration including sea ground by varying the target error from $1E^{-9}$ to $9E^{-9}$, as shown in Figure 5.6. Prahl's algorithm requires almost 3 days to reach the minimum target error, while Xiao 1-R completes the task in 2 days and 5 hours, and Xiao 2-R achieves it in less than 2 days, significantly reducing the overall computation time.

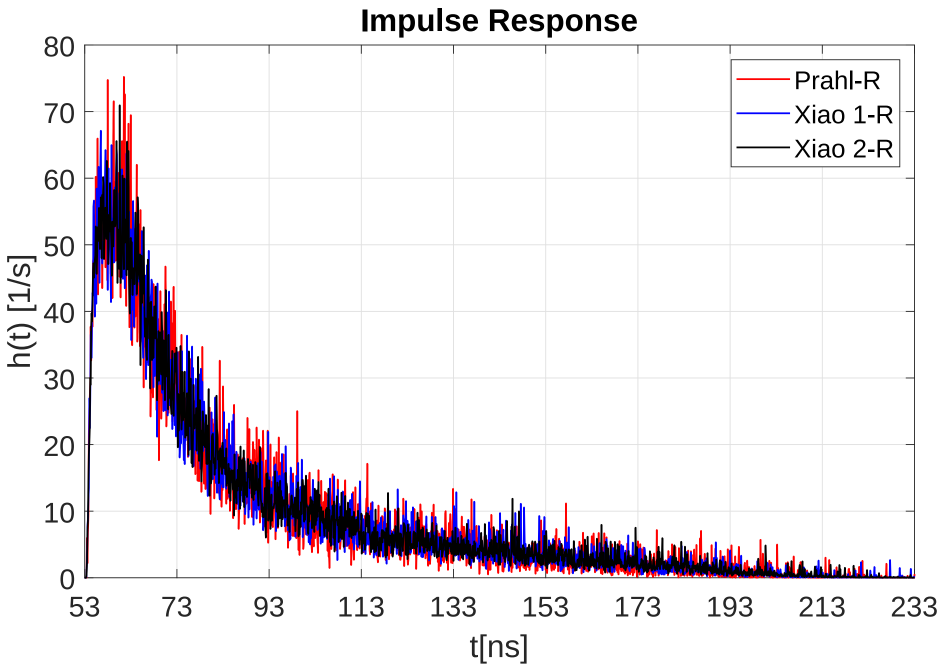


Figure 5.4: CIR obtained with initial configuration including sea ground.

Table 5.1: Simulation Results for Initial configuration including sea ground using Prahl-R, Xiao 1-R and Xiao 2-R algorithms.

	Prahl-R	Xiao 1-R	Xiao 2-R
Channel Gain	$1.6854E^{-6}$	$1.6887E^{-6}$	$1.6869E^{-6}$
Unbiased sample variance	$3.9235E^{-7}$	$3.1918E^{-7}$	$2.0978E^{-7}$
Sample error	$1.4006E^{-8}$	$1.2633E^{-8}$	$1.0242E^{-8}$
Confidence interval with a 95% confidence level	$[1.6579E^{-6}, 1.7129E^{-6}]$	$[1.6639E^{-6}, 1.7135E^{-6}]$	$[1.6668E^{-6}, 1.7070E^{-6}]$
Computation time (h:m:s)	00:20:49	00:20:10	00:26:59

5.2.2 Using different water types

This section investigates the behaviours of the three simulation algorithms considering different water types with the Initial configuration including sea ground. Their corresponding results are presented in Table 5.2. It can be observed that less

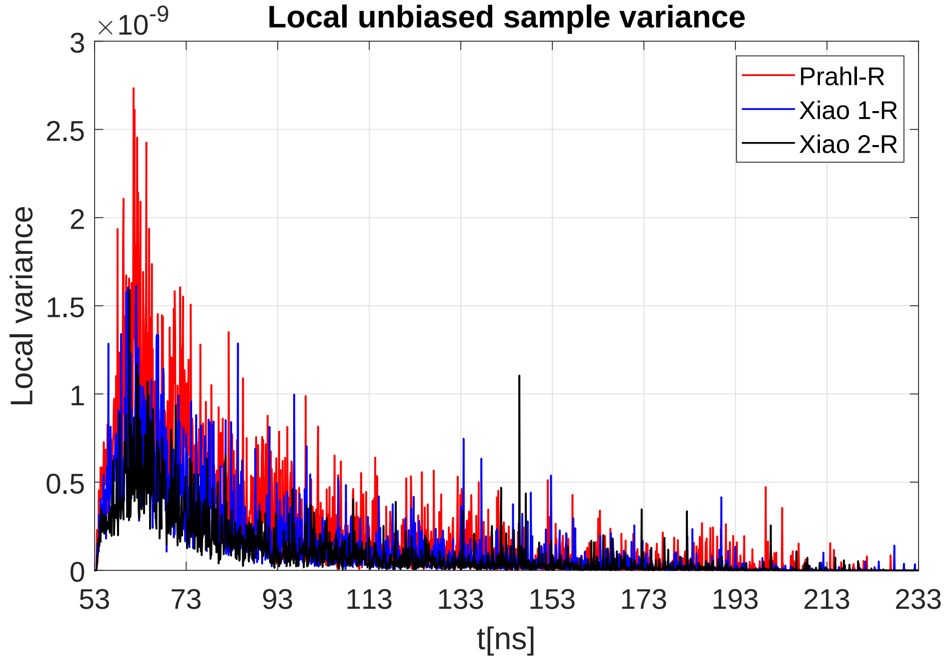


Figure 5.5: Local unbiased sample variances obtained with initial configuration including sea ground.

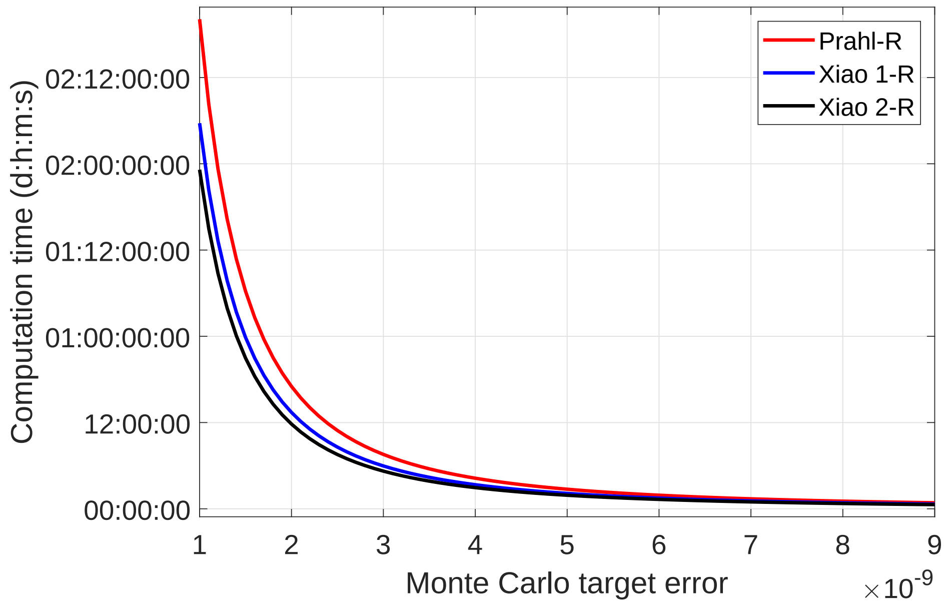


Figure 5.6: Expected computation time varying the target error with Initial configuration including sea ground.

diffuse water has higher channel gain. It holds true regardless of $NbReflections$, as evidenced in Figure 5.7, which displays the channel gain with different $NbReflections$. This is because less light get scattered for less diffuse water. Notably, the contributions having at least 1 reflection are more significant for more diffuse water, especially for Harbor water. This can be explained by the increased occurrence of scattering events in more diffuse water, leading to a higher probability of light being deviated from direct path and so being reflected. But still whatever the water type is, both Xiao 1-R and Xiao 2-R algorithm exhibits superior performance with lower unbiased sample variance, as compared to Prahl-R algorithm. The difference is also illustrated in Figure 5.8-5.10: most of the light flux is received in less than 3 ns for Pure Sea, Clear Ocean and Coastal waters, as compared to the 180 ns shown in Figure 5.4. For the same level of accuracy, they still have greater efficiency, especially Xiao 2-R, as seen in Figure 5.11.

Table 5.2: Unbiased sample variance obtained from Prahl-R, Xiao 1-R and Xiao 2-R algorithms varying the water type.

Water type	Channel gain	Prahl-R	Xiao 1-R	Xiao 2-R
Pure Sea	$3.8488E^{-5}$	$7.0970E^{-5}$	$3.7141E^{-5}$	$8.5898E^{-9}$
Clear Ocean	$3.0294E^{-5}$	$3.7740E^{-5}$	$1.3165E^{-5}$	$1.1373E^{-6}$
Coastal	$2.3147E^{-5}$	$2.1186E^{-5}$	$9.7336E^{-6}$	$3.0439E^{-6}$
Harbor	$1.6869E^{-6}$	$3.9235E^{-7}$	$3.1918E^{-7}$	$2.0978E^{-7}$

5.2.3 Changing the albedo of sea ground

The albedo of sea ground directly determines how much light of underlying surface get reflected. Changing this value from 0.1 to 0.9, causes an increase of channel gain from $7.0245E^{-7}$ to $6.9007E^{-6}$, as shown in Table 5.3. This is also supported by the Figure 5.12, presenting the channel gain with different $NbReflections$ values. Obviously there is no difference in channel gain with different albedo when $NbReflections = 0$ (it is only MC noise). As the albedo grows, the contributions with at least 1 reflection become significant for overall channel gains. As a result, the corresponding unbiased sample variances are also increased.

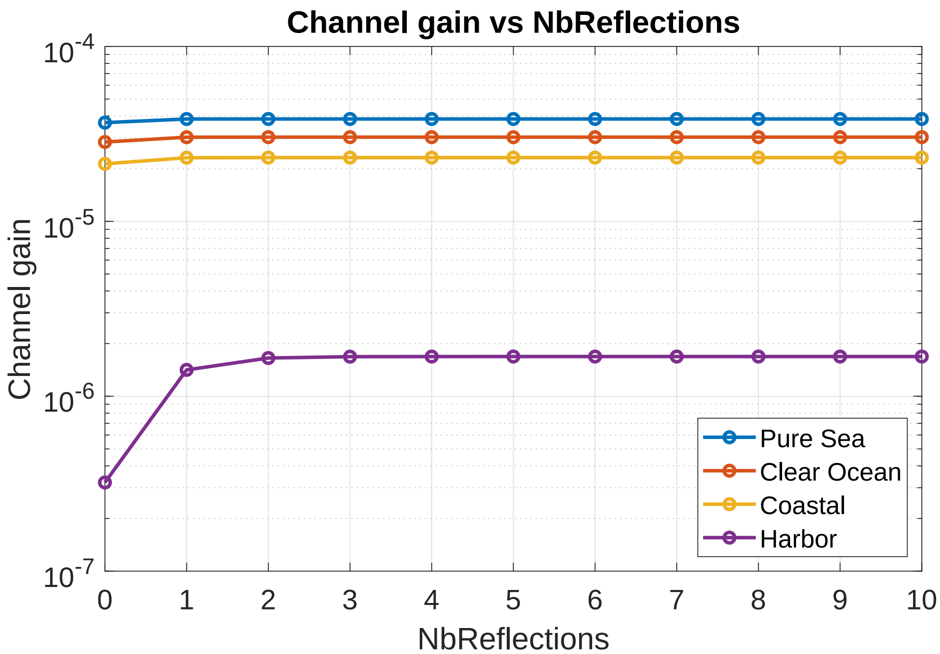


Figure 5.7: Channel gain of different water types varying $NbReflections$.

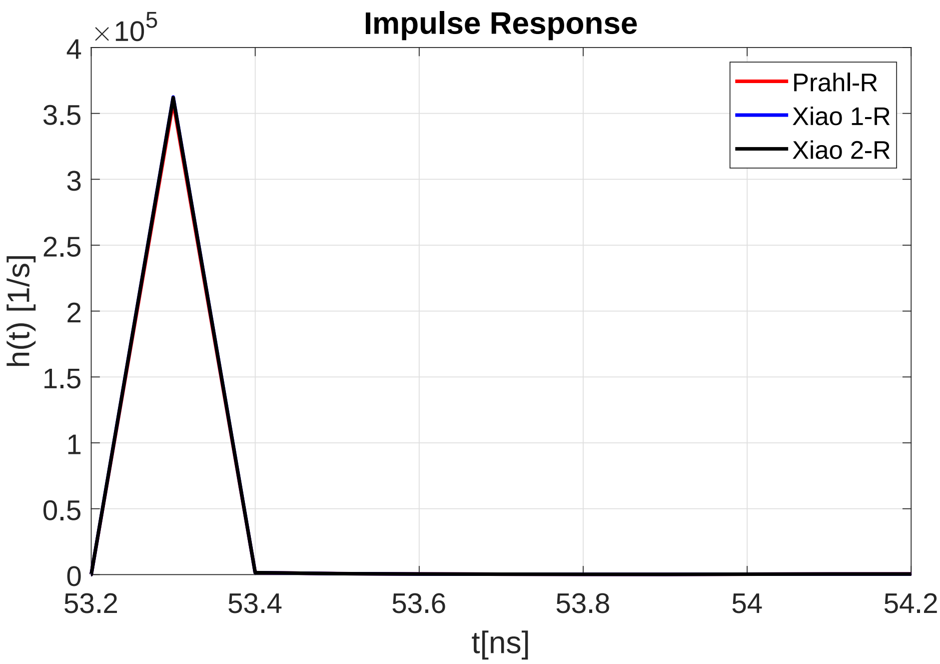


Figure 5.8: CIR obtained with Pure Sea.

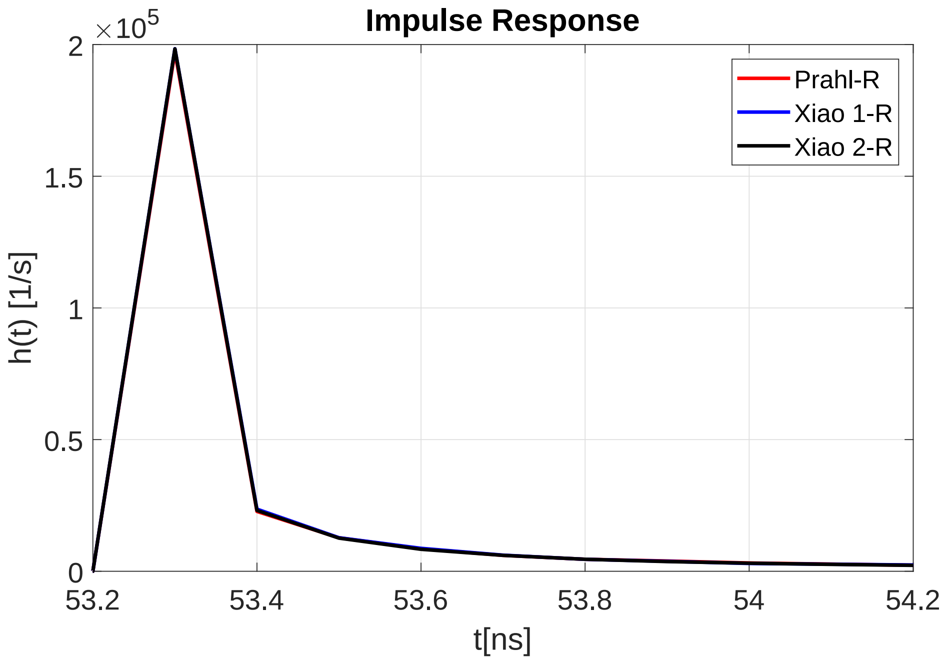


Figure 5.9: CIR obtained with Clear Ocean.

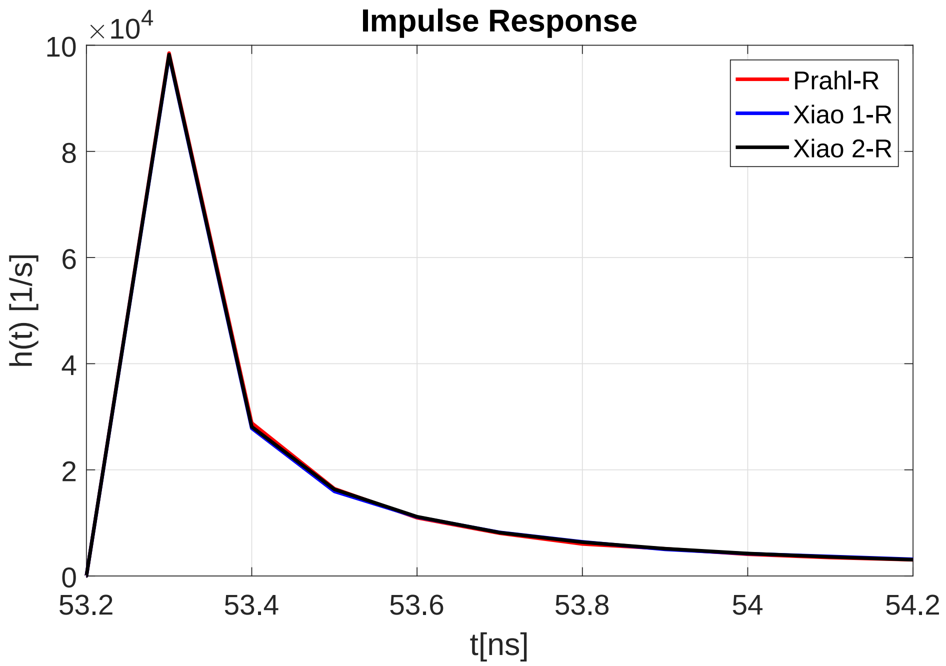
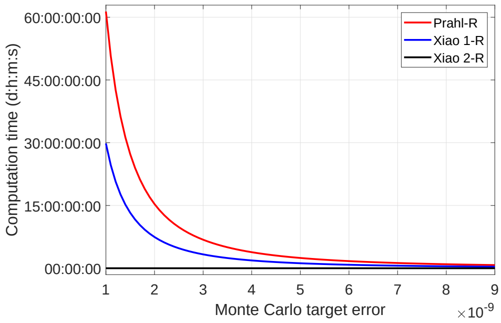
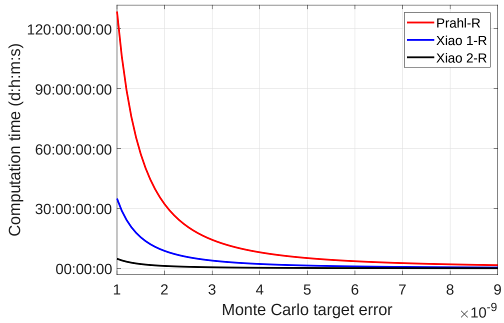


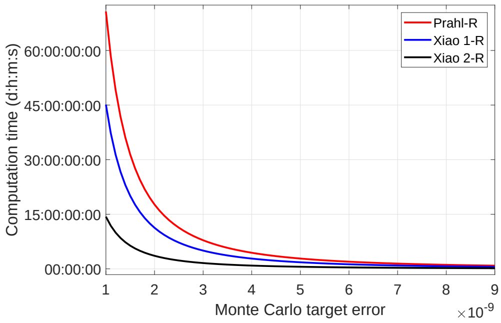
Figure 5.10: CIR obtained with Coastal.



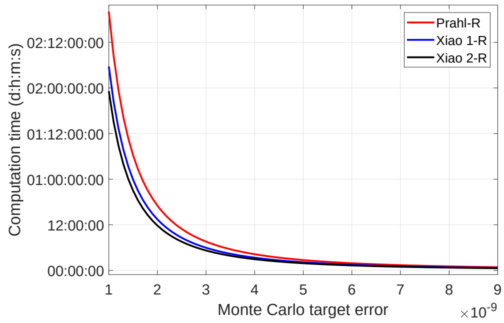
(a) Pure Sea



(b) Clear Ocean



(c) Coastal water



(d) Harbor water

Figure 5.11: Expected computation time varying the target error for different water types with initial configuration.

This study was conducted also for other water types, *i.e.* Pure Sea, Clear Ocean and Coastal. The corresponding results are given in Appendices D, E and F. Same behaviour can be observed according to Table D.11-F.11 and Figure D.6-F.6. Still, both Xiao 1-R and Xiao 2-R algorithm reduces the variance and so provide more precise results, in comparison to Prahl-R algorithm.

5.2.4 Changing the distance between sea ground and sensors

This section explores the impact of the distance between sea ground and sensors on the performance of these 3 algorithms. Table 5.4 shows the results obtained by moving the sensors above sea ground at distances ranging from 0.1 to 50 meters. The channel gain initially decreases as the distance increases until reaching 15 m, after which it stabilizes. This behaviour can be attributed to the larger water volume space associated with longer distances, resulting in a greater amount of light getting scattered. When the distance reaches 15m or more, its impact on channel gain is very

Table 5.3: Harbor - Unbiased sample variance obtained from Prahl-R, Xiao 1-R and Xiao 2-R algorithms varying the albedo of sea ground.

Albedo	Channel's gain	Prahl-R	Xiao 1-R	Xiao 2-R
0.1	$7.0245E^{-7}$	$6.3686E^{-8}$	$5.0984E^{-8}$	$3.3209E^{-8}$
0.2	$1.1739E^{-6}$	$1.8121E^{-7}$	$1.4627E^{-7}$	$9.8499E^{-8}$
0.3	$1.6869E^{-6}$	$3.9235E^{-7}$	$3.1918E^{-7}$	$2.0978E^{-7}$
0.4	$2.2918E^{-6}$	$7.2715E^{-7}$	$5.7734E^{-7}$	$3.8665E^{-7}$
0.5	$2.9567E^{-6}$	$1.2387E^{-6}$	$9.6351E^{-7}$	$6.4407E^{-7}$
0.6	$3.7241E^{-6}$	$1.8670E^{-6}$	$1.5657E^{-6}$	$9.7624E^{-7}$
0.7	$4.7120E^{-6}$	$2.9334E^{-6}$	$2.4045E^{-6}$	$1.6285E^{-6}$
0.8	$5.7300E^{-6}$	$4.4023E^{-6}$	$3.4372E^{-6}$	$2.3534E^{-6}$
0.9	$6.9007E^{-6}$	$6.8112E^{-6}$	$5.4410E^{-6}$	$3.6117E^{-6}$

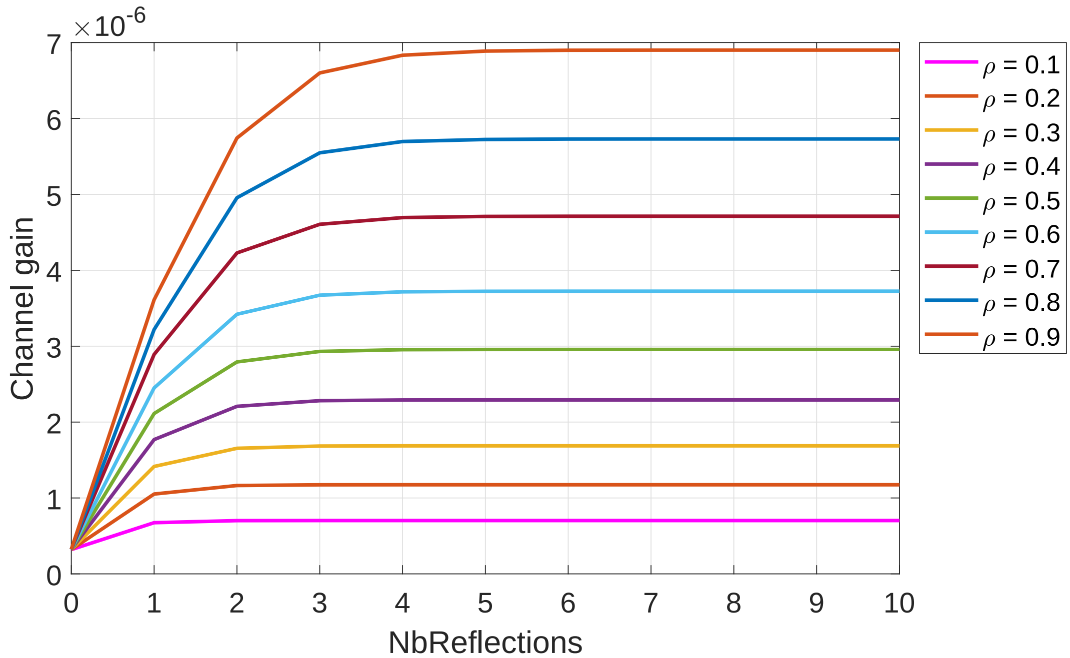


Figure 5.12: Channel gain varying $NbReflections$ with different albedo of sea ground for Harbor water

small. This is because the probability of light getting reflected is very low in such cases, and most light stops to propagate in the farther water volume. These findings are also supported by Figure 5.13, which illustrates the channel gain for different values of $NbReflections$. As the distance increases, contributions with at least one reflection become less significant for the overall channel gain. Additionally, the difference in channel gain between no reflection and one reflection becomes less prominent. These changes in channel gain correspond to similar behaviours in their respective unbiased sample variances.

But in the case of other water types, *i.e.* Pure Sea, Clear Ocean and Coastal, according to Tables D.12-F.12, and Figures D.7-F.7 of Appendices D, E and F, an opposite trend is observed. The channel gain increases as the distance increases until 5m, and then stabilizes after that. Nevertheless, Xiao 1-R and Xiao 2-R algorithm provides better results in terms of unbiased sample variance.

Table 5.4: Harbor - Unbiased sample variance obtained from Prahl's, Xiao 1-R and Xiao 2-R algorithms varying the distance between sensors and the sea ground.

Distance (m)	Channel's gain	Prahl-R	Xiao 1-R	Xiao 2-R
0.1	$1.8645E^{-6}$	$5.2876E^{-7}$	$5.6768E^{-7}$	$3.8440E^{-7}$
0.5	$1.8063E^{-6}$	$4.8146E^{-7}$	$4.2309E^{-7}$	$2.3574E^{-7}$
1	$1.6869E^{-6}$	$3.9235E^{-7}$	$3.1918E^{-7}$	$2.0978E^{-7}$
2	$1.3607E^{-6}$	$2.3802E^{-7}$	$1.8074E^{-7}$	$1.3050E^{-7}$
5	$6.7695E^{-7}$	$5.1675E^{-8}$	$4.1709E^{-8}$	$2.9440E^{-8}$
10	$4.8201E^{-7}$	$3.2928E^{-8}$	$2.7072E^{-8}$	$1.9144E^{-8}$
15	$4.6854E^{-7}$	$3.3595E^{-8}$	$2.5484E^{-8}$	$1.8131E^{-8}$
20	$4.7574E^{-7}$	$3.3350E^{-8}$	$2.6837E^{-8}$	$1.8792E^{-8}$
50	$4.7034E^{-7}$	$3.2965E^{-8}$	$2.6488E^{-8}$	$1.8573E^{-8}$

5.2.5 Discussion

The results presented in this chapter allow some general conclusions to be drawn about the effectiveness of the UWOC algorithms generalized to include reflections, *i.e.* the Prahl-R, Xiao 1-R and Xiao 2-R algorithms. Regardless of the water types, the seabed albedo and the distance between the seabed and sensors, the Xiao 1-R algorithm and, even more so, the Xiao 2-R algorithm consistently outperform the Prahl-R algorithm in terms of variance reduction and computation time. Incidentally, all these algorithms produce the same CIRs in each of the configurations tested, thereby demonstrating, if

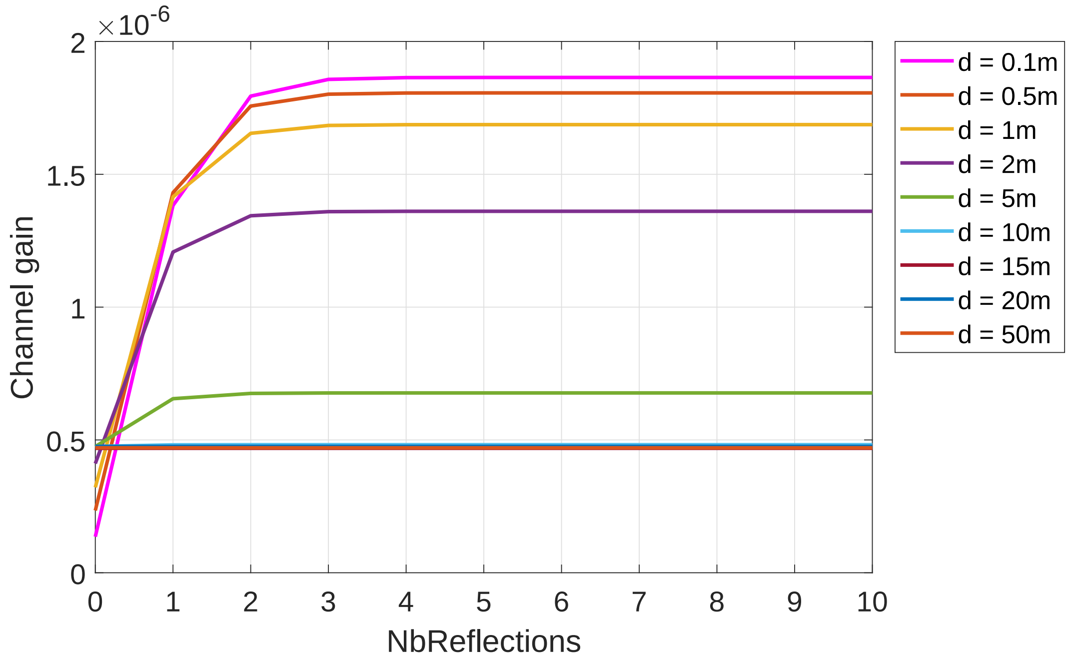


Figure 5.13: Channel gain varying $NbReflections$ with different distance between sea ground and sensors for Harbor water

not the accuracy of the formalisation provided in this chapter, at least its consistency.

Chapter 6

Application - Underwater Localization

With the rise of climate-related disasters, the global energy crisis, and mounting security concerns, the development of underwater activities has taken on paramount importance. Plenty of applications have emerged in response to these challenges, encompassing ecosystem research, exploration of natural resources, and the monitoring of sensitive sites [99]. Accommodating such applications necessitates the utilization of wireless networks that can offer efficient and adaptable connectivity among deployed sensors. This requirement has given rise to the paradigm of the Internet of Underwater Things (IoUT), defined as a global network of diverse interconnected underwater objects that communicate with each other to perform monitoring tasks in vast and unexplored water areas [100]. Communication technologies employed in IoUT are based either on acoustic, radio frequency (RF) or optical waves. Acoustic solutions offer long-range capabilities but suffer from high latency and low data rates, and can be also harmful to marine mammals. The use of RF waves faces restrictions due to their high attenuation in water, resulting in low range and bandwidth. A viable alternative lies in the utilization of optical waves, which enable high-speed communications, a range of hundreds of meters, high throughput, and the use of low-cost devices [101], especially within the visible range.

Some IoUT applications require the positioning knowledge of the mobile underwater devices equipped with sensors. A few papers study the positioning of autonomous underwater vehicles or divers using underwater visible light communication. Two algorithms have been proposed recently in [13, 102], using a few transmitters placed on the sea bed. The first assumes a perfect understanding of the underwater environment in terms of turbidity levels, while the second one handles scenarios with unknown turbidity levels. They build one or some databases (DB) containing channel's gains for different regular positions in the considered underwater environments using the Beer-Lambert (BL) model [103], a very simple light propagation model. In addition, the validation of the two algorithms was conducted using virtual measurements of channel's gains at unknown positions using the same BL model. The BL model only considers the Line Of Sight (LOS) attenuation caused by water absorption and out-scattering. It neglects the high number of incoming contributions due to scattering events that occur as light propagates through water, leading to an underestimation of the channel's gain [104]. Conversely, there exists more accurate approaches based on the Monte-Carlo evaluation of the integral form of the RTE. It is able to approximate the channel's gain that can be expected under real-world conditions, by taking into account up to hundreds scattering events in any water types.

The objective of this chapter is to investigate whether the two underwater visible light positioning (UVLP) algorithms proposed in [13, 102] are still valid when using a more realistic light propagation model based on Monte-Carlo, and more precisely the Xiao 2 algorithm. For this purpose, this chapter proposes to analyse the performance of the two UVLP algorithms by assessing their accuracies:

- using MC model for estimating the pseudo-measurements while keeping the database computed with BL model;
- using MC model to compute both the database and the pseudo-measurement.

This chapter is organized as follows. First, it recalls the two UVLP algorithms proposed in [13, 102]. Then it presents the BL model and compares it to MC model in terms of channel's gains. Afterwards, it details the methodology and the considered scenarios for the evaluation of the UVLP algorithms considering BL and MC models. Then, it presents some simulation results highlighting the impact of the used channel model on the UVLP algorithms' performance level. Finally, it concludes with a summary of the findings and perspectives for future research in this area.

6.1 Previous works

6.1.1 UVLP algorithms

In the recent years, there has been a growing interest in the use of optical wireless signals for underwater localization purposes [13, 102, 105, 106]. Traditionally, underwater localization sensors have primarily relied on acoustic technology, which comes with certain limitations regarding localization accuracy, typically in the order of meters. In contrast, UVLP systems are expected to be used in conjunction with high-data rate real-time communication and control applications, such as large-scale IoUT and video-surveillance via autonomous underwater vehicles (AUV) for instance. Considering the harsh nature of underwater environment, the deployment of UVLP techniques requires careful consideration. Several factors have the potential to impede underwater communication links. One such factor is the noise caused by wave motion, which can lead to link misalignment. Additionally, the interference from sunlight rays and the presence of the moving obstacles, *i.e.* fishes and other water beings, may result in blockage in communication links. However, there are UVLP solutions that neglect a few of such aspects, for example by laying LED-based anchor nodes on the seabed, and by omitting the wave motion and sunlight rays [13, 102].

In [102], the position estimation occurs by means of a foot printing approach. This UVLP algorithm is called Static DB, since it relies on a single database while it considers the water type well known. It goes through in the following three steps:

1. DB construction: it builds a DB of channel gains simulated by a given physical model between the fixed LEDs and the reception positions. Notice that the DB corresponds to the channel's gains for all the positions following a regular distribution in the given 3D space, as seen in Figure 6.1. It uses 4 LEDs into the considered volume.
2. Measurement acquisition: : it then considers the measured channel gains (virtual in this study, but real one in real use cases) for a set of unknown positions of a moving device in the given 3D space.
3. Position estimation: it searches the estimated positions that minimize the distance between the measured channel gains and the ones stored into the DB.

The propagation of light in water is influenced by the turbidity level, that has an impact on both the absorption and scattering levels, leading to very different channel gains. As such, it is necessary to store the DB that corresponds to the known turbidity level. In the evaluation of Static DB algorithm, authors have considered 4 different water types in [102]: Pure Sea, Clear Ocean, Coastal, and Harbour. Moreover, authors observed that the estimated positions are drastically modified when using the wrong DB. This indicates that the localization accuracy depends on the knowledge of the real environment turbidity level, allowing for the selection of the correct database. However, acquiring this information poses some challenges, owing to the vast amount of data and the rapid variation in time of turbidity in underwater environments. The

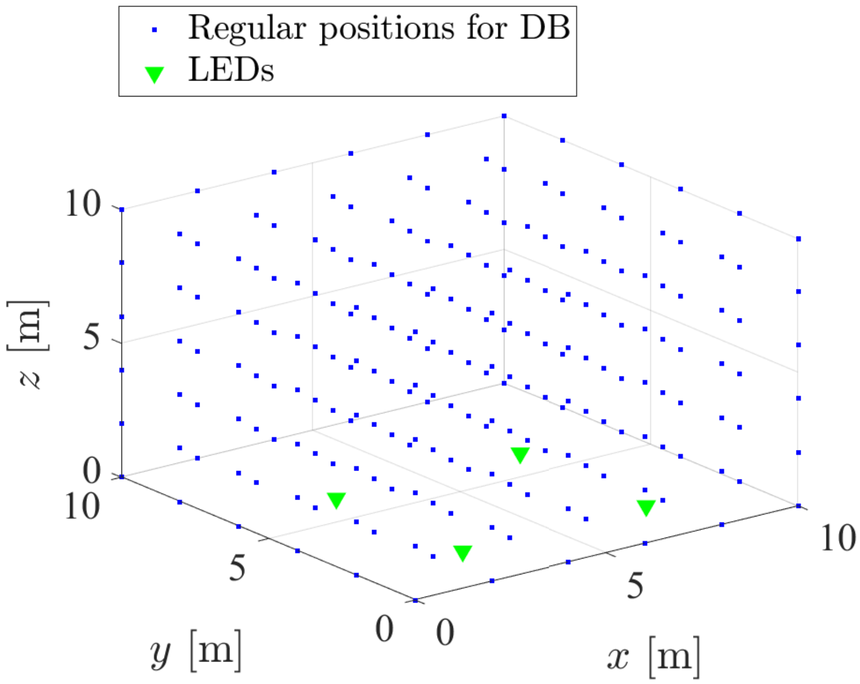


Figure 6.1: Regular positions for DB in a given 3D space

dynamics of the underwater environment, leading to fluctuating turbidity levels due to water currents, can sometimes cause the use of a wrong DB compared to the real water type in which is realized the measurement. Consequently, these mismatches result in increased errors in localization estimation.

To address this issue, an algorithm called Adaptive UVLP was proposed in [13]. On the basis of the Static-DB algorithm, it selects the reference database according to the information of Localization Mean Square Error (*LMSE*). The *LMSE* is defined as the distance between the real and estimated positions of the moving device. *LMSE* higher than a given threshold indicates the potential use of a "wrongly-selected" database that does not accurately represent the real underwater scenario. Then the reference database is updated accordingly. This adaptive approach helps to maintain high accuracy in localization estimation, with acceptable values in the order of a few centimetres.

6.1.2 Basic underwater light propagation models

Many studies have been conducted since 1960 to simulate the propagation of light in the ocean [27]. These efforts have primarily focused on solving RTE using MC

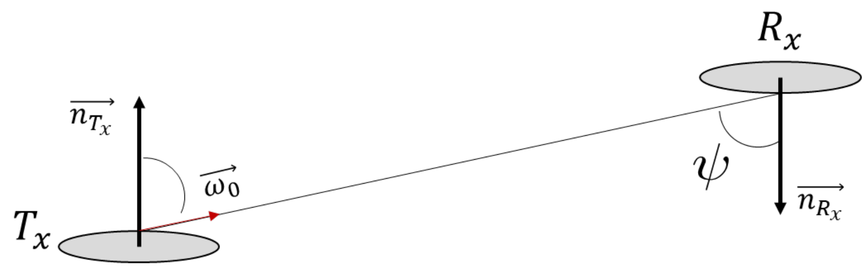


Figure 6.2: Light propagation in water based on the Beer-Lambert model

methods, where the underwater RTE revolves around the concept of transmittance T. In the case of a homogeneous participating medium, the transmittance is given by Equation (2.37), which is the well-known Beer's law, or Beer-Lambert's law, or the Beer-Lambert-Bouguer's law. It is used to approximate the channel gain between two sensors (the emitter Tx and the receiver R_x), by ignoring all the incoming light thanks to the scattering process, as shown in Figure 6.2. Based on Equation (2.37), the CIR model can be derived to characterize the received signal between T_x and R_x as follows:

$$h(T_x, R_x, t) = \frac{T(l) A_{R_x}(m+1)}{2\pi l^2} \mathrm{rect}\left(\frac{\psi}{\psi_c}\right) \times \delta(t-\tau)\left(-\overrightarrow{\omega_0} \cdot \overrightarrow{n_{R_x}}\right)\left(\overrightarrow{\omega_0} \cdot \overrightarrow{n_{T_x}}\right)^m, \quad (6.1)$$

where:

- τ is the propagation delay between T_x and R_x, expressed in seconds;
- A_{R_x} is the area of the receiver, expressed in square meters;
- m is the Lambert order. It depends on the definition of the Lambert emission model provided in Chapter 2.
- l is the distance between R_x and T_x, expressed in meters;
- ψ is the receiving angle;
- ψ_c is the receiver's field of view (FoV);
- $\overrightarrow{\omega_0}$ is the radiating direction from T_x;
- $\overrightarrow{n_{T_x}}$ is the normal vector at transmitter;
- $\overrightarrow{n_{R_x}}$ is the normal vector at receiver;
- $\mathrm{rect}(\cdot)$ is the well-known rectangle function;
- $\delta(\cdot)$ is the delta-Dirac function.

Notice that $\mathrm{rect}(\cdot)$ models the FoV, such as rays outside the FoV lead to a zero contribution.

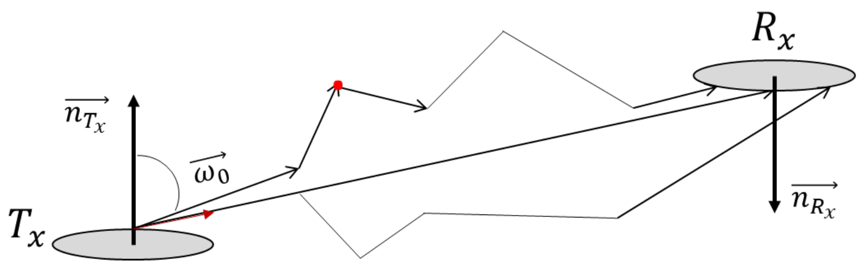


Figure 6.3: Light propagation in water based on Monte Carlo model

6.1.3 Comparison between BL model and MC model

Many works have employed the differential RTE in relation with MC simulation for underwater light propagation from 1965 [37] to 2020 [87]. Most of the current approaches rely on the Prahl's one [1]. In Chapter 3, the mathematical formalization based on Prahl's algorithm was proposed, leading to the development of two new MC models: the Xiao 1 and Xiao 2 algorithms. These models exhibit enhanced performance in terms of variance reduction and computation time. Particularly, the Xiao 2 algorithm demonstrates notable improvements, and is selected for computing the channel's gain for both the database and the virtual measurements.

The BL model only considers absorption and out-scattering, while the MC model also accounts for in-scattering, as illustrated in Figure 6.3. This results in a difference $Diff$ between channel's gains from the two models for the Initial configuration, as depicted in Table 6.1. With Pure Sea, where only a few scattering events occur, $Diff$ is small, and then both models yield comparable channel gains. However, the $Diff$ value increases significantly using waters having higher turbidity. This is due to the fact that in more turbid waters, the contributions from links involving multiple scatterings become more significant. As a result, the MC model produces higher gains compared to the BL model.

6.2 Analysis Set-up

6.2.1 Methodology

This section introduces the two scenarios employed to assess the performance of the two UVLP algorithms. The evaluation focuses on the positioning accuracy in various

Table 6.1: The difference in Channel's gain between BL model and MC model with the Initial configuration

Channel' gain	Pure sea	Clear ocean	Coastal	Harbour
BL	-23.3dB	-28.2dB	-36.2dB	-133.4dB
MC	-23.2dB	-25.5dB	-28.1dB	-57.8dB
$Diff$	0.1dB	2.7dB	8.1dB	75.6dB

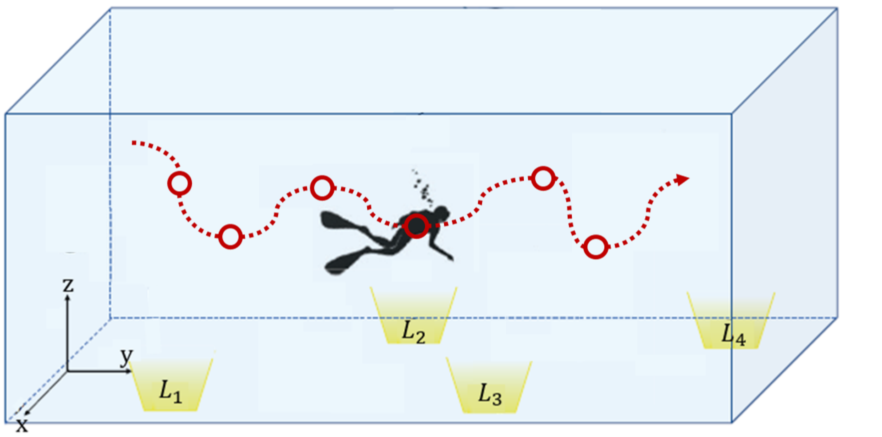


Figure 6.4: Underwater scenario illustration

water types, including Pure Sea, Clear Ocean, Coastal and Harbour waters. In this study, the assumed setting involves a diver moving within a $10 \times 10 \times 10$m water volume $\mathscr{S}$, as seen in Figure 6.4. The considered underwater network consists of a photodiode (PD) worn by the diver and pointing toward the seabed on one side, and four LED transmitters placed on the seabed and pointing upward on the other side. The DB is composed of channel gains corresponding to a set of reception locations, computed from a given model, *i.e.* BL model or MC integration. Notice that these reception localizations are regularly distributed in the water volume $\mathscr{S}$, with a spatial resolution of $\Delta = 0.1$m, leading to 1 million positions. All the relevant parameters for the sensors and water characteristics can be found in Table 6.2 and Table 6.3 respectively. The row g in Table 6.3 corresponds to the parameter of the Henyey-Greenstein VSF detailed in Equation (2.27).

The performance analysis is done according to various mobility scenarios where the diver moves within the water volume $\mathscr{S}$, resulting in a series of successive positions P_i.

Table 6.2: Scenario's Parameters

Parameters	**values**
Monitoring space, $\mathscr{S}$	$10 \times 10 \times 10$m
L_1 coordinates	$(2.7, 1.9, 0)$m
L_2 coordinates	$(2.7, 6.2, 0)$m
L_3 coordinates	$(7.5, 1.9, 0)$m
L_4 coordinates	$(7.5, 6.2, 0)$m
LEDs power	1W (normalized)
LEDs shape / area	Circular / 1cm^2
LEDs radiation pattern	Lambertian, 50°, $m = 1.57$
PD's shape / A_{Rx}	Circular / 1cm^2
FoV of the receivers	60°

Table 6.3: Water Characteristics

	Pure sea	**Clear ocean**	**Coastal**	**Harbour**
σ_a	0.053	0.07	0.088	0.295
σ_s	0.003	0.08	0.217	1.875
σ_t	0.056	0.15	0.305	2.17
g	0.924	0.924	0.947	0.9199
MC samples	$60E6$	$60E6$	$60E6$	$300E6$

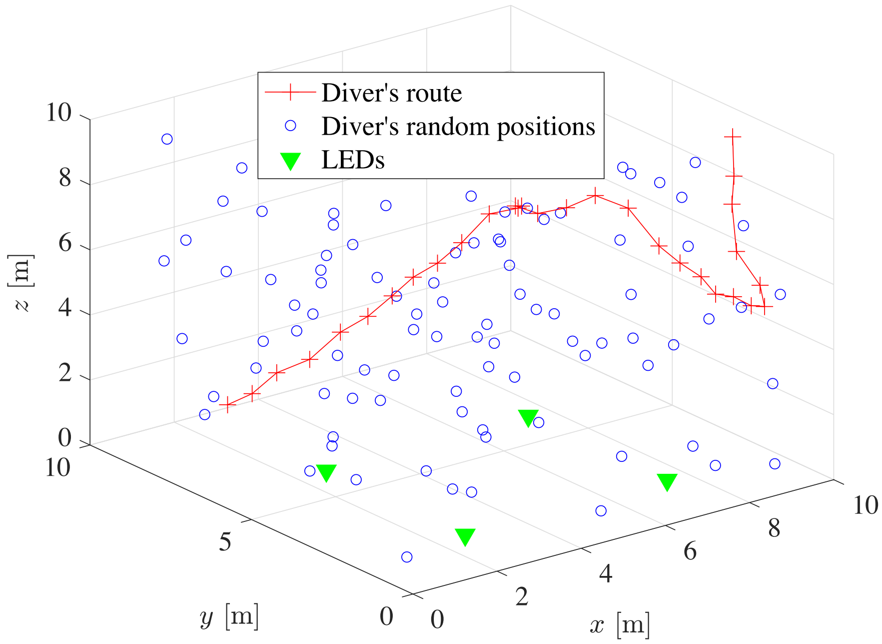


Figure 6.5: Two mobility scenarios.

Based on a given channel model, the performance of the UVLP system is characterized by the estimation error of the i^{th} diver's position P_i and its estimated position $\hat{P}_i$, using the Euclidean distance $d(P_i, \hat{P}_i)$. It leads to the following error criterion:

$$LMSE\left(P_i, \hat{P}_i\right) = d(P_i, \hat{P}_i). \tag{6.2}$$

6.2.2 Considered scenarios

The first considered mobility scenario (Scenario 1) comes from [102], consisting of a virtual measurement route with a total of 30 positions, marked in red in Figure 6.5. These positions are spread out along the x-axis from 0.2 to 9.9 meters, along the y-axis from 0.2 to 6 meters, and along the z-axis from 3 to 9.9 meters. It is worth noting that some reception localizations with small height like from 0 to 3 meters are not considered in this scenario, where the BL model leads to null channel gains. Indeed, in such cases the ULP is modelled as a single straight line falling outside the receivers' FoV. In reality, the light propagation direction to the receiver is changed as it undergoes scattering events. This leads to several contributions falling inside the FoV for all receiving positions, and consequently to non-zero channel gains. To avoid this restriction and enable a comprehensive performance evaluation for all possible diver's positions, a second and more generic scenario (Scenario 2) is employed. It involves a virtual measurement path of 100 reception positions randomly distributed according to a uniform distribution in all the water volume $\mathscr{S}$. These positions are marked in blue circles as illustrated in Figure 6.5.

6.3 Simulation Results

This section presents the simulated results by assessing the UVLP algorithms based on (i) BL model and (ii) Xiao 2. Their effectiveness are primarily evaluated using $LMSE$. The position estimation is performed using a footprint approach, where different DBs can be chosen according to the water turbidity level. Two approaches are considered: Static-DB and A-UVLP approaches, as described in [102] and [13], respectively. The notation adopted in this section is "DB/Meas.", where DB and Meas. represent the model adopted for building the DB and for estimating the virtual measure, respectively. Notice that both DB and Meas. can be either BL for Beer-Lambert model or MC for Monte Carlo one.

6.3.1 Static-DB approach

6.3.1.1 BL/BL

Scenario 1

At first, the water type in which the virtual measurements were conducted is assumed known. In [102], only BL model is taken into account for both computing the DB and then evaluating the channel gains along the diver's path, depicted in red on Figure 6.5. The corresponding results are presented in Table 6.4, involving the mean (μ) and the standard deviation (σ) of the $LMSE$ along the diver's path according to four different water types. Notice that both μ and σ are expressed in meters. It can be observed that the mean value of the $LMSE$ varies from 0.07m to 0.25m according to the water type, indicating a good accuracy of the algorithm. This is further emphasized in Figure 6.6, which displays the positioning errors for each diver's position in Harbour water using colour levels.

Scenario 2

In order to ensure a comprehensive performance evaluation for all possible diver's

Table 6.4: Static-DB. Mean and standard deviation values of *LMSE* [m] in scenario 1

	Pure sea		**Clear ocean**		**Coastal**		**Harbour**	
DB/Meas.	μ	σ	μ	σ	μ	σ	μ	σ
BL/BL	0.09	0.13	0.07	0.06	0.09	0.10	0.25	0.22
BL/MC	0.15	0.18	1.87	0.56	3.93	0.93	6.09	1.78
MC/MC	0.07	0.06	0.07	0.03	0.07	0.04	0.67	1.87

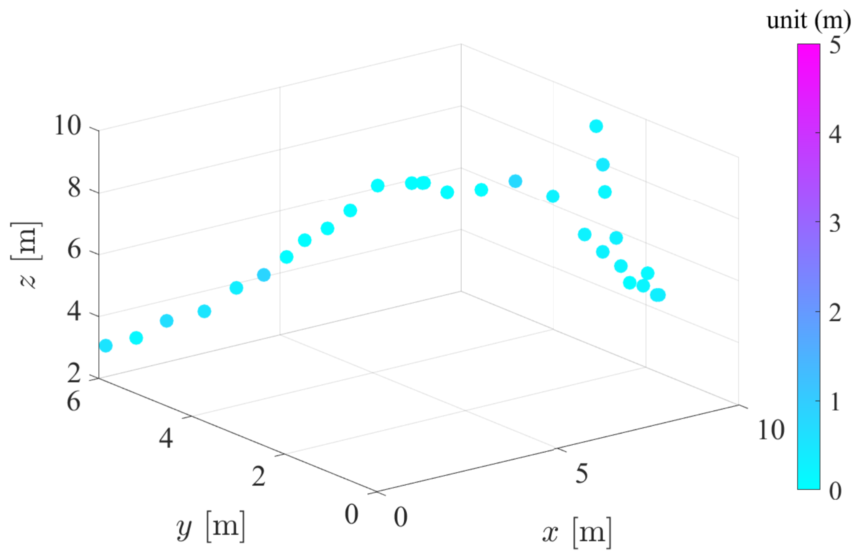


Figure 6.6: *LMSE* for each diver's position considering BL/BL in Harbour water (scenario 1)

positions, a more versatile scenario is used. This scenario encompasses a total of 100 random diver positions, marked with blue circles in Figure 6.5. The corresponding results presented in Table 6.5 shows a significant increase of the positioning errors regardless of the water types, as compared to scenario 1. The mean *LMSE* values are multiplied by factors ranging from 6 to 17 according to water type. The reason behind this increase in error can be better understood by referring Figure 6.7, which depicts the positioning errors at different diver's positions using colour levels. It becomes apparent that the highest errors are observed for the lowest receiving positions. This observation is further supported by Figure 6.8, which focuses on harbour water as an example. It illustrates the evolution of the mean *LMSE* values computed for the receiving positions of scenario 2 located in 1m thick horizontal planes at different heights given in abscissa. Clearly, the highest errors occur for the lowest receiving positions, where the BL model fails to accurately predict the channel gain, owing to the limited receiver's FoV.

Table 6.5: Static-DB. Mean and standard deviation values of *LMSE* [m] in scenario 2

	Pure sea		**Clear ocean**		**Coastal**		**Harbour**	
DB/Meas.	μ	σ	μ	σ	μ	σ	μ	σ
BL/BL	1.18	2.52	1.18	2.54	1.13	2.52	1.49	2.42
BL/MC	0.56	1.00	1.71	0.96	3.04	1.71	4.60	2.17
MC/MC	0.41	0.66	0.35	0.59	0.36	0.58	0.56	1.15

6.3.1.2 BL/MC

Scenario 1

A realistic evaluation of the Static-DB algorithm deployed in a real environment is an important goal. However, conducting real measurement for this evaluation is not an easy task, and it was not done during this PhD. But, it is possible to estimate the measurements using an MC model, that provides realistic channel model especially when the water turbidity grows. With scenario 1, the simulated results using the MC model described in Chapter 4 for the virtual measurement only (and not the DB) are provided in Table 6.4. It appears that using BL model for DB and MC model for

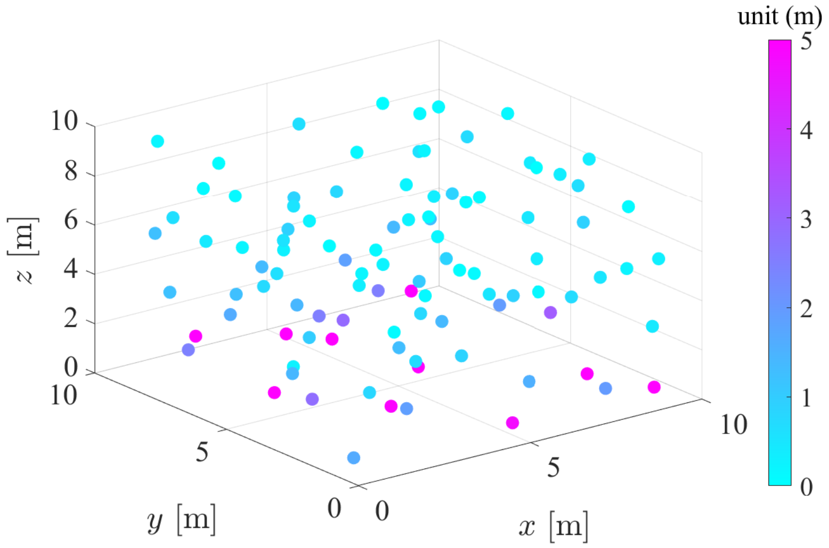


Figure 6.7: *LMSE* along the diver's positions for BL/BL in Harbour water: scenario 2.

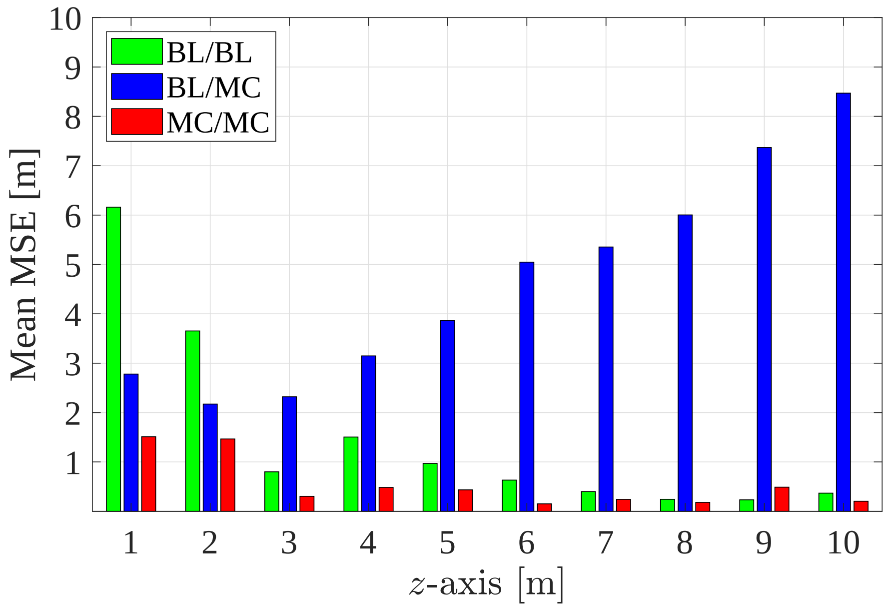
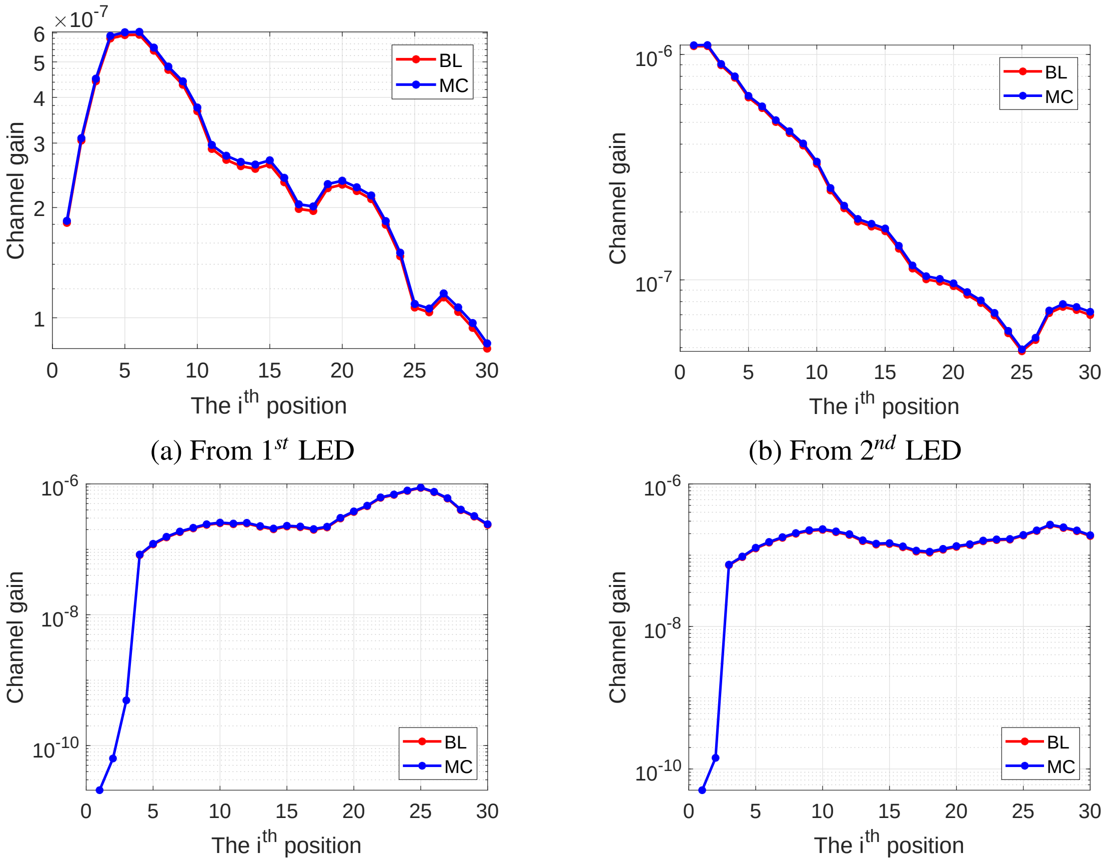


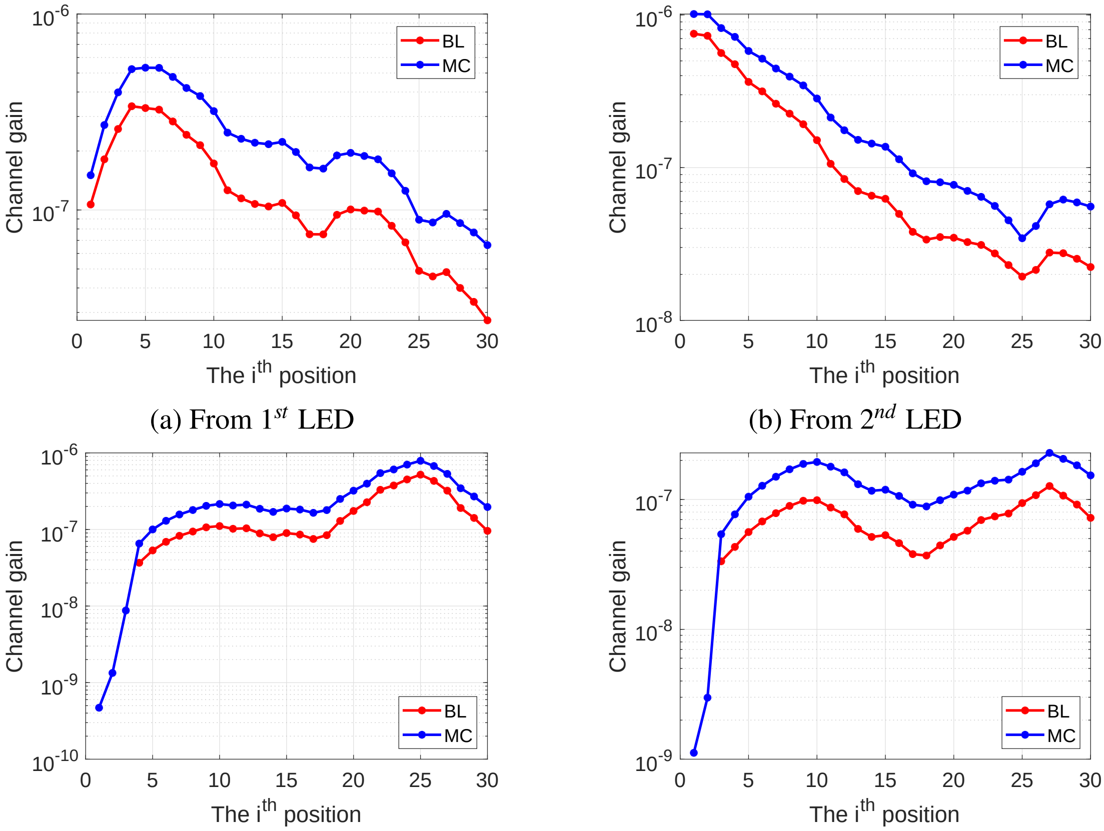
Figure 6.8: Mean *LMSE* values *vs.* the receiving height in Harbour.

virtual measurement decreases the accuracy level of the Static-DB algorithm significantly compared to that of using BL for both DB and virtual measurement. The mean *LMSE* values are multiplied by factors of 1.7 for Pure Water, 26.77 for Clear Ocean, 43.7 for Coastal, and 24.4 for Harbour. This decrease in accuracy is notable except for Pure Sea, with the maximum positioning error reaching up to 6.09 meters for Harbour. The contributions of links having multiple scatterings are more significant in channel gain for more diffuse water, leading to more significant difference between BL and MC model for more diffuse water, as seen in Figure 6.9 for Pure Sea, Figure 6.10 for Clear Ocean, Figure 6.11 for Coast water and Figure 6.12 for Harbour water. This explains the observed small decrease in accuracy for Pure Sea in comparison to that for other 3 water types. An assumption that may lead to this decrease of accuracy is the unrealistic nature of the model used to compute the DB.



(a) From 1^{st} LED
(b) From 2^{nd} LED
(c) From 3^{rd} LED
(d) From 4^{th} LED

Figure 6.9: Pure Sea - the measured channel gains between LEDs and 30 reception positions (Scenario 1)



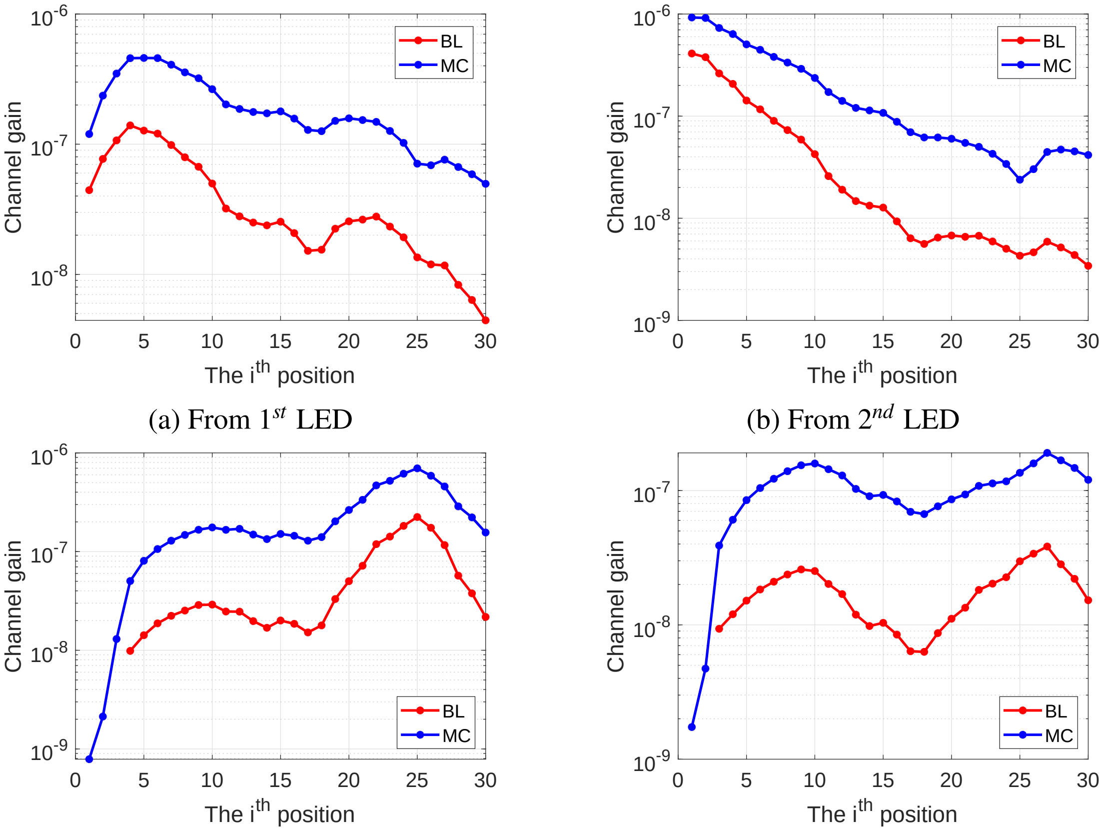
(a) From 1^{st} LED

(b) From 2^{nd} LED

(c) From 3^{rd} LED

(d) From 4^{th} LED

Figure 6.10: Clear Ocean - the measured channel gains between LEDs and 30 reception positions (Scenario 1)



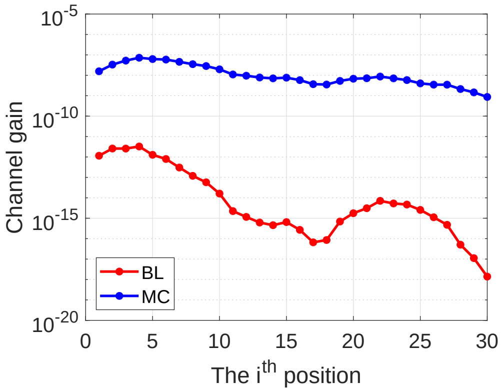
(a) From 1st LED

(b) From 2nd LED

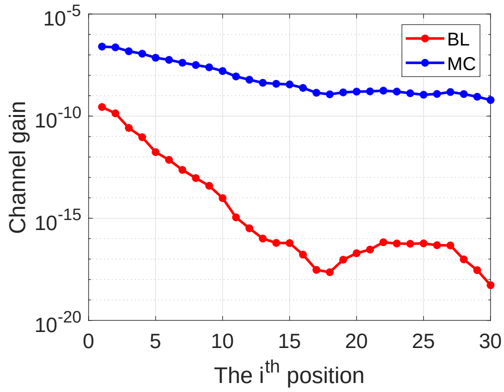
(c) From 3rd LED

(d) From 4th LED

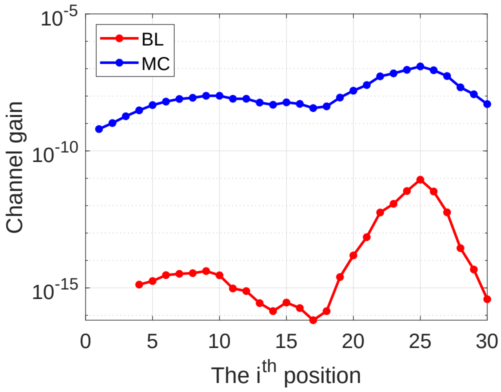
Figure 6.11: Coastal - the measured channel gains between LEDs and 30 reception positions (Scenario 1)



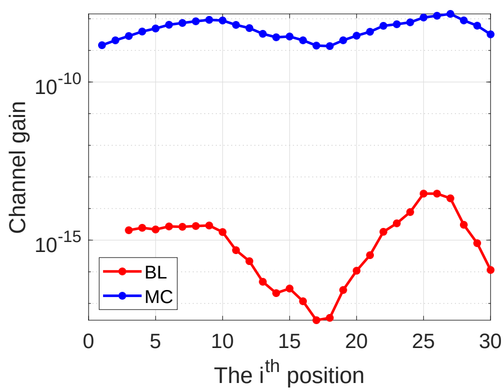
(a) From 1^{st} LED



(b) From 2^{nd} LED



(c) From 3^{rd} LED



(d) From 4^{th} LED

Figure 6.12: Harbour - the measured channel gains between LEDs and 30 reception positions (Scenario 1)

Scenario 2

When using a more generic scenario, the similar behaviour as for scenario 1 can be observed in Table 6.5. Comparing to the results of BL/BL corresponding to scenario 2, the mean *LMS* values are multiplied by factors ranging from 0.48 for coastal water to 3.09 for harbour water. There are some positions in scenario 2 where the measured channel gains evaluated by BL model is 0, leading to bad estimated positions. But in fact, there exist some contributions at these positions due to multiple scattering events. So Static-DB can find the positions not very far from the real ones. Notice that the difference in measured channel gains between BL and MC is quite small for Pure sea. So there is an increase in accuracy in the case of Pure sea. But for other water types, the difference in measured channel gain is significant, so the accuracy decreases. Here again, a possible reason resulting in this behaviour is the nature of BL model, which does not take into account the scattering events in the DB computation that contribute to an increase in the channel gain, particularly for longer distances between the transmitter and reception positions. In this case, the long distance implies a higher number of scattering events. As confirmed in Figure 6.8 for harbour water (depicted in blue bars), the highest positioning errors occur for the highest receiving positions.

6.3.1.3 MC/MC

Scenario 1

To investigate whether the nature of BL model to compute the DB causes a decrease in positioning accuracies, the DBs are recomputed by using the MC model, and then the Static-DB algorithm is retried along the diver's route (the red path in Figure 6.5). The corresponding results are given in Table 6.4. Remarkably, the mean *LMSE* values are of the same order as those obtained with the BL/BL approach. This highlights the importance of considering a realistic channel model for the DB computation, to avoid underestimating the performance of the UVLP algorithm. In the same time, it confirms the effectiveness of the Static-DB algorithm.

Scenario 2

Similarly, in the case of scenario 2, the corresponding results presented in Table 6.5 show a good accuracy level with a maximum positioning error in the same order as for scenario 1. This holds true even for the lowest receiving positions, as confirmed by red bars in Figure 6.8. The improved accuracy is attributed to the nature of MC approach, which considers as many scattering events as necessary to model the propagation of light in a non-straight line from the optical source, so that the light falls within the FoV of the receivers.

6.3.2 A-UVLP approach

This section analyses the performance level of the A-UVLP algorithm proposed in [13] for estimating the diver's position in the presence of unknown water types. In [13], the authors demonstrated that this algorithm can reduce the mean *LMSE* by dynamically switching different DBs associated with different water types, compared to using a single potentially incorrect DB. However, the evaluation in scenario 1 was limited to receivers located in a narrow horizontal layer with a height of less than 3m. To provide unbiased and comprehensive results, the evaluation in this study considers scenario 2, which encompasses the entire space $\mathscr{S}$ depicted in Figure 6.5. The results are given in Table 6.6 according to virtual measurements performed in different water types. The mean *LMSE* statistics were averaged over the 4 possible initial DBs, *i.e.* pure sea, clear ocean, coastal, and harbour waters.

Table 6.6: A-UVLP. Mean and standard deviation values of *LMSE* [m] in scenario 2

	Pure sea		**Clear ocean**		**Coastal**		**Harbour**	
DB/Meas.	μ	σ	μ	σ	μ	σ	μ	σ
BL/BL	1.93	2.53	2.49	2.21	4.28	2.79	3.73	3.38
BL/MC	2.88	1.68	2.68	2.06	2.68	2.06	4.77	2.89
MC/MC	0.79	1.10	0.66	0.90	0.58	0.70	3.03	2.78

When the BL model is employed to evaluate the DB, the A-UVLP algorithm exhibits significantly higher positioning errors, particularly when the virtual measurements are evaluated using a realistic MC model. In such cases, the mean *LMSE* can exceed 4.7m, indicating a substantial degradation in accuracy. In contrast, when the MC approach is utilized for both DB computation and virtual measurement evaluation, the mean *LMSE* can be limited to a same order as the one obtain by knowing the water type, *i.e.* the

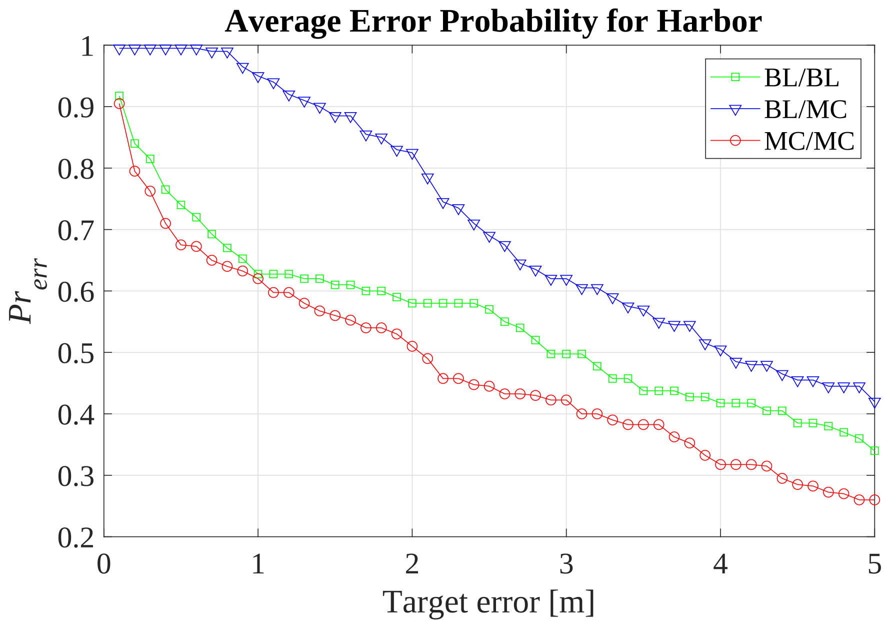


Figure 6.13: Average error probability *vs.* target error for harbour water.

one provided by considering a single good DB (*cf.* Subsection 6.3.1.3), except for harbour water for which the error is higher. This can be attributed to a relatively low number of MC samples used for building the database, resulting in increased variance in CIR estimation, especially at lower z positions. Finally, Figure 6.13 depicts the error probability Pr_{err} computed for a given position P_i in harbour water and expressed as:

$$Pr_{err} = P\{LMSE^{(P_i)} > \xi\}, \tag{6.3}$$

where ξ is a target error, assumed in the range $[0.1, 5]$ meters. One more time, these curves confirm the need for a realistic model to compute the DB in order to not underestimate the positioning errors. As an example, 50% of positioning errors are higher than 2m when the DB is computed by the MC (red) model compared to 4m when the DB is computed by BL (blue) model.

6.4 Conclusions and new perspectives for UVLP algorithm

Using a more accurate MC propagation model including as many scattering events as necessary, this chapter explores the validity of two recent UVLP algorithms, originally designed and validated with the Beer-Lambert light propagation model for computing both its DB and the virtual measurements. The findings highlight that these UVLP algorithms remain effective when a realistic model is used to compute both the DB and virtual measurements, yielding roughly higher levels of accuracy. However, this chapter also emphasizes the importance of using a robust propagation model for the DB computation, as using the Beer-Lambert model yields poor results when paired with the virtual measurements generated by MC model. Since the MC model closely resembles real underwater measurements compared to the Beer-Lambert model, similar results can be expected when using real measurements.

This chapter has some interesting prospects for future research. Firstly, it raises the question of validating these algorithms using real measurements, which would require an independent system to obtain the expected localization. However, such validation would be challenging as it would necessitate the measurement of various reference parameters, such as the AUV's position and the water type, etc. Additionally, it would be worthwhile to investigate the behaviour of these algorithms by incorporating more data into the DB, considering various locations, different water types, and different receiver orientations and models.

Besides, there is potential for enhancing the effectiveness of UVLP algorithms. One approach involves extracting a set of points with the same channel gains from the results obtained from the MC simulation model. Due to the symmetrical radiation/detection patterns of the sensors, these point sets correspond to circles. Each circle is characterized by its height h and the distance d between the transmitter and the receiving positions, as depicted in Figure 6.14. We have one measured gain for each transmitter T_x (4). Each measured gain value corresponds to one (or more) circle, so on (or more) couple (d,h) for each T_x. Each circle or couple (d,h) is so a set of potential locations of the mobile sensor. Theoretically, the real location of the mobile sensor should be defined as being the intersection between these sets of potential solutions. In case of more than one circle/couple (d,h), the use of an optimization algorithm as Levenberg-Marquardt could help to find the global optimum corresponding to the real location. The future research will continue to explore the implementation of the approach.

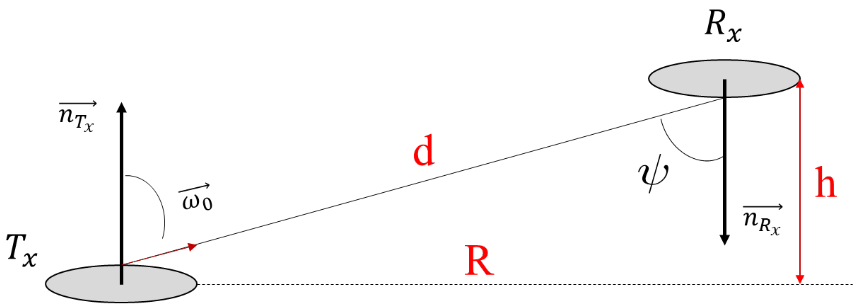


Figure 6.14: Schematic of the deployment of the transmitter and the receiver

Chapter 7

Conclusions and perspectives

7.1 Conclusions

The context of this thesis is to develop accurate and fast simulation algorithms for the light propagation through participating media, which are necessary for the design of new efficient OWC systems. In the realistic environment, different physical phenomena may occur as light interacts with participating media: light absorption, due to particles absorbing and converting the light flux into heat; light scattering, due to particles that deflect the luminous flux in other directions, causing the dispersion of the light. Moreover, owing to the presence of obstacles, surface reflections may occur and affect the behaviour of light propagation channel. These imply that simulating the OWC channel in such environments can be extremely difficult.

In the previous work, most of the approaches is based on an algorithm stated in 1989 by Prahl, which itself is built upon the earlier work established by Collins and Wells, based on neutron transport simulations. These former solutions rely on Monte Carlo simulation, requiring a large amount of photons to obtain good results with low error. With Monte Carlo integration, there exist some optimization techniques to reduce variance. However, suggesting improvements of this old algorithm necessitates a well-defined mathematical foundation using integrals, which is the primary task of this thesis.

To realistically and efficiently simulate light propagation in some environments containing participating media, the adopted method involves considering configurations of increasing complexity. Initially, a participating medium of infinite size with no surface is considered, followed by the general case of a participating medium containing surfaces.

In this context, we therefore first propose a new mathematical formalization of light propagation through participating media, considering only scattering events. This allowed to retrieve the original Prahl's algorithm but using the Monte Carlo integration, *i.e.* a numerical integration technique using random numbers. The well-known Monte Carlo variance reduction techniques can be then employed in this new formalism, like importance sampling which is the basis of Monte Carlo simulation. Our new formalization having given rise to different variations, we have also proposed two new algorithms (Xiao 1 and Xiao 2) to estimate the impulse response of the light propagation channel. Different from Prahl's method, the former removes the importance sampling on the last propagating distance, leading to higher probability of light arriving at the receiver. On this basis, the latter directly samples a solid angle enclosing the receiver with importance, further increasing this probability.

We have performed extensive simulations considering water medium with different turbidities as examples. The obtained results show that these two new algorithms allow for a given number of MC samples to obtain better performance with lower sample variances and therefore lower error as compared to Prahl's, especially Xiao 2 algorithm. With less diffuse water, Xiao 2 provides more significant variance reduction considering the Initial configuration. From Pure Sea, Clear Ocean, Coastal to Harbor, the unbiased sample variance of Xiao 2 has been reduced to 0.01%, 2.93%, 14.33% and 57.14% of Prahl's one. Besides, this algorithm also exhibits more significant variance reduction with smaller receiver, regardless of water turbidities. For instance, in the case of Harbor, with the radius of the receiver increasing from 0.5 cm to 50 cm, its variance has been reduced to from 0.64% to 81.87% of Prahl's one. Alternatively and for a given target error, our new algorithms are faster than Prahl's in all tested situations, especially Xiao 2. Similarly, for less diffuse water and for smaller receiver, Xiao 2 provides greater efficiency as compared to Prahl's one.

Considering the realistic environment with obstacles and surface, *i.e.* sea surface, or other obstacles as seabed in underwater environment, reflections can impact the light propagation channel. To account for this, we proposed to generalize the path space considering both volume and surface, providing an explicit formalization of the light propagation channel. Based on the new mathematical formalism considering both participating media and obstacles, we then develop two extended algorithms (Xiao 1-R and Xiao 2-R). Likewise from the simulation results with water medium, in terms of accuracy and efficiency, these two algorithms exhibit better performance in all tested situations, especially Xiao 2-R. For less diffuse water, Xiao 2-R exhibits more significant variance reduction and greater efficacy.

As an important application of UWOC study, underwater localization represents a challenge due to the complex and dynamic nature of underwater environment. In recent research, the UVLP algorithm has been utilized to estimate localizations by employing Beer-Lambert light propagation model. Nevertheless, this model solely takes into account absorption and out-scattering while neglecting the crucial in-scattering effect. To address this limitation, we propose the adoption of a more realistic MC model, more specifically the model corresponding to Xiao 2 algorithm. As expected, it provides more accurate position estimation as compared to Beer-Lambert model, *i.e.* the mean localization error being 0.56 m when using MC (Xiao 2), and 1.49 m when using BL, considering 100 random positions in Harbor. This highlights the importance of considering a realistic channel model in such an application.

7.2 Perspectives

The first perspective is to conduct actual measurements in underwater environments to further validate the novel algorithms proposed in this dissertation. However, conducting such measurements needs careful considerations. These include specialized equipment for underwater data collection, precise calibration to account for underwater complexities, and environmental factors that can influence the measurements.

The second perspective is to confirm the performance of these new algorithms with other configuration parameter of scattering model, *i.e.* TTHG model, Rayleigh model. Especially for Xiao 2 or Xiao 2-R algorithms, these models are directly related to the computation of normalization factor. Besides, Rayleigh model is limited to very small particles composing the medium: the particle's size is up to a tenth of the wavelength of light. This requires to carefully consider both the size of media particles and the wavelength of the light when using such model.

The third perspective is to apply our algorithms to multiple layered participating media, *i.e.* from air to water, from air to glass. When light interacts with the edges of these media, it may undergo reflections or scattering. This requires careful consideration of transitioning from surface space $\Omega_{S/2}$ to volume space Ω_S, accounting for the complexities introduced by the different scattering properties of each participating medium.

Lastly, these new algorithms can be applied in the new underwater localization method, mentioned as a perspective of UVLP algorithm in Chapter 6 used to estimate the positions of the moving target. This approach involves extracting a set of points with identical channel gains from the obtained results using these simulation algorithms. The channel gain of the moving target is impacted by the water turbidities. As a part of future work, it will be interesting to use this approach to estimate turbidities of the water composing the environment.

REFERENCES

[1] S. A. Prahl, "A Monte Carlo model of light propagation in tissue," in *Dosimetry of laser radiation in medicine and biology*, vol. 10305. SPIE, 1989, pp. 105–114.

[2] W. Jarosz, *Efficient Monte Carlo methods for light transport in scattering media*. University of California, San Diego, 2008.

[3] C. Singh, J. John, Y. Singh, and K. Tripathi, "A review of indoor optical wireless systems," *IETE Technical review*, vol. 19, no. 1-2, pp. 3–17, 2002.

[4] H. Le Minh, Z. Ghassemlooy, D. O'Brien, and G. Faulkner, "Indoor gigabit optical wireless communications: challenges and possibilities," in *2010 12th International Conference on Transparent Optical Networks*. IEEE, 2010, pp. 1–6.

[5] S. Arnon, J. Barry, and G. Karagiannidis, *Advanced optical wireless communication systems*. Cambridge university press, 2012.

[6] M. Uysal and H. Nouri, "Optical wireless communicationsan emerging technology," in *2014 16th international conference on transparent optical networks (ICTON)*. IEEE, 2014, pp. 1–7.

[7] M. Kavehrad, M. S. Chowdhury, and Z. Zhou, *Short-range optical wireless: theory and applications*. John Wiley & Sons, 2016.

[8] I. K. Son and S. Mao, "A survey of free space optical networks," *Digital communications and networks*, vol. 3, no. 2, pp. 67–77, 2017.

[9] M. Higgins, Z. Rihawi, Z. A. Mutalip, R. Green, and M. S. Leeson, "Optical wireless communications in vehicular systems," *Communication in Transportation Systems*, pp. 209–222, 2013.

[10] H. Mei, J. Ding, J. Zheng, X. Chen, and W. Liu, "Overview of vehicle optical wireless communications," *IEEE Access*, vol. 8, pp. 173 461–173 480, 2020.

[11] L. Lanbo, Z. Shengli, and C. Jun-Hong, "Prospects and problems of wireless communication for underwater sensor networks," *Wireless Communications and Mobile Computing*, vol. 8, no. 8, pp. 977–994, 2008.

[12] S. Zhu, X. Chen, X. Liu, G. Zhang, and P. Tian, "Recent progress in and perspectives of underwater wireless optical communication," *Progress in Quantum Electronics*, vol. 73, p. 100274, 2020.

[13] A. M. Vegni and V. Loscrí, "Adaptive visible light positioning with mse inner loop for underwater environment," in *2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*. IEEE, 2022, pp. 601–606.

[14] B. Patnaik and P. K. Sahu, "Inter-satellite optical wireless communication system design and simulation," *IET Communications*, vol. 6, no. 16, pp. 2561–2567, 2012.

[15] A. Alipour, A. Mir, and A. Sheikhi, "Ultra high capacity inter-satellite optical wireless communication system using different optimized modulation formats," *Optik*, vol. 127, no. 19, pp. 8135–8143, 2016.

[16] A. Choudhary and N. K. Agrawal, "Inter-satellite optical wireless

communication (isowc) systems challenges and applications: A comprehensive review," *Journal of Optical Communications*, no. 0, 2022.

[17] A. M. Pravilov, *Radiometry in modern scientific experiments*. Springer, 2011.

[18] H. Kaushal and G. Kaddoum, "Underwater optical wireless communication," *IEEE access*, vol. 4, pp. 1518–1547, 2016.

[19] A. Behlouli, P. Combeau, and L. Aveneau, "Mcmc methods for realistic indoor wireless optical channels simulation," *Journal of Lightwave Technology*, vol. 35, no. 9, pp. 1575–1587, 2017.

[20] J. F. Blinn, "Models of light reflection for computer synthesized pictures," in *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, 1977, pp. 192–198.

[21] R. Montes and C. Ureña, "An overview of brdf models," *University of Grenada, Technical Report LSI-2012*, vol. 1, p. 19, 2012.

[22] J. T. Kajiya, "The rendering equation," in *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, 1986, pp. 143–150.

[23] A. Behlouli, P. Combeau, L. Aveneau, S. Sahuguede, and A. Julien-Vergonjanne, "Efficient simulation of optical wireless channel application to wbans with miso link," *Procedia Computer Science*, vol. 40, pp. 190–197, 2014.

[24] A. Schuster, "Radiation through a foggy atmosphere," *Astrophysical Journal,*

vol. 21, p. 1, vol. 21, p. 1, 1905.

[25] R. W. Preisendorfer, “A mathematical foundation for radiative transfer theory,” *Journal of Mathematics and Mechanics*, pp. 685–730, 1957.

[26] S. Chandrasekar, “Radiative transfer dover publications,” *New York*, 1960.

[27] R. W. Preisendorfer, *Radiative transfer on discrete spaces*. Pergamon Press, 1965.

[28] M. Pharr, W. Jakob, and G. Humphreys, *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.

[29] Y. Dong, H. Zhang, and X. Zhang, “On impulse response modeling for underwater wireless optical mimo links,” in *2014 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, 2014, pp. 151–155.

[30] C. Mobley, “Light and water: Radiative transfer in natural waters’ academic press,” *San Diego, California*, 1994.

[31] V. I. Haltrin, “One-parameter two-term henyey-greenstein phase function for light scattering in seawater,” *Applied Optics*, vol. 41, no. 6, pp. 1022–1028, 2002.

[32] C. Gabriel, M.-A. Khalighi, S. Bourennane, P. Léon, and V. Rigaud, “Monte-carlo-based channel characterization for underwater optical communication systems,” *Journal of Optical Communications and Networking*, vol. 5, no. 1, pp. 1–12, 2013.

[33] V. I. Haltrin, "Two-term henyey-greenstein light scattering phase function for seawater," in *IEEE 1999 International Geoscience and Remote Sensing Symposium. IGARSS'99 (Cat. No. 99CH36293)*, vol. 2. IEEE, 1999, pp. 1423–1425.

[34] S. Twomey, H. Jacobowitz, and H. Howell, "Matrix methods for multiple-scattering problems," *Journal of Atmospheric Sciences*, vol. 23, no. 3, pp. 289–298, 1966.

[35] ——, "Light scattering by cloud layers," *Journal of Atmospheric Sciences*, vol. 24, no. 1, pp. 70–79, 1967.

[36] D. Collins and M. Wells, "Monte carlo codes for study of light transport in the atmosphere. vol. i, description of codes. us department of commerce," *Institute for Applied Technology*, 1965.

[37] D. G. Collins and M. B. Wells, "Monte carlo codes for study of light transport in the atmosphere. volume 2: Utilization instructions," RADIATION RESEARCH ASSOCIATES INC FORT WORTH TX, Tech. Rep., 1965.

[38] G. N. Plass and G. W. Kattawar, "Influence of single scattering albedo on reflected and transmitted light from clouds," *Applied Optics*, vol. 7, no. 2, pp. 361–367, 1968.

[39] ——, "Monte carlo calculations of light scattering from clouds," *Applied optics*, vol. 7, no. 3, pp. 415–419, 1968.

[40] G. W. Kattawar and G. N. Plass, "Influence of particle size distribution on reflected and transmitted light from clouds," *Applied optics*, vol. 7, no. 5, pp.

869–878, 1968.

[41] ——, "Radiance and polarization of multiple scattered light from haze and clouds," *Applied optics*, vol. 7, no. 8, pp. 1519–1527, 1968.

[42] G. N. Plass and G. W. Kattawar, "Radiative transfer in an atmosphere–ocean system," *Applied Optics*, vol. 8, no. 2, pp. 455–466, 1969.

[43] G. W. Kattawar and G. N. Plass, "Infrared cloud radiance," *Applied Optics*, vol. 8, no. 6, pp. 1169–1178, 1969.

[44] G. N. Plass and G. W. Kattawar, "Reflection of light pulses from clouds," *Applied optics*, vol. 10, no. 10, pp. 2304–2310, 1971.

[45] G. W. Kattawar and G. N. Plass, "Radiative transfer in the earth's atmosphere-ocean system: Ii. radiance in the atmosphere and ocean," *Journal of Physical Oceanography*, vol. 2, no. 2, pp. 146–156, 1972.

[46] H. R. Gordon and O. B. Brown, "Irradiance reflectivity of a flat ocean as a function of its optical properties," *Applied Optics*, vol. 12, no. 7, pp. 1549–1551, 1973.

[47] ——, "Influence of bottom depth and albedo on the diffuse reflectance of a flat homogeneous ocean," *Applied Optics*, vol. 13, no. 9, pp. 2153–2159, 1974.

[48] H. R. Gordon, O. B. Brown, and M. M. Jacobs, "Computed relationships between the inherent and apparent optical properties of a flat homogeneous ocean," *Applied optics*, vol. 14, no. 2, pp. 417–427, 1975.

[49] H. R. Gordon and O. B. Brown, “Diffuse reflectance of the ocean: some effects of vertical structure,” *Applied Optics*, vol. 14, no. 12, pp. 2892–2895, 1975.

[50] T. J. Petzold, “Volume scattering functions for selected ocean waters,” Scripps Institution of Oceanography La Jolla Ca Visibility Lab, Tech. Rep., 1972.

[51] L. R. Poole, D. D. Venable, and J. W. Campbell, “Semianalytic monte carlo radiative transfer model for oceanographic lidar systems,” *Applied optics*, vol. 20, no. 20, pp. 3653–3656, 1981.

[52] J. T. O. Kirk, “Monte carlo procedure for simulating the penetration of light into natural waters,” *Technical Paper-Commonwealth Scientific and Industrial Research Organization*, 1981.

[53] J. Kirk, “Monte carlo study of the nature of the underwater light field in, and the relationships between optical properties of, turbid yellow waters,” *Marine and Freshwater Research*, vol. 32, no. 4, pp. 517–532, 1981.

[54] R. M. Lerner and J. D. Summers, “Monte carlo description of time-and space-resolved multiple forward scatter in natural water,” *Applied optics*, vol. 21, no. 5, pp. 861–869, 1982.

[55] L. Johnson, R. Green, and M. Leeson, “A survey of channel models for underwater optical wireless communication,” in *2013 2nd International Workshop on Optical Wireless Communications (IWOW)*. IEEE, 2013, pp. 1–5.

[56] L. K. Gkoura, G. D. Roumelas, H. E. Nistazakis, H. G. Sandalidis, A. Vavoulas, A. D. Tsigopoulos, and G. S. Tombras, “Underwater optical

wireless communication systems: A concise review," *Turbulence Modelling ApproachesCurrent State, Development Prospects, Applications*, 2017.

[57] Z. Zeng, S. Fu, H. Zhang, Y. Dong, and J. Cheng, "A survey of underwater optical wireless communications," *IEEE communications surveys & tutorials*, vol. 19, no. 1, pp. 204–238, 2016.

[58] N. Saeed, A. Celik, T. Y. Al-Naffouri, and M.-S. Alouini, "Underwater optical wireless communications, networking, and localization: A survey," *Ad Hoc Networks*, vol. 94, p. 101935, 2019.

[59] M. Jouhari, K. Ibrahimi, H. Tembine, and J. Ben-Othman, "Underwater wireless sensor networks: A survey on enabling technologies, localization protocols, and internet of underwater things," *IEEE Access*, vol. 7, pp. 96 879–96 899, 2019.

[60] S. Li, W. Qu, C. Liu, T. Qiu, and Z. Zhao, "Survey on high reliability wireless communication for underwater sensor networks," *Journal of Network and Computer Applications*, vol. 148, p. 102446, 2019.

[61] T. Islam and S.-H. Park, "A comprehensive survey of the recently proposed localization protocols for underwater sensor networks," *IEEE Access*, vol. 8, pp. 179 224–179 243, 2020.

[62] G. Schirripa Spagnolo, L. Cozzella, and F. Leccese, "Underwater optical wireless communications: Overview," *Sensors*, vol. 20, no. 8, p. 2261, 2020.

[63] S. Jaruwatanadilok, "Underwater wireless optical communication channel modeling and performance evaluation using vector radiative transfer theory,"

IEEE Journal on selected areas in communications, vol. 26, no. 9, pp. 1620–1627, 2008.

[64] C. Gabriel, M.-A. Khalighi, S. Bourennane, P. Leon, and V. Rigaud, "Channel modeling for underwater optical communication," in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*. IEEE, 2011, pp. 833–837.

[65] J. Li, Y. Ma, Q. Zhou, B. Zhou, and H. Wang, "Channel capacity study of underwater wireless optical communications links based on monte carlo simulation," *Journal of optics*, vol. 14, no. 1, p. 015403, 2011.

[66] ——, "Monte carlo study on pulse response of underwater optical channel," *Optical Engineering*, vol. 51, no. 6, pp. 066 001–066 001, 2012.

[67] F. Jasman and R. Green, "Monte carlo simulation for underwater optical wireless communications," in *2013 2nd international workshop on optical wireless communications (IWOW)*. IEEE, 2013, pp. 113–117.

[68] S. Tang, Y. Dong, and X. Zhang, "On path loss of nlos underwater wireless optical communication links," in *2013 MTS/IEEE OCEANS-Bergen*. IEEE, 2013, pp. 1–3.

[69] S. Tang, X. Zhang, and Y. Dong, "On impulse response for underwater wireless optical links," in *2013 MTS/IEEE OCEANS-Bergen*. IEEE, 2013, pp. 1–4.

[70] V. Guerra, O. El-Asmar, C. Suarez-Rodriguez, R. Perez-Jimenez, and J. Luna-Rivera, "Statistical study of the channel parameters in underwater wireless optical links," in *3rd IEEE International Work-Conference on Bioinspired Intelligence*. IEEE, 2014, pp. 124–127.

[71] W. Liu, D. Zou, P. Wang, Z. Xu, and L. Yang, "Wavelength dependent channel characterization for underwater optical wireless communications," in *2014 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. IEEE, 2014, pp. 895–899.

[72] S. Tang, Y. Dong, and X. Zhang, "Impulse response modeling for underwater wireless optical communication links," *IEEE transactions on communications*, vol. 62, no. 1, pp. 226–234, 2013.

[73] W. Cox and J. Muth, "Simulating channel losses in an underwater optical communication system," *JOSA A*, vol. 31, no. 5, pp. 920–934, 2014.

[74] W. Liu, D. Zou, Z. Xu, and J. Yu, "Non-line-of-sight scattering channel modeling for underwater optical wireless communication," in *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, 2015, pp. 1265–1268.

[75] Y. Dong, J. Liu, and S. Qian, "On impulse response of underwater optical wireless miso links with a uca of sources," in *OCEANS 2016-Shanghai*. IEEE, 2016, pp. 1–4.

[76] J. Liu, Y. Dong, and H. Zhang, "On received intensity for misaligned underwater wireless optical links," in *OCEANS 2016-Shanghai*. IEEE, 2016, pp. 1–4.

[77] F. Dong, L. Xu, D. Jiang, and T. Zhang, "Monte-carlo-based impulse response modeling for underwater wireless optical communication," *Progress in Electromagnetics Research M*, vol. 54, pp. 137–144, 2017.

[78] S. K. Sahu and P. Shanmugam, "A theoretical study on the impact of particle

scattering on the channel characteristics of underwater optical communication system," *Optics Communications*, vol. 408, pp. 3–14, 2018.

[79] F. Miramirkhani and M. Uysal, "Visible light communication channel modeling for underwater environments with blocking and shadowing," *IEEE Access*, vol. 6, pp. 1082–1090, 2017.

[80] Z. Yingluo, W. Yingmin, and H. Aiping, "Analysis of underwater laser transmission characteristics under monte carlo simulation," in *2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO)*. IEEE, 2018, pp. 1–5.

[81] C. T. Geldard, J. Thompson, and W. O. Popoola, "A study of spatial and temporal dispersion in turbulent underwater optical wireless channel," in *2019 15th International Conference on Telecommunications (ConTEL)*. IEEE, 2019, pp. 1–5.

[82] A.-A. B. Umar, M. S. Leeson, and I. Abdullahi, "Modelling impulse response for nlos underwater optical wireless communications," in *2019 15th International Conference on Electronics, Computer and Computation (ICECCO)*. IEEE, 2019, pp. 1–6.

[83] B. Majlesein, A. Gholami, and Z. Ghassemlooy, "The channel impulse response of simo underwater optical wireless communication link based on monte carlo simulation," in *2019 2nd West Asian Colloquium on Optical Wireless Communications (WACOWC)*. IEEE, 2019, pp. 69–73.

[84] D. Liu, P. Xu, Y. Zhou, W. Chen, B. Han, X. Zhu, Y. He, Z. Mao, C. Le, P. Chen *et al.*, "Lidar remote sensing of seawater optical properties: experiment and monte carlo simulation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 9489–9498, 2019.

[85] Z. Zhou, H. Yin, and Y. Yao, "Wireless optical communication performance simulation and full-duplex communication experimental system with different seawater environment," in *2019 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. IEEE, 2019, pp. 1–4.

[86] J. Qin, M. Fu, M. Sun, C. Zhen, R. Ji, and B. Zheng, "Simulation of beam characteristics in long-distance underwater optical communication," in *Global Oceans 2020: Singapore–US Gulf Coast*. IEEE, 2020, pp. 1–5.

[87] W. Wang, X. Li, S. Rajbhandari, and Y. Li, "Investigation of 3 db optical intensity spot radius of laser beam under scattering underwater channel," *Sensors*, vol. 20, no. 2, p. 422, 2020.

[88] A. S. Ghazy, S. Hranilovic, and M.-A. Khalighi, "Angular mimo for underwater wireless optical communications: Link modeling and tracking," *IEEE Journal of Oceanic Engineering*, vol. 46, no. 4, pp. 1391–1407, 2021.

[89] E. Illi, F. El Bouanani, and F. Ayoub, "A high accuracy solver for rte in underwater optical communication path loss prediction," in *2018 International Conference on Advanced Communication Technologies and Networking (CommNet)*. IEEE, 2018, pp. 1–8.

[90] E. Illi, F. El Bouanani, K.-H. Park, F. Ayoub, and M.-S. Alouini, "An improved accurate solver for the time-dependent rte in underwater optical wireless communications," *IEEE Access*, vol. 7, pp. 96 478–96 494, 2019.

[91] H. Gao and H. Zhao, "A fast-forward solver of radiative transfer equation," *Transport Theory and Statistical Physics*, vol. 38, no. 3, pp. 149–192, 2009.

[92] H. Ding, Z. Xu, and B. M. Sadler, "A path loss model for non-line-of-sight ultraviolet multiple scattering channels," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, pp. 1–12, 2010.

[93] H. Zhang, Y. Dong, and X. Zhang, "On stochastic model for underwater wireless optical links," in *2014 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, 2014, pp. 156–160.

[94] Y. Zhou and Y. Dong, "Single scattering impulse response modeling of underwater wireless optical channels," in *OCEANS 2016-Shanghai*. IEEE, 2016, pp. 1–4.

[95] R. Yuan, J. Ma, P. Su, Y. Dong, and J. Cheng, "Monte-carlo integration models for multiple scattering based optical wireless communication," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 334–348, 2019.

[96] D. P. Kroese, T. Taimre, and Z. I. Botev, *Handbook of monte carlo methods*. John Wiley & Sons, 2013.

[97] W. A. Jakob, *Light transport on path-space manifolds*. Cornell University, 2013.

[98] R. A. Leathers and N. J. McCormick, "Algorithms for ocean-bottom albedo determination from in-water natural-light measurements," *Applied optics*, vol. 38, no. 15, pp. 3199–3205, 1999.

[99] M. F. Ali, D. N. K. Jayakody, Y. A. Chursin, S. Affes, and S. Dmitry, "Recent advances and future directions on underwater wireless communications,"

Archives of Computational Methods in Engineering, vol. 27, pp. 1379–1412, 2020.

[100] M. Jahanbakht, W. Xiang, L. Hanzo, and M. R. Azghadi, “Internet of underwater things and big marine data analyticsa comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 904–956, 2021.

[101] X. Sun, C. H. Kang, M. Kong, O. Alkhazragi, Y. Guo, M. Ouhssain, Y. Weng, B. H. Jones, T. K. Ng, and B. S. Ooi, “A review on practical considerations and solutions in underwater wireless optical communication,” *Journal of Lightwave Technology*, vol. 38, no. 2, pp. 421–431, 2020.

[102] A. M. Vegni, M. Hammouda, and V. Loscrí, “A vlc-based footprinting localization algorithm for internet of underwater things in 6g networks,” in *2021 17th International symposium on wireless communication systems (ISWCS)*. IEEE, 2021, pp. 1–6.

[103] C. D. Mobley, B. Gentili, H. R. Gordon, Z. Jin, G. W. Kattawar, A. Morel, P. Reinersman, K. Stamnes, and R. H. Stavn, “Comparison of numerical models for computing underwater light fields,” *Applied Optics*, vol. 32, no. 36, pp. 7484–7504, 1993.

[104] M. Uysal, S. Ghasvarianjahromi, M. Karbalayghareh, P. D. Diamantoulakis, G. K. Karagiannidis, and S. M. Sait, “Slipt for underwater visible light communications: Performance analysis and optimization,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6715–6728, 2021.

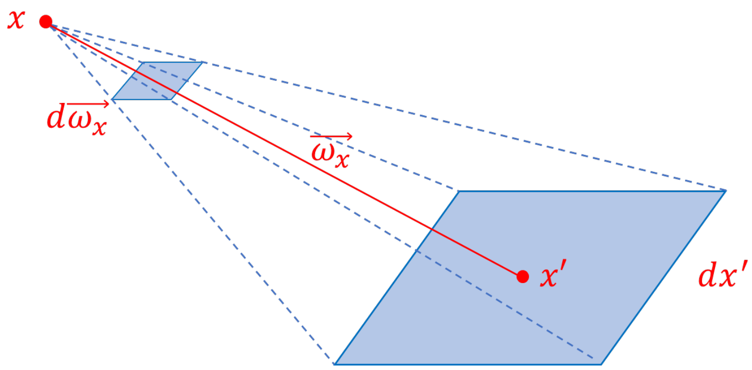
[105] A. M. Ghonim, W. M. Salama, A. E.-R. A. El-Fikky, A. A. Khalaf, and H. M. Shalaby, “Underwater localization system based on visible-light

communications using neural networks," *Applied Optics*, vol. 60, no. 13, pp. 3977–3988, 2021.

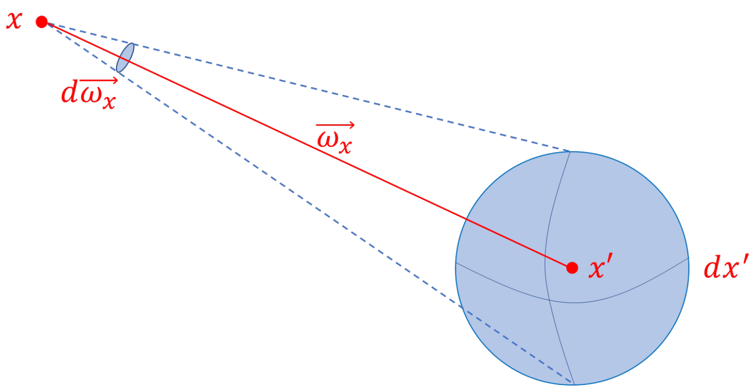
[106] D. Nivedhitha, D. Karthik *et al.*, "Localization systems for autonomous operation of underwater robotic vehicles: A survey," in *OCEANS 2022-Chennai*. IEEE, 2022, pp. 1–8.

APPENDICES

Appendix A: A Slid Angle between Elementary Points



(a) direction and a surface point



(b) direction and a volume point

Figure A.1: link between a direction and (a) a surface point or (b) a volume point.

The integrals manipulated into the optical channel modelling rely on three different kinds of domain or space: directions $\in \Omega$, surface points $\in A$, and volume points $\in V$. It is possible to move from one domain to another, using some simple transformations. The solid angle $\mathrm{d}\overrightarrow{\omega_x}$ corresponding to an elementary surface point $\mathrm{d}x'$ seen from a given position x is the most well known (see Figure A.1a). By correcting the solid angle via a cosine to account for the tilt of the surface $\mathrm{d}x'$, it is expressed using the following two useful equations:

$$\mathrm{d}\overrightarrow{\omega_x} = \frac{|\overrightarrow{\omega} \cdot \overrightarrow{n}|}{\|\overrightarrow{x'x}\|^2}\mathrm{d}x' \quad \text{or} \quad \mathrm{d}x' = \frac{\|\overrightarrow{x'x}\|^2}{|\overrightarrow{\omega} . \overrightarrow{n}|}\mathrm{d}\overrightarrow{\omega_x}. \tag{A.1}$$

The solid angle $\mathrm{d}\overrightarrow{\omega_x}$ corresponding to an elementary volume point $\mathrm{d}x'$ seen from position x is very similar to the surface case, except the cosine that disappears (see Figure A.1b). Indeed, the elementary volume point has no surface, being just a volume. It leads to the following two useful equations:

$$\mathrm{d}\overrightarrow{\omega_x} = \frac{1}{\|\overrightarrow{x'x}\|^2}\mathrm{d}x' \quad \text{or} \quad \mathrm{d}x' = \|\overrightarrow{x'x}\|^2\mathrm{d}\overrightarrow{\omega_x}. \tag{A.2}$$

Appendix B: The monotonicity of the function $\varphi(\theta)$

According to Equation (4.3), it is difficult to know the monotonicity of the function $\varphi(\theta)$ for an interval $[\theta^-, \theta^+]$, where θ^- and θ^+ follow as Equation (4.2). So we need to compute $\varphi'(\theta)$, the derivative of $\varphi(\theta)$. For this purpose, we first recall here some rules:

- The composition rule: $f'(g(x)) = f'(g(x)) \times g'(x)$.
- The quotient rule: $(f(x)/g(x))' = (f'(x)g(x) - f(x)g'(x))/g^2(x)$
- The derivative of $\cos^{-1}(x)$ is $-(1-x^2)^{-\frac{1}{2}}$.
- The derivative of $\cos(x)$ is $-\sin(x)$, and the derivative of $\sin(x)$ is $\cos(x)$.

So, we will use the following functions:

- $u(\theta) = \cos\alpha - \cos\theta_0 \cos\theta$ with derivative $u'(\theta) = \cos\theta_0 \sin\theta$.
- $v(\theta) = \sin\theta_0 \sin\theta$ with derivative $v'(\theta) = \sin\theta_0 \cos\theta$.
- $g(\theta) = \frac{u(\theta)}{v(\theta)} = \frac{\cos\alpha - \cos\theta_0\cos\theta}{\sin\theta_0 \sin\theta}$. Its derivative is obtained from the quotient rule:

$$
\begin{aligned}
g'(\theta) &= \frac{v(\theta)u'(\theta) - u(\theta)v'(\theta)}{v^2(\theta)} \\
&= \frac{\sin\theta_0 \sin\theta \cos\theta_0 \sin\theta - (\cos\alpha - \cos\theta_0\cos\theta)\sin\theta_0\cos\theta}{\sin^2\theta_0 \sin^2\theta} \\
&= \frac{\sin^2\theta\cos\theta_0 - \cos\alpha\cos\theta + \cos\theta_0\cos^2\theta}{\sin\theta_0\sin^2\theta} \\
&= \frac{\cos\theta_0 - \cos\alpha\cos\theta}{\sin\theta_0\sin^2\theta}
\end{aligned}
$$

- $f(x) = \cos^{-1}(x)$ with derivative $f'(x) = -(1-x^2)^{-1/2}$.

Then, $\varphi(\theta) = f(g(\theta))$. Let us compute its derivative thanks to the composition rule:

$$
\begin{aligned}
\varphi'(\theta) &= f'(g(\theta)) \times g'(\theta) \\
&= -\left(1 - \left(\frac{\cos\alpha - \cos\theta_0\cos\theta}{\sin\theta_0\sin\theta}\right)^2\right)^{-\frac{1}{2}} \times \frac{\cos\theta_0 - \cos\alpha\cos\theta}{\sin\theta_0\sin^2\theta} \\
&= -\frac{\cos\theta_0 - \cos\alpha\cos\theta}{\sin\theta_0\sin^2\theta\sqrt{1 - \frac{(\cos\alpha - \cos\theta_0\cos\theta)^2}{(\sin\theta_0\sin\theta)^2}}} \\
&= -\frac{\sin\theta_0\sin\theta(\cos\theta_0 - \cos\alpha\cos\theta)}{\sin\theta_0\sin^2\theta\sqrt{\sin^2\theta_0\sin^2\theta - (\cos\alpha - \cos\theta_0\cos\theta)^2}} \\
&= -\frac{\cos\theta_0 - \cos\alpha\cos\theta}{\sin\theta\sqrt{\sin^2\theta_0\sin^2\theta - (\cos\alpha - \cos\theta_0\cos\theta)^2}}
\end{aligned}
$$

Assuming $0 < \theta < \pi$, the denominator is always positive if it exists. Let us check the existence of the denominator. We need to verify the positiveness of the square root parameter:

$$\begin{aligned}
0 &\leq \sin^2\theta_0 \sin^2\theta - (\cos\alpha - \cos\theta_0\cos\theta)^2 \\
0 &\leq \sin^2\theta_0 \sin^2\theta - \cos^2\alpha - \cos^2\theta_0\cos^2\theta + 2\cos\alpha\cos\theta_0\cos\theta \\
0 &\leq \sin^2\theta_0 \sin^2\theta - \cos^2\alpha - (1-\sin^2\theta_0)\cos^2\theta + 2\cos\alpha\cos\theta_0\cos\theta \\
0 &\leq \sin^2\theta_0 - \cos^2\alpha - \cos^2\theta + 2\cos\alpha\cos\theta_0\cos\theta \\
0 &\leq \sin^2\theta_0 - \cos^2\alpha - (\cos\theta - \cos\alpha\cos\theta_0)^2 + \cos^2\alpha\cos^2\theta_0 \\
0 &\leq \sin^2\theta_0 - \cos^2\alpha\sin^2\theta_0 - (\cos\theta - \cos\alpha\cos\theta_0)^2 \\
0 &\leq \sin^2\theta_0 \sin^2\alpha - (\cos\theta - \cos\alpha\cos\theta_0)^2 \\
(\cos\theta - \cos\alpha\cos\theta_0)^2 &\leq \sin^2\theta_0 \sin^2\alpha
\end{aligned}$$

Assuming $0 < \theta < \pi$, and also $\theta_0 - \alpha < \theta < \alpha + \theta_0$, the bounds for $\cos\theta$ are $\cos(\theta_0 - \alpha) = \cos\alpha\cos\theta_0 + \sin\alpha\sin\theta_0$ and $\cos(\alpha+\theta_0) = \cos\alpha\cos\theta_0 - \sin\alpha\sin\theta_0$. Since $\alpha \in [0,\pi]$, it follows:

$$\begin{array}{ccccc}
\cos(\alpha+\theta_0) & \leq & \cos\theta & \leq & \cos(\theta_0-\alpha) \\
-\sin\alpha\sin\theta_0 & \leq & \cos\theta - \cos\alpha\cos\theta_0 & \leq & \sin\alpha\sin\theta_0
\end{array}$$

This proves the positiveness of the square root parameter.

The denominator is always positive, so the roots are given using the numerator only:

$$\begin{aligned}
0 &= -\cos\theta_0 + \cos\alpha\cos\theta \\
\Rightarrow \cos\theta &= \frac{\cos\theta_0}{\cos\alpha} \\
\Rightarrow \theta &= \cos^{-1}\left(\frac{\cos\theta_0}{\cos\alpha}\right)
\end{aligned}$$

Clearly, there is a single root only if $\frac{\cos\theta_0}{\cos\alpha} \in [-1,1]$. In the other case, then the function is monotonic onto the domain $[\theta_0 - \alpha, \theta_0 + \alpha]$.

Appendix C: Flowchart Monte Carlo simulation steps

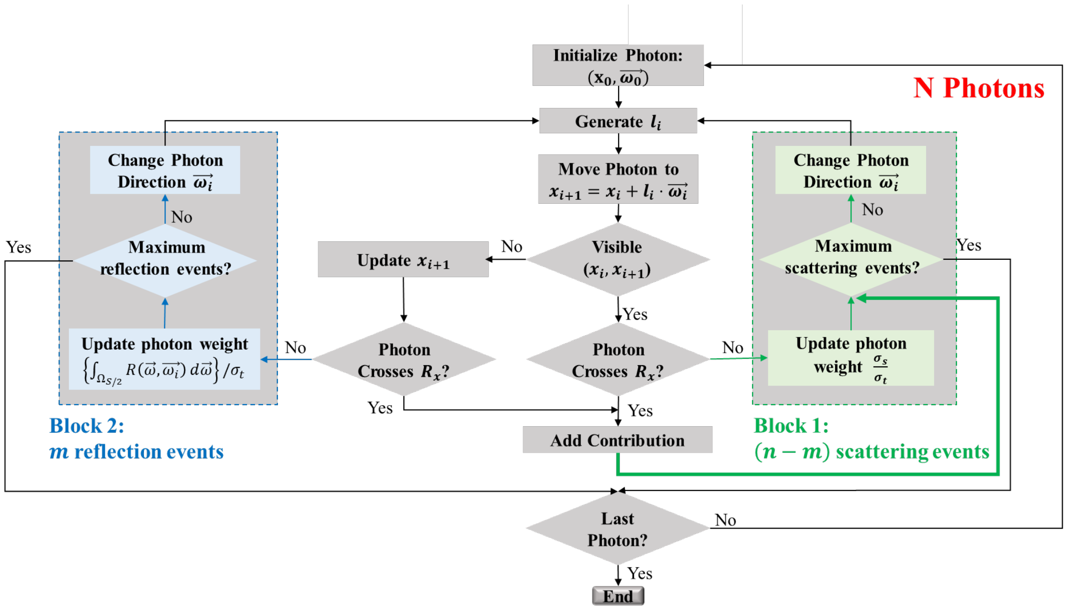


Figure C.1: Flowchart for MC simulation corresponding to Xiao 1-R and Xiao 2-R algorithms

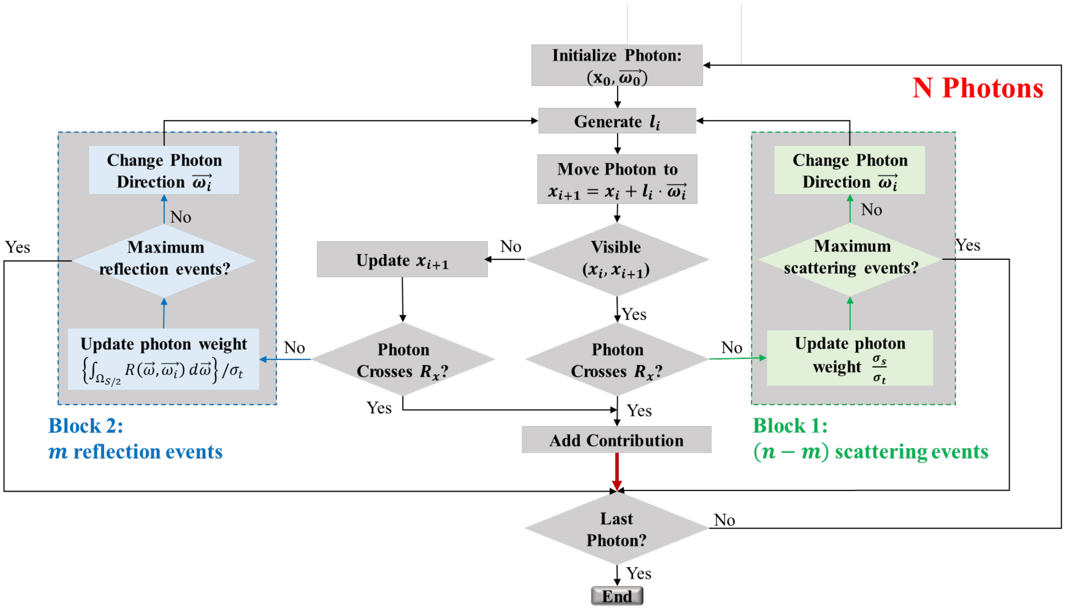


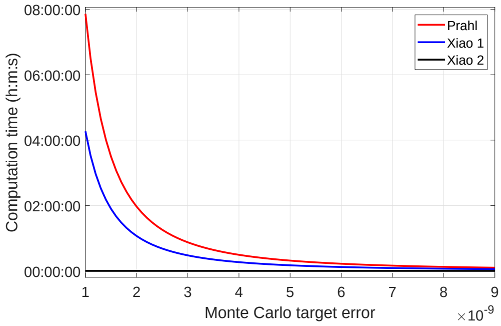
Figure C.2: Flowchart for MC simulation corresponding to Prahl's algorithm

Appendix D: Simulation results for Pure sea

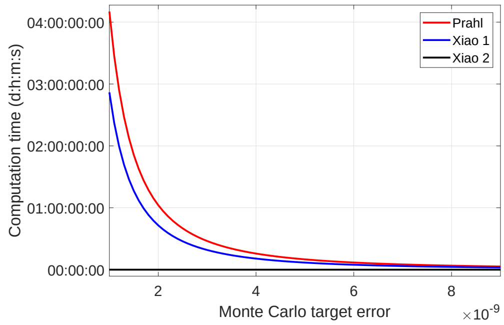
D.1: Changing receiver's area

Table D.1: Pure sea - unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the receiver's radius (in cm).

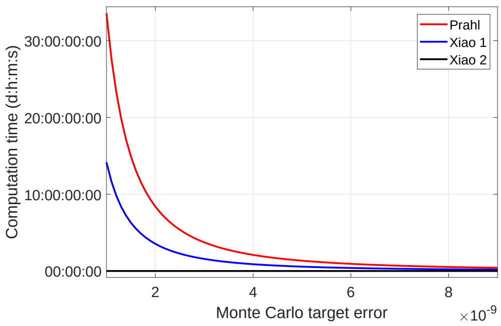
Radius (cm)	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
0.5	$2.9812E^{-6}$	$2.8520E^{-6}$	$1.4917E^{-6}$	52.30%
2	$4.7715E^{-5}$	$4.6817E^{-5}$	$2.3877E^{-5}$	51.00%
5	$2.9783E^{-4}$	$2.9121E^{-4}$	$1.4911E^{-4}$	51.20%
8	$7.6272E^{-4}$	$7.4547E^{-4}$	$3.8185E^{-4}$	51.22%
20	$4.7663E^{-3}$	$4.6429E^{-3}$	$2.3760E^{-3}$	51.17%
50	$2.9775E^{-2}$	$2.8260E^{-2}$	$1.4217E^{-2}$	50.31 %



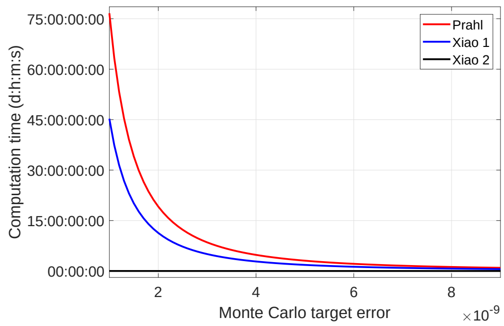
(a) Diameter = 0.5cm



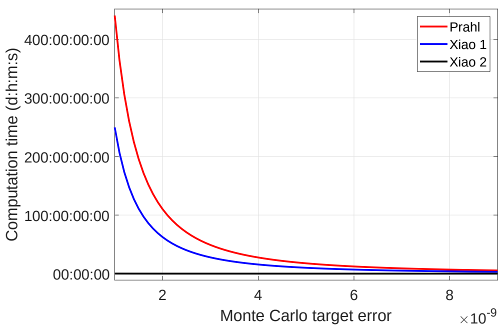
(b) Diameter = 2cm



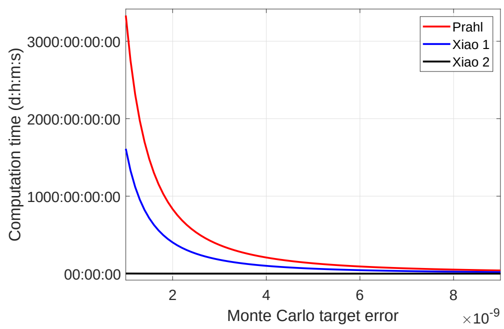
(c) Diameter = 5cm



(d) Diameter = 8cm



(e) Diameter = 20cm



(f) Diameter = 50cm

Figure D.1: Expected computation time varying the target error for Pure sea with different receiver's area.

Table D.2: Pure sea - unbiased sample variances obtained from algorithm Xiao 2 varying the receiver's radius (in cm).

Radius (cm)	Algorithm Xiao 2	Ratio 1	Ratio 2
0.5	$4.9632E^{-12}$	0.0002%	0.0003%
2	$5.1114E^{-10}$	0.0011%	0.0021%
5	$8.2507E^{-9}$	0.0028%	0.0055%
8	$3.3564E^{-8}$	0.0045%	0.0088%
20	$5.0532E^{-7}$	0.0109%	0.0217%
50	$7.1237E^{-6}$	0.0252%	0.05108%

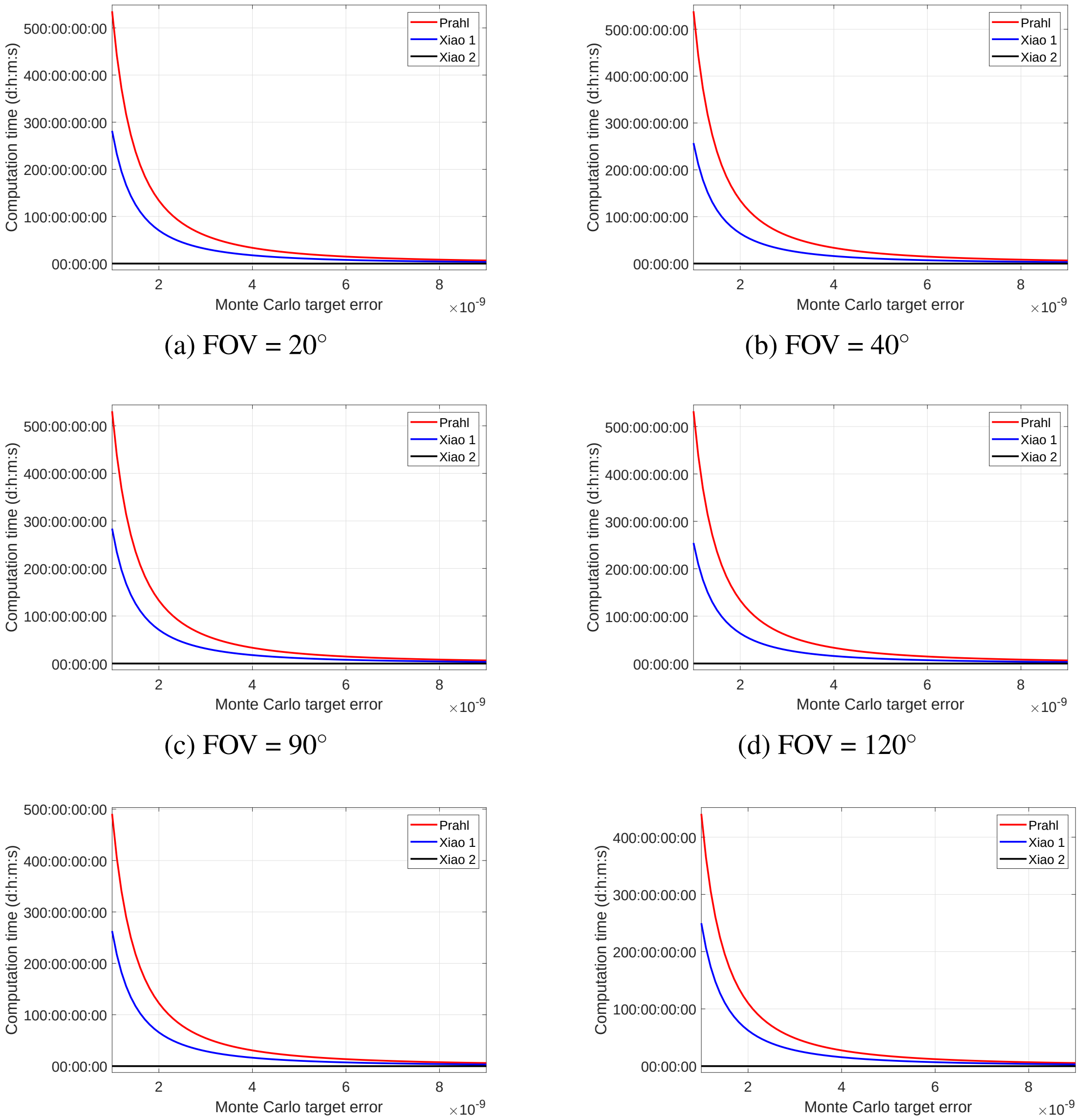
D.2: Changing receiver's fov

Table D.3: Pure sea - unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the receiver's FOV.

FOV	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
20°	$4.7542E^{-3}$	$4.6437E^{-3}$	$2.3737E^{-3}$	51.12%
40°	$4.7640E^{-3}$	$4.6423E^{-3}$	$2.3759E^{-3}$	51.18%
90°	$4.7678E^{-3}$	$4.6441E^{-3}$	$2.3768E^{-3}$	51.18%
120°	$4.7653E^{-3}$	$4.6427E^{-3}$	$2.3754E^{-3}$	51.17%
150°	$4.7653E^{-3}$	$4.6429E^{-3}$	$2.3754E^{-3}$	51.16%
180°	$4.7663E^{-3}$	$4.6429E^{-3}$	$2.3760E^{-3}$	51.17%

Table D.4: Pure sea - unbiased sample variances obtained from algorithm Xiao 2 varying the receiver's FOV.

FOV	Algorithm Xiao 2	Ratio 1	Ratio 2
20°	$4.3774E^{-7}$	0.0094%	0.0184%
40°	$4.8760E^{-7}$	0.0105%	0.0205%
90°	$5.0423E^{-7}$	0.0109%	0.0212%
120°	$5.0431E^{-7}$	0.0109%	0.0212%
150°	$5.0584E^{-7}$	0.0109%	0.0213%
180°	$5.0532E^{-7}$	0.0109%	0.0217%



(a) FOV = 20°

(b) FOV = 40°

(c) FOV = 90°

(d) FOV = 120°

(e) FOV = 150°

(f) FOV = 180°

Figure D.2: Expected computation time varying the target error for Pure sea with different receiver's FOV.

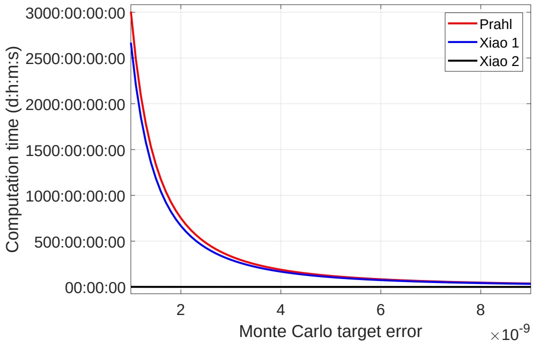
D.3: Changing the distance between T_x and R_x

Table D.5: Pure sea - unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the distance between T_x and R_x.

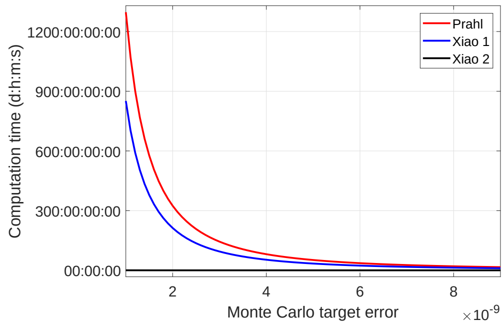
Distance (m)	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
5	$4.0084E^{-2}$	$3.8124E^{-2}$	$2.8592E^{-2}$	75.00%
8	$1.3310E^{-2}$	$1.2948E^{-2}$	$8.2584E^{-3}$	63.78%
12	$4.7663E^{-3}$	$4.6429E^{-3}$	$2.3760E^{-3}$	51.17%
16	$2.1596E^{-3}$	$2.0933E^{-3}$	$8.5846E^{-4}$	41.01%
20	$1.1137E^{-3}$	$1.0710E^{-3}$	$3.5231E^{-4}$	32.90%

Table D.6: Pure sea - unbiased sample variances obtained from algorithm Xiao 2 varying the distance between T_x and R_x.

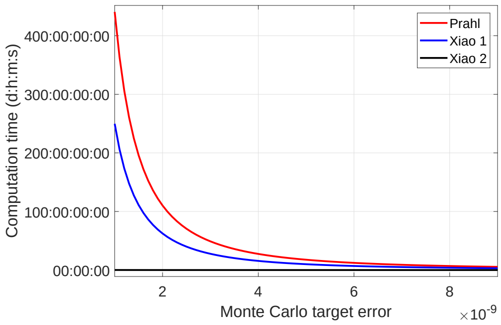
Distance (m)	Algorithm Xiao 2	Ratio 1	Ratio 2
5	$4.0256E^{-6}$	0.0106%	0.0141%
8	$1.3876E^{-6}$	0.0107%	0.0168%
12	$5.0532E^{-7}$	0.0109%	0.0217%
16	$2.3116E^{-7}$	0.0110%	0.0269%
20	$1.1856E^{-7}$	0.0111%	0.0337%



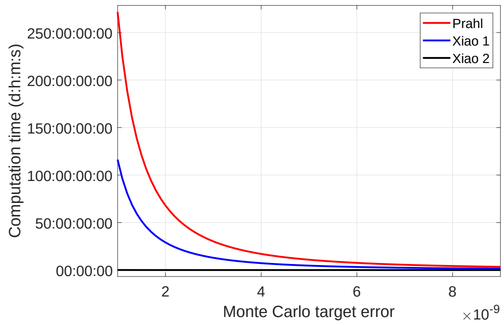
(a) Distance = 5m



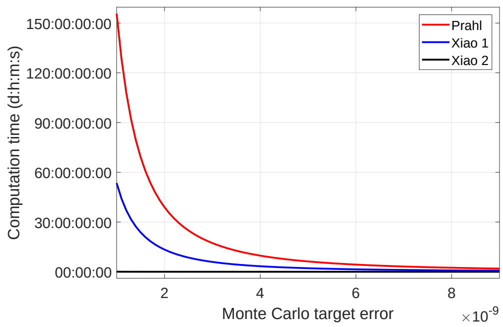
(b) Distance = 8m



(c) Distance = 12m



(d) Distance = 16m



(e) Distance = 20m

Figure D.3: Expected computation time varying the target error for Pure sea with different distance between R_x and T_x.

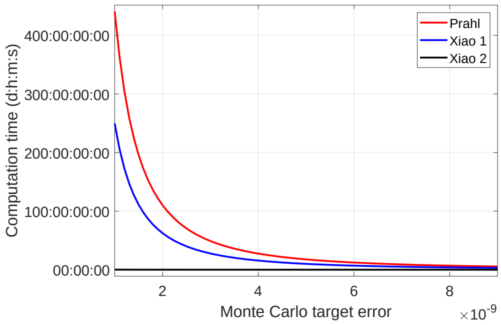
D.4: Changing the receiver's orientation

Table D.7: Pure sea - unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the receiver's orientation.

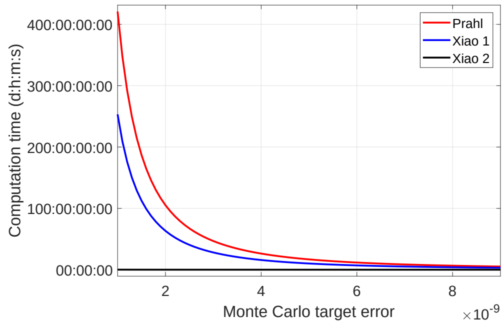
Orientation	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
0°	$4.7663E^{-3}$	$4.6429E^{-3}$	$2.3760E^{-3}$	51.17%
15°	$4.2311E^{-3}$	$4.1232E^{-3}$	$2.1106E^{-3}$	51.19%
30°	$4.1279E^{-3}$	$4.0231E^{-3}$	$2.0594E^{-3}$	51.19%
45°	$3.3706E^{-3}$	$3.2879E^{-3}$	$1.6832E^{-3}$	51.19%
60°	$2.3833E^{-3}$	$2.3247E^{-3}$	$1.1914E^{-3}$	51.25%
75°	$1.2337E^{-3}$	$1.2059E^{-3}$	$6.1712E^{-4}$	51.18%

Table D.8: Pure sea - unbiased sample variances obtained from algorithm Xiao 2 varying the receiver's orientation.

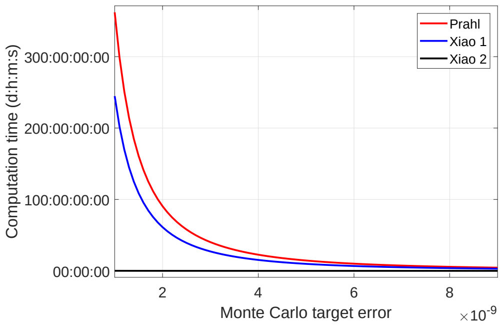
Orientation	Algorithm Xiao 2	Ratio 1	Ratio 2
0°	$5.0532E^{-7}$	0.0109%	0.0217%
15°	$4.6083E^{-7}$	0.0112%	0.0218%
30°	$4.5190E^{-7}$	0.0112%	0.0219%
45°	$3.7705E^{-7}$	0.0115%	0.0224%
60°	$2.7325E^{-7}$	0.0118%	0.0229%
75°	$1.4506E^{-7}$	0.0120%	0.0235%



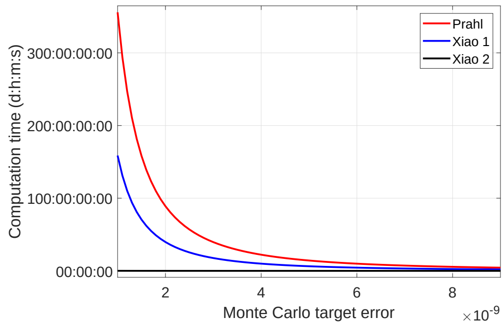
(a) Orientation = 0°



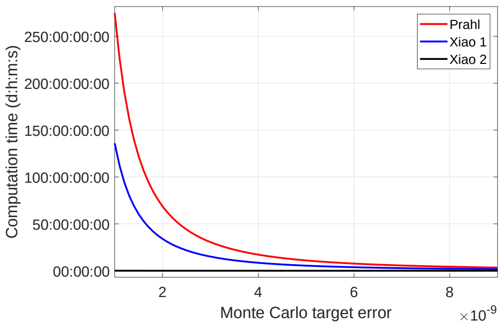
(b) Orientation = 15°



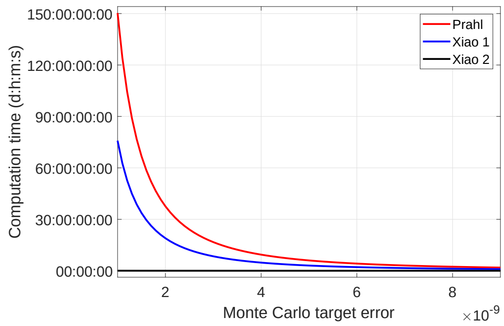
(c) Orientation = 30°



(d) Orientation = 45°



(e) Orientation = 60°



(f) Orientation = 75°

Figure D.4: Expected computation time varying the target error for Pure sea with different receiver's orientation.

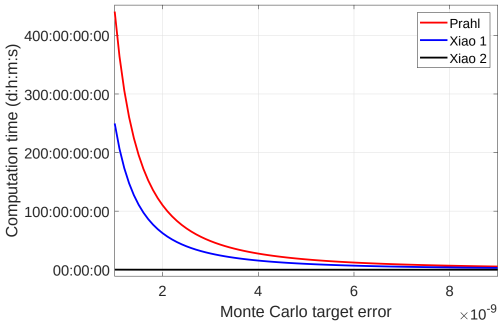
D.5: Changing the receiver's relation position to T_x

Table D.9: Pure sea - unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the receiver's relative position.

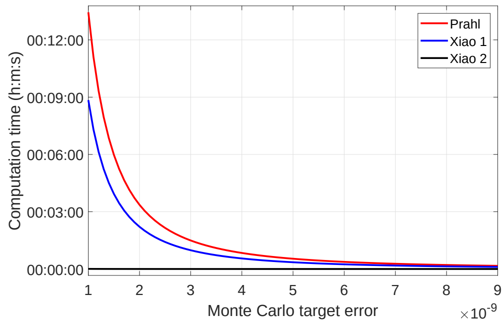
Rotation angle	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
0°	$4.7663E^{-3}$	$4.6429E^{-3}$	$2.3760E^{-3}$	51.17%
15°	$1.5321E^{-6}$	$7.9625E^{-8}$	$5.0692E^{-8}$	63.66%
30°	$1.7991E^{-7}$	$9.0931E^{-9}$	$5.6110E^{-9}$	61.71%
45°	$5.2057E^{-8}$	$2.6518E^{-9}$	$1.5364E^{-9}$	57.94%
60°	$2.1126E^{-8}$	$1.1104E^{-9}$	$5.9079E^{-10}$	53.20%
75°	$1.1039E^{-8}$	$5.9544E^{-10}$	$2.9475E^{-10}$	49.50%

Table D.10: Pure sea - unbiased sample variances obtained from algorithm Xiao 2 varying the receiver's relative position for Pure sea.

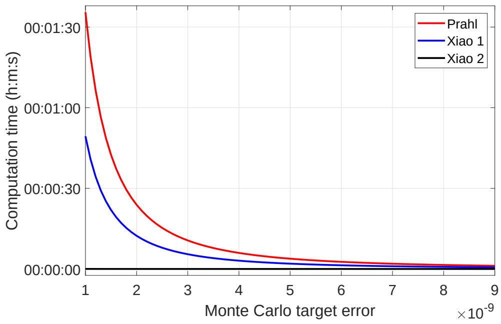
Rotation angle	Algorithm Xiao 2	Ratio 1	Ratio 2
0°	$5.0532E^{-7}$	0.0109%	0.0217%
15°	$2.0792E^{-11}$	0.0261%	0.0410%
30°	$1.1658E^{-12}$	0.0128%	0.0208%
45°	$3.2584E^{-13}$	0.0123%	0.0212%
60°	$1.2551E^{-13}$	0.0113%	0.0213%
75°	$5.6591E^{-14}$	0.0095%	0.0192%



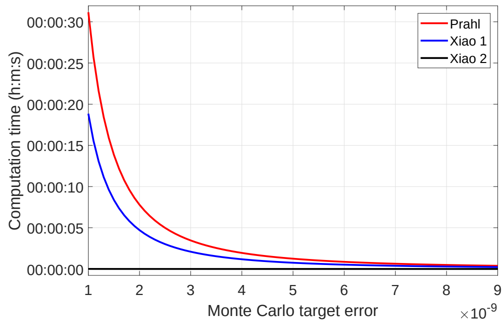
(a) Rotation angle = 0°



(b) Rotation angle = 15°

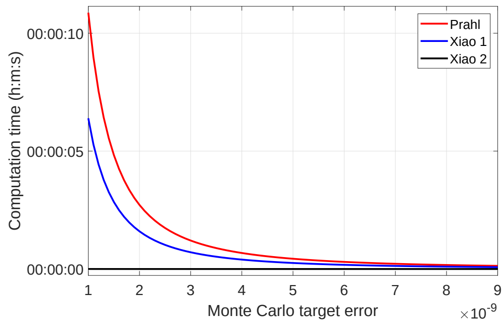


(c) Rotation angle = 30°



(d) Rotation angle = 45°

(e) Rotation angle = 60°



(f) Rotation angle = 75°

Figure D.5: Expected computation time varying the target error for Pure sea with different receiver's relative position to T_x.

D.6: Changing the albedo of sea ground

Table D.11: Pure Sea - Unbiased sample variance obtained from Prahl-R, Xiao 1-R and Xiao 2-R algorithms varying the albedo of sea ground.

Albedo	Channel's gain	Prahl-R	Xiao 1-R	Xiao 2-R
0.1	$3.7274E^{-5}$	$7.1072E^{-5}$	$3.6478E^{-5}$	$8.1248E^{-9}$
0.2	$3.7878E^{-5}$	$7.1404E^{-5}$	$3.6531E^{-5}$	$8.2047E^{-9}$
0.3	$3.8488E^{-5}$	$7.0970E^{-5}$	$3.7141E^{-5}$	$8.5898E^{-9}$
0.4	$3.9096E^{-5}$	$7.1828E^{-5}$	$3.7487E^{-5}$	$9.0933E^{-9}$
0.5	$3.9706E^{-5}$	$7.3112E^{-5}$	$3.8378E^{-5}$	$9.7815E^{-9}$
0.6	$4.0317E^{-5}$	$7.3396E^{-5}$	$3.8814E^{-5}$	$1.0247E^{-8}$
0.7	$4.0934E^{-5}$	$7.4418E^{-5}$	$3.9592E^{-5}$	$1.1226E^{-8}$
0.8	$4.1551E^{-5}$	$7.6170E^{-5}$	$4.0704E^{-5}$	$1.2248E^{-8}$
0.9	$4.2170E^{-5}$	$7.7210E^{-5}$	$4.1611E^{-5}$	$1.3222E^{-8}$

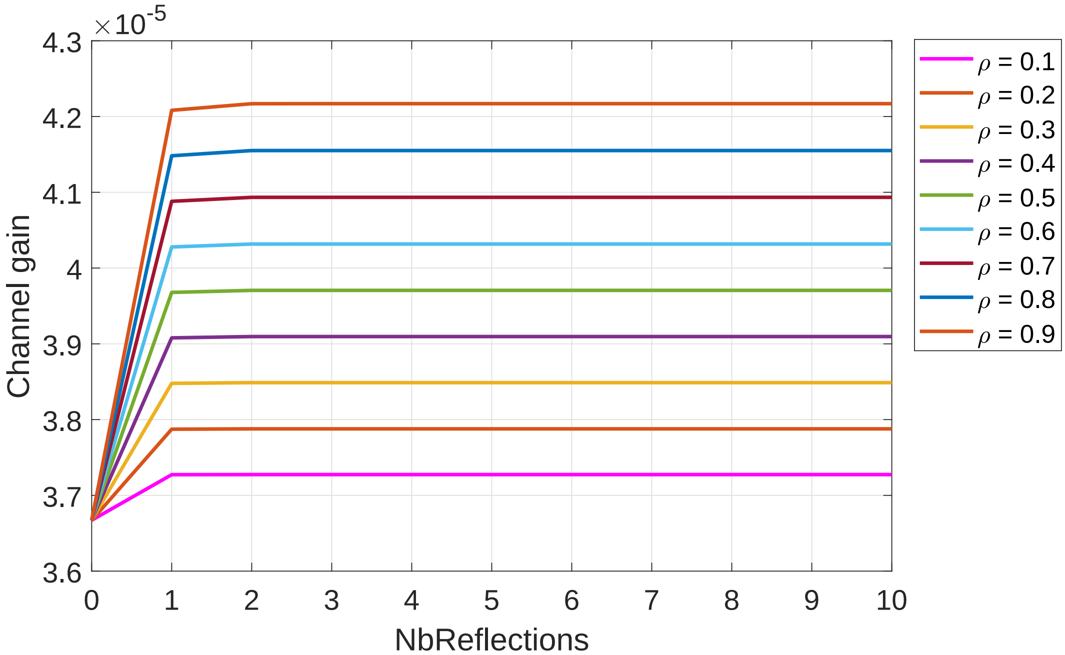


Figure D.6: Channel gain varying $NbReflections$ with different albedo of sea ground for Pure Sea

D.7: Changing the distance between sea ground and sensors

Table D.12: Pure Sea - Unbiased sample variance obtained from Prahl-R, Xiao 1-R and Xiao 2-R algorithms varying the distance between sensors and the sea ground.

Distance (m)	Channel's gain	Prahl-R	Xiao 1-R	Xiao 2-R
0.1	$3.6717E^{-5}$	$7.1468E^{-5}$	$3.6556E^{-5}$	$2.1823E^{-7}$
0.5	$3.7605E^{-5}$	$7.0877E^{-5}$	$3.6815E^{-5}$	$9.4720E^{-9}$
1	$3.8488E^{-5}$	$7.0970E^{-5}$	$3.7141E^{-5}$	$8.5898E^{-9}$
2	$3.9719E^{-5}$	$7.2614E^{-5}$	$3.7763E^{-5}$	$8.0430E^{-9}$
5	$3.9018E^{-5}$	$7.1898E^{-5}$	$3.7406E^{-5}$	$8.2305E^{-9}$
10	$3.6703E^{-5}$	$7.1086E^{-5}$	$3.6769E^{-5}$	$7.7538E^{-9}$
15	$3.6704E^{-5}$	$7.1397E^{-5}$	$3.6413E^{-5}$	$7.7275E^{-9}$
20	$3.6704E^{-5}$	$7.0949E^{-5}$	$3.6558E^{-5}$	$7.6697E^{-9}$
50	$3.6707E^{-5}$	$7.1108E^{-5}$	$3.6656E^{-5}$	$7.9541E^{-9}$

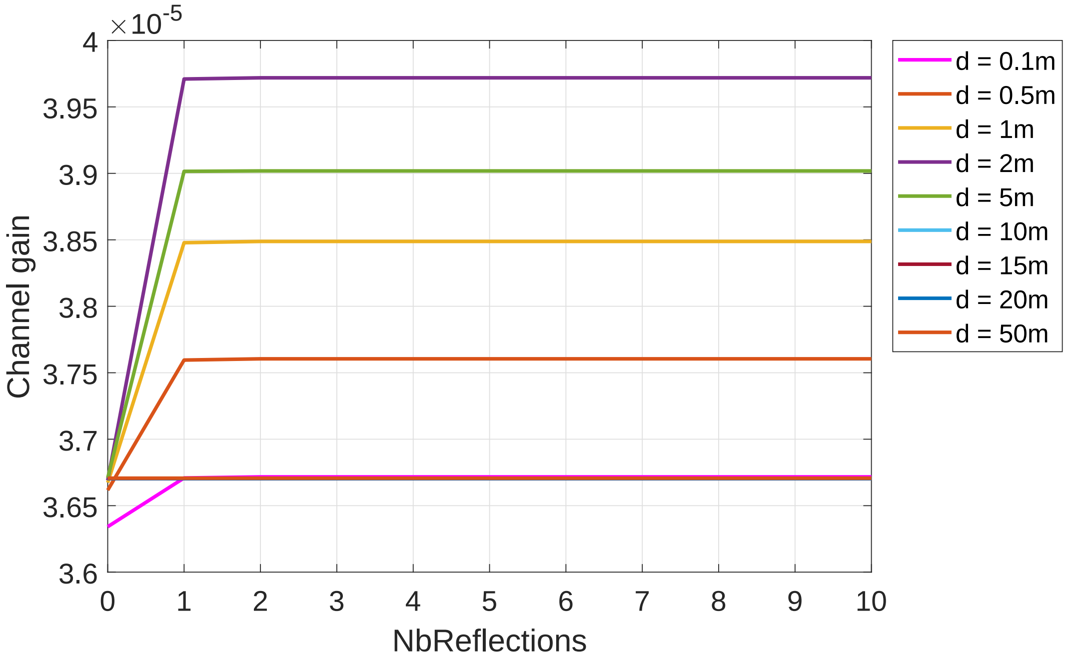


Figure D.7: Channel gain varying $NbReflections$ with different distance between sea ground and sensors for Pure Sea

Appendix E: Simulation results for Clear Ocean

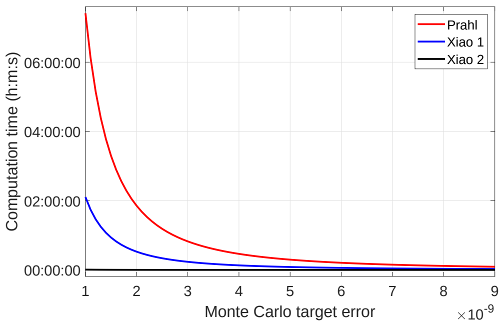
E.1: Changing receiver's area

Table E.1: Clear ocean - unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the receiver's radius (in cm).

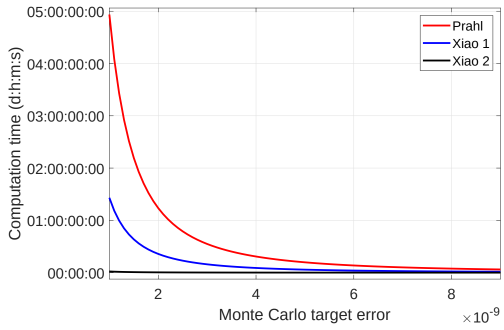
Radius (cm)	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
0.5	$1.7283E^{-6}$	$1.3353E^{-6}$	$3.6187E^{-7}$	27.10%
2	$2.7928E^{-5}$	$2.1334E^{-5}$	$5.9232E^{-6}$	27.76%
5	$1.7461E^{-4}$	$1.3220E^{-4}$	$3.7836E^{-5}$	28.62%
8	$4.4713E^{-4}$	$3.3737E^{-4}$	$9.8969E^{-5}$	29.34%
20	$2.7930E^{-3}$	$2.1041E^{-3}$	$6.6040E^{-4}$	31.39%
50	$1.7394E^{-2}$	$1.2868E^{-2}$	$4.4804E^{-3}$	34.82%

Table E.2: Clear ocean - unbiased sample variances obtained from algorithm Xiao 2 varying the receiver's radius (in cm).

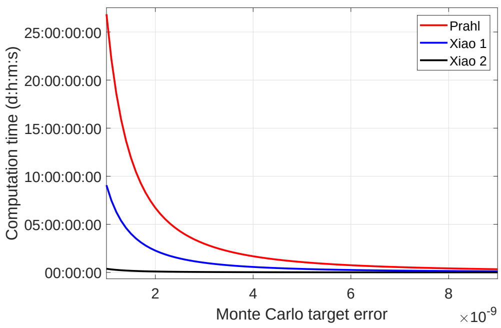
Radius (cm)	Algorithm Xiao 2	Ratio 1	Ratio 2
0.5	$1.0137E^{-9}$	0.08%	0.28%
2	$6.2690E^{-8}$	0.29%	1.06%
5	$1.0076E^{-6}$	0.76%	2.66%
8	$4.0771E^{-6}$	1.21%	4.12%
20	$6.1639E^{-5}$	2.93%	9.33%
50	$8.5647E^{-4}$	6.66%	19.12%



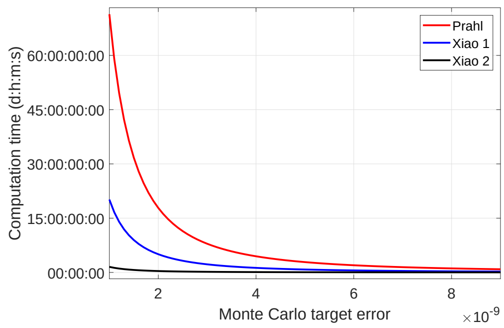
(a) Diameter = 0.5cm



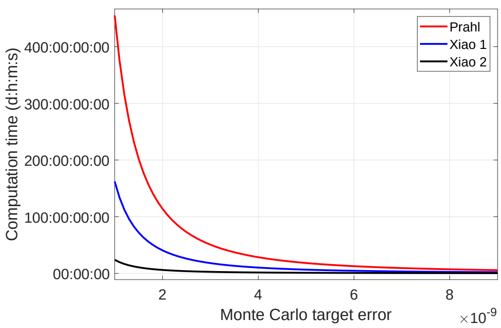
(b) Diameter = 2cm



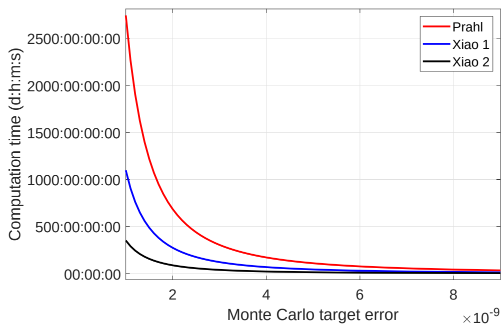
(c) Diameter = 5cm



(d) Diameter = 8cm



(e) Diameter = 20cm



(f) Diameter = 50cm

Figure E.1: Expected computation time varying the target error for Clear ocean with different receiver's area.

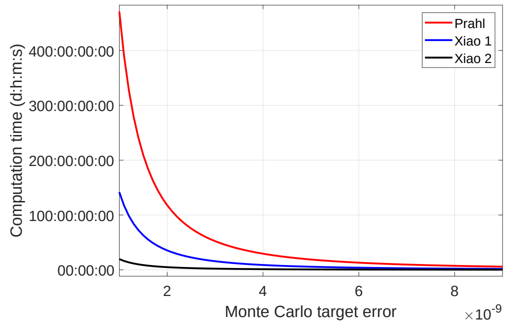
E.2: Changing receiver's FOV

Table E.3: Clear ocean - unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the receiver's FOV.

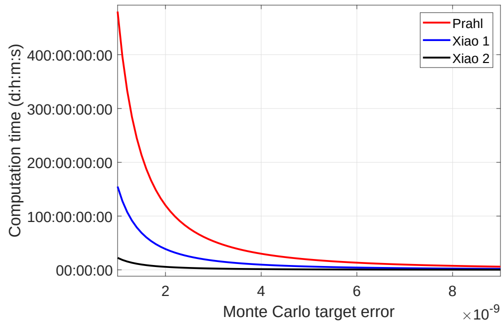
FOV	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
20°	$2.6211E^{-3}$	$2.0354E^{-3}$	$5.8873E^{-4}$	28.92%
40°	$2.7440E^{-3}$	$2.0849E^{-3}$	$6.3949E^{-4}$	30.67%
90°	$2.7870E^{-3}$	$2.1028E^{-3}$	$6.5772E^{-4}$	31.28%
120°	$2.7922E^{-3}$	$2.1034E^{-3}$	$6.5993E^{-4}$	31.37%
150°	$2.7930E^{-3}$	$2.1042E^{-3}$	$6.6027E^{-4}$	31.38%
180°	$2.7930E^{-3}$	$2.1041E^{-3}$	$6.6040E^{-4}$	31.39%

Table E.4: Clear ocean - unbiased sample variances obtained from algorithm Xiao 2 varying the receiver's FOV.

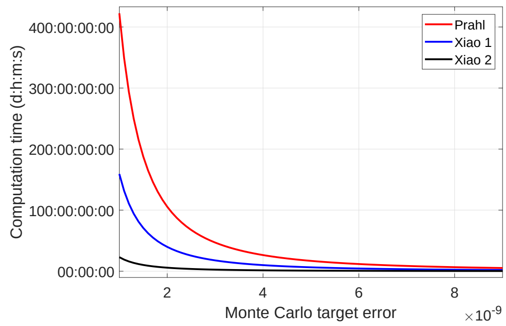
FOV	Algorithm Xiao 2	Ratio 1	Ratio 2
20°	$5.1030E^{-5}$	2.51%	8.67%
40°	$5.8764E^{-5}$	2.82%	9.19%
90°	$6.1356E^{-5}$	2.92%	9.33%
120°	$6.1449E^{-5}$	2.92%	9.31%
150°	$6.1538E^{-5}$	2.92%	9.32%
180°	$6.1639E^{-5}$	2.93%	9.33%



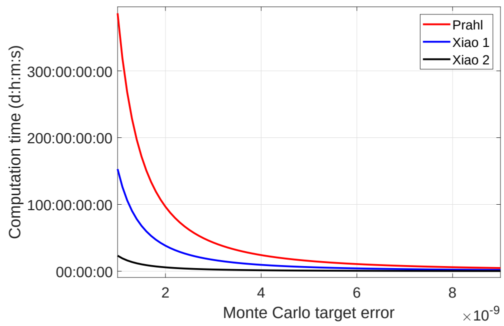
(a) FOV = 20°



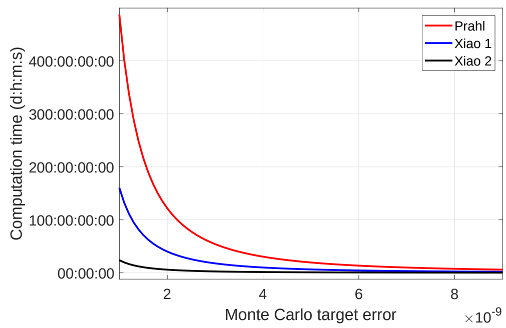
(b) FOV = 40°



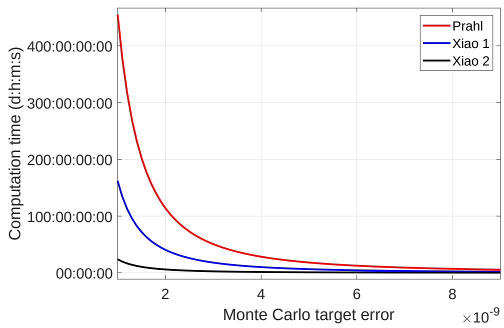
(c) FOV = 90°



(d) FOV = 120°



(e) FOV = 150°



(f) FOV = 180°

Figure E.2: Expected computation time varying the target error for Clear ocean with different receiver's FOV.

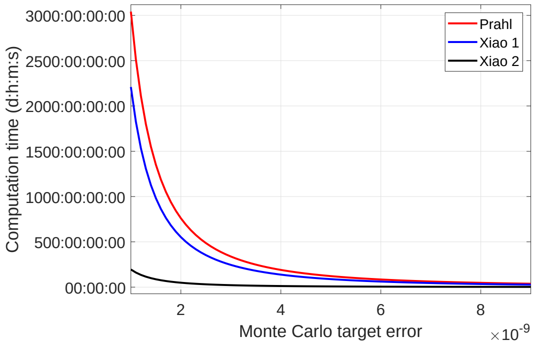
E.3: Changing the distance between T_x and R_x

Table E.5: Clear ocean - unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the distance between T_x and R_x.

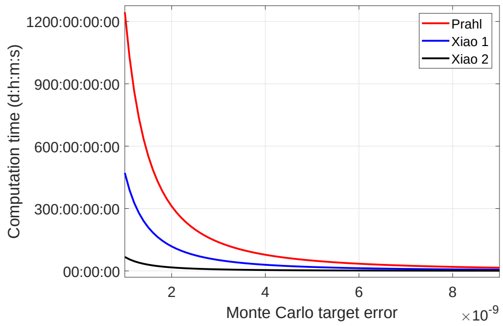
Distance (m)	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
5	$3.2056E^{-2}$	$2.7518E^{-2}$	$1.5717E^{-2}$	57.11%
8	$9.3231E^{-3}$	$7.6500E^{-3}$	$3.2345E^{-3}$	42.28%
12	$2.7930E^{-3}$	$2.1041E^{-3}$	$6.6040E^{-4}$	31.39%
16	$1.0586E^{-3}$	$7.2859E^{-4}$	$1.9072E^{-4}$	26.18%
20	$4.5653E^{-4}$	$2.8623E^{-4}$	$6.8356E^{-5}$	23.88%

Table E.6: Clear ocean - unbiased sample variances obtained from algorithm Xiao 2 varying the distance between T_x and R_x.

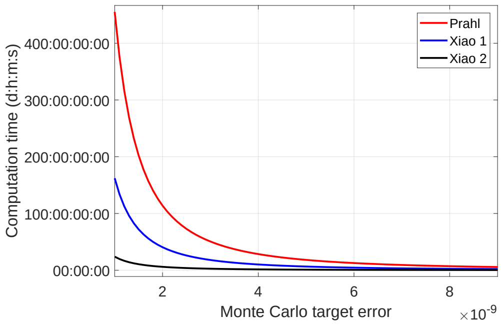
Distance (m)	Algorithm Xiao 2	Ratio 1	Ratio 2
5	$7.6130E^{-4}$	2.77%	4.84%
8	$2.1803E^{-4}$	2.85%	6.74%
12	$6.1639E^{-5}$	2.93%	9.33%
16	$2.1680E^{-5}$	2.98%	11.37%
20	$8.5447E^{-6}$	2.99%	12.50%



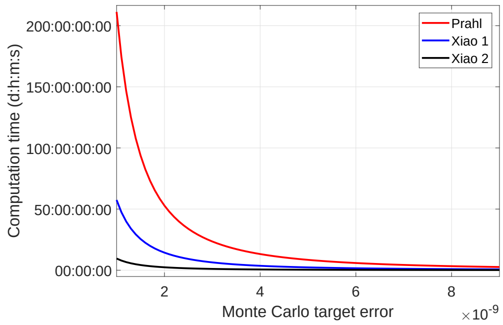
(a) Distance = 5m



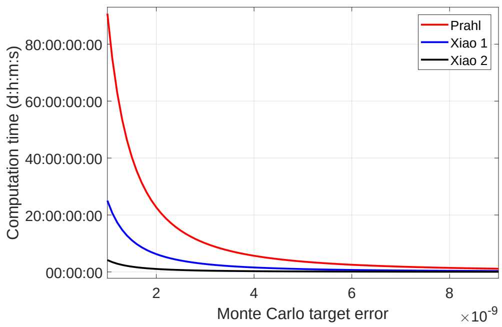
(b) Distance = 8m



(c) Distance = 12m



(d) Distance = 16m



(e) Distance = 20m

Figure E.3: Expected computation time varying the target error for Clear ocean with different distance between R_x and T_x.

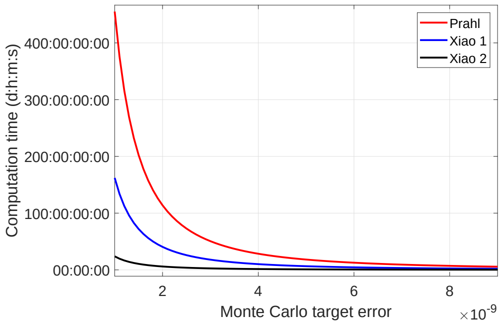
E.4: Changing the receiver's orientation

Table E.7: Clear ocean - unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the receiver's orientation.

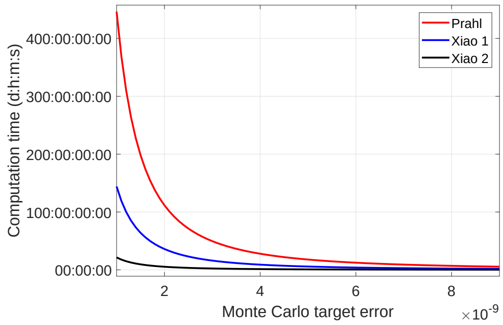
Orientation	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
0°	$2.7930E^{-3}$	$2.1041E^{-3}$	$6.6040E^{-4}$	31.39%
15°	$2.4801E^{-3}$	$1.8692E^{-3}$	$5.8298E^{-4}$	31.19%
30°	$2.4201E^{-3}$	$1.8231E^{-3}$	$5.6854E^{-4}$	31.19%
45°	$1.9762E^{-3}$	$1.4901E^{-3}$	$4.5985E^{-4}$	30.86%
60°	$1.3995E^{-3}$	$1.0559E^{-3}$	$3.2085E^{-4}$	30.39%
75°	$7.2909E^{-4}$	$5.4845E^{-4}$	$1.6342E^{-4}$	29.80%

Table E.8: Clear ocean - unbiased sample variances obtained from algorithm Xiao 2 varying the receiver's orientation.

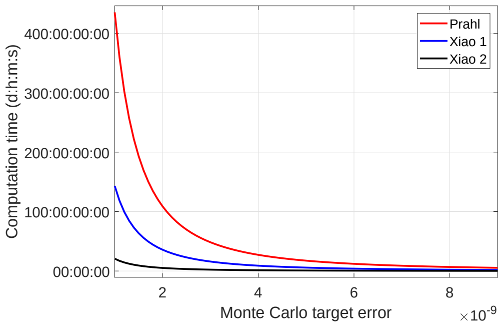
Orientation	Algorithm Xiao 2	Ratio 1	Ratio 2
0°	$6.1639E^{-5}$	2.93%	9.33%
15°	$5.5620E^{-5}$	2.98%	9.54%
30°	$5.4426E^{-5}$	2.99%	9.57%
45°	$4.5029E^{-5}$	3.02%	9.79%
60°	$3.2287E^{-5}$	3.06%	10.06%
75°	$1.7023E^{-5}$	3.10%	10.42%



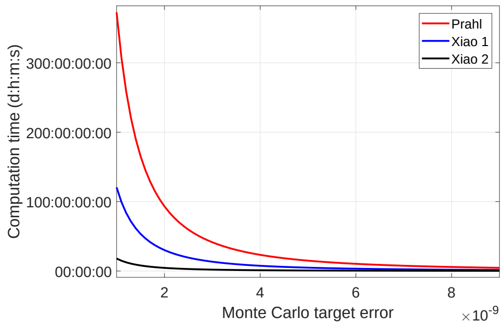
(a) Orientation = 0°



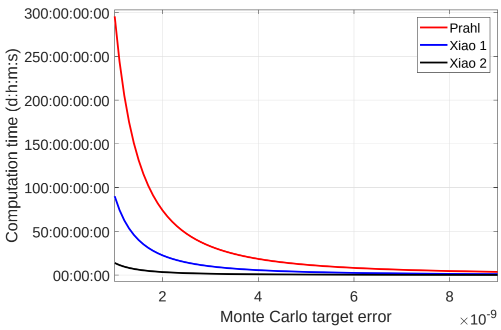
(b) Orientation = 15°



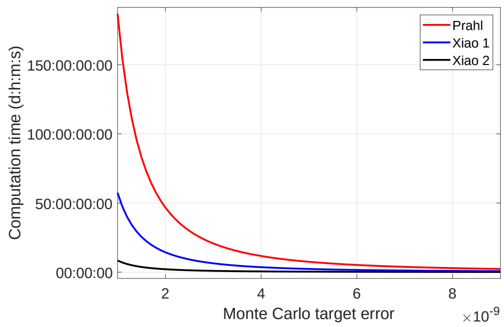
(c) Orientation = 30°



(d) Orientation = 45°



(e) Orientation = 60°



(f) Orientation = 75°

Figure E.4: Expected computation time varying the target error for Clear ocean with different receiver's orientation.

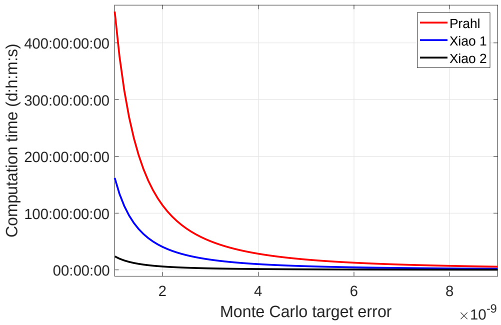
E.5: Changing the receiver's relation position to T_x

Table E.9: Clear ocean - unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the receiver's relative position.

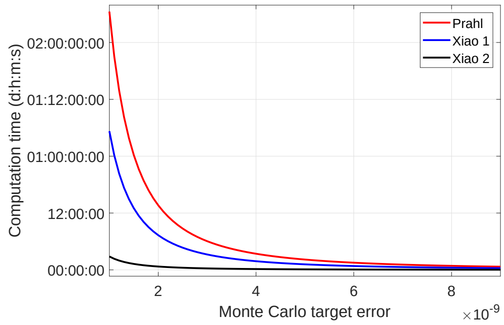
Rotation angle	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
$0°$	$2.7930E^{-3}$	$2.1041E^{-3}$	$6.6040E^{-4}$	31.39%
$15°$	$2.9836E^{-5}$	$1.0758E^{-5}$	$4.8740E^{-6}$	45.30%
$30°$	$3.7775E^{-6}$	$1.2496E^{-6}$	$5.2387E^{-7}$	41.92%
$45°$	$1.1010E^{-6}$	$3.5870E^{-7}$	$1.3686E^{-7}$	38.16%
$60°$	$4.4897E^{-7}$	$1.3905E^{-7}$	$4.9977E^{-8}$	35.94%
$75°$	$2.3146E^{-7}$	$7.0731E^{-8}$	$2.4892E^{-8}$	35.19%

Table E.10: Clear ocean - unbiased sample variances obtained from algorithm Xiao 2 varying the receiver's relative position for Clear ocean.

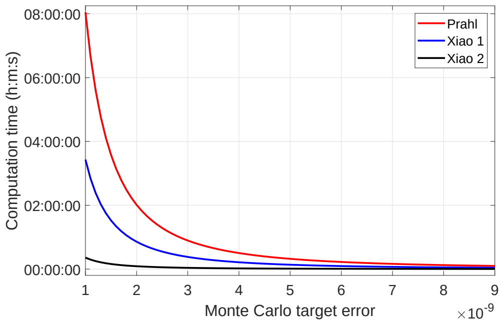
Rotation angle	Algorithm Xiao 2	Ratio 1	Ratio 2
$0°$	$6.1639E^{-5}$	2.93%	9.33%
$15°$	$3.2060E^{-7}$	2.98%	6.58%
$30°$	$3.8281E^{-8}$	3.06%	7.31%
$45°$	$1.0771E^{-8}$	3.00%	7.87%
$60°$	$4.3100E^{-9}$	3.10%	8.62%
$75°$	$1.9084E^{-9}$	2.70%	7.67%



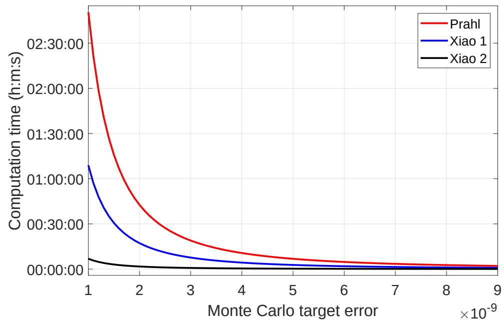
(a) Rotation angle = 0°



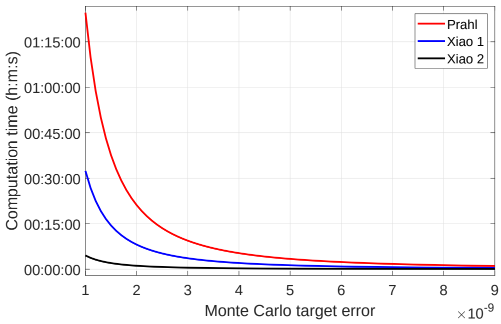
(b) Rotation angle = 15°



(c) Rotation angle = 30°



(d) Rotation angle = 45°



(e) Rotation angle = 60°

(f) Rotation angle = 75°

Figure E.5: Expected computation time varying the target error for Clear ocean with different receiver's relative position to T_x.

E.6: Changing the albedo of sea ground

Table E.11: Clear Ocean - Unbiased sample variance obtained from Prahl-R, Xiao 1-R and Xiao 2-R algorithms varying the albedo of sea ground.

Albedo	Channel's gain	Prahl-R	Xiao 1-R	Xiao 2-R
0.1	$2.9048E^{-5}$	$3.7178E^{-5}$	$1.2482E^{-5}$	$1.1250E^{-6}$
0.2	$2.9656E^{-5}$	$3.7670E^{-5}$	$1.2695E^{-5}$	$1.1111E^{-6}$
0.3	$3.0294E^{-5}$	$3.7740E^{-5}$	$1.3165E^{-5}$	$1.1373E^{-6}$
0.4	$3.0948E^{-5}$	$3.8356E^{-5}$	$1.3509E^{-5}$	$1.1531E^{-6}$
0.5	$3.1574E^{-5}$	$3.9164E^{-5}$	$1.3924E^{-5}$	$1.1725E^{-6}$
0.6	$3.2202E^{-5}$	$4.0157E^{-5}$	$1.4464E^{-5}$	$1.1886E^{-6}$
0.7	$3.2885E^{-5}$	$4.0953E^{-5}$	$1.5438E^{-5}$	$1.2367E^{-6}$
0.8	$3.3526E^{-5}$	$4.1902E^{-5}$	$1.6131E^{-5}$	$1.2349E^{-6}$
0.9	$3.4211E^{-5}$	$4.3204E^{-5}$	$1.7308E^{-5}$	$1.2716E^{-6}$

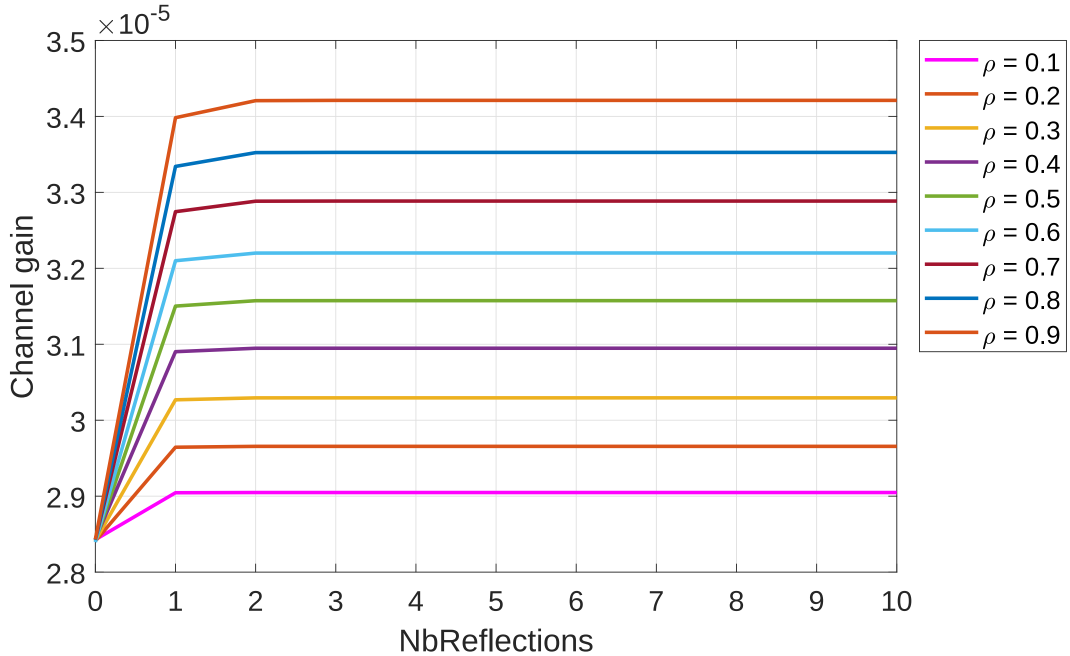


Figure E.6: Channel gain varying $NbReflections$ with different albedo of sea ground for Clear Ocean

E.7: Changing the distance between sea ground and sensors

Table E.12: Clear Ocean - Unbiased sample variance obtained from Prahl-R, Xiao 1-R and Xiao 2-R algorithms varying the distance between sensors and the sea ground.

Distance (m)	Channel's gain	Prahl-R	Xiao 1-R	Xiao 2-R
0.1	$2.3864E^{-5}$	$3.3389E^{-5}$	$1.0461E^{-5}$	$1.0914E^{-6}$
0.5	$2.8576E^{-5}$	$3.6904E^{-5}$	$1.2489E^{-5}$	$1.1204E^{-6}$
1	$3.0294E^{-5}$	$3.7740E^{-5}$	$1.3165E^{-5}$	$1.1373E^{-6}$
2	$3.1735E^{-5}$	$3.8402E^{-5}$	$1.3311E^{-5}$	$1.1456E^{-6}$
5	$3.1081E^{-5}$	$3.8431E^{-5}$	$1.3064E^{-5}$	$1.1466E^{-6}$
10	$2.9165E^{-5}$	$3.7880E^{-5}$	$1.2760E^{-5}$	$1.1425E^{-6}$
15	$2.9147E^{-5}$	$3.7611E^{-5}$	$1.2814E^{-5}$	$1.1192E^{-6}$
20	$2.9155E^{-5}$	$3.7615E^{-5}$	$1.2771E^{-5}$	$1.1217E^{-6}$
50	$2.9160E^{-5}$	$3.7694E^{-5}$	$1.2800E^{-5}$	$1.1540E^{-6}$

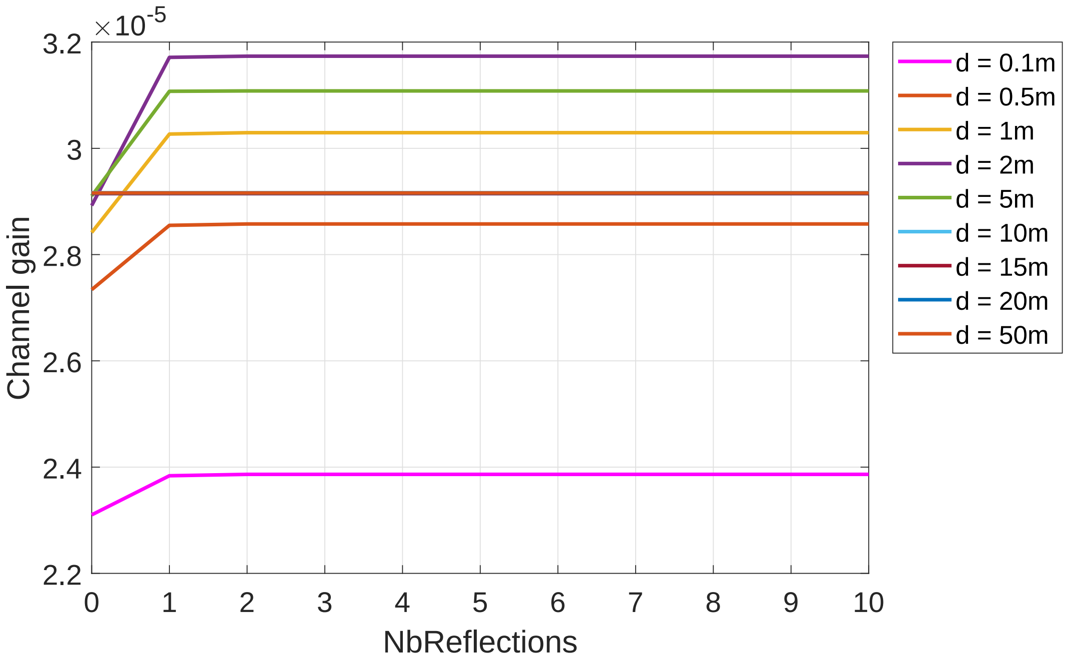


Figure E.7: Channel gain varying $NbReflections$ with different distance between sea ground and sensors for Clear Ocean

Appendix F: Simulation results for Coastal water

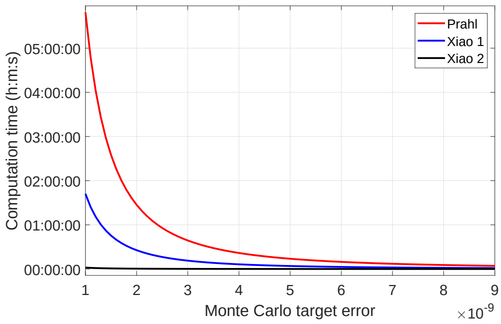
F.1: Changing receiver's area

Table F.1: Coastal - unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the receiver's radius (in cm).

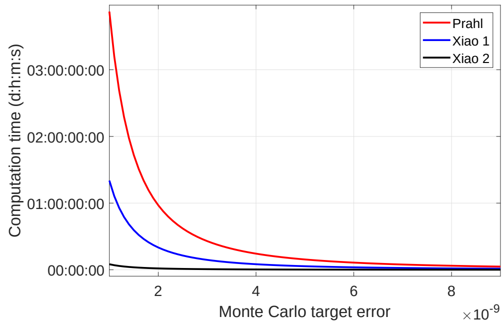
Radius (cm)	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
0.5	$9.4209E^{-7}$	$5.6571E^{-7}$	$1.7368E^{-7}$	30.70%
2	$1.5358E^{-5}$	$8.9878E^{-6}$	$2.9601E^{-6}$	32.93%
5	$9.5881E^{-5}$	$5.5785E^{-5}$	$1.9851E^{-5}$	35.58%
8	$2.4554E^{-4}$	$1.4324E^{-4}$	$5.3508E^{-5}$	37.35%
20	$1.5327E^{-3}$	$8.9104E^{-4}$	$3.8410E^{-4}$	43.11%
50	$9.5127E^{-3}$	$5.4647E^{-3}$	$2.7934E^{-3}$	51.12%

Table F.2: Coastal - unbiased sample variances obtained from algorithm Xiao 2 varying the receiver's radius (in cm).

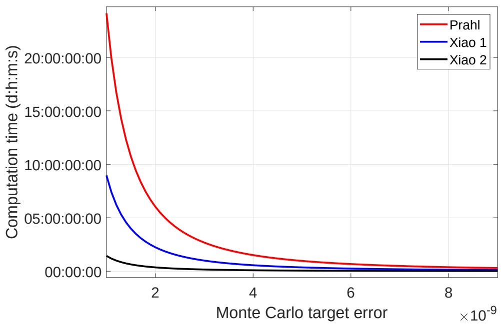
Radius (cm)	Algorithm Xiao 2	Ratio 1	Ratio 2
0.5	$2.3345E^{-9}$	0.41%	1.34%
2	$1.4706E^{-7}$	1.64%	4.97%
5	$2.2208E^{-6}$	3.98%	11.19%
8	$8.9007E^{-6}$	6.21%	16.63%
20	$1.2772E^{-4}$	14.33%	33.25%
50	$1.6137E^{-3}$	29.53%	57.77%



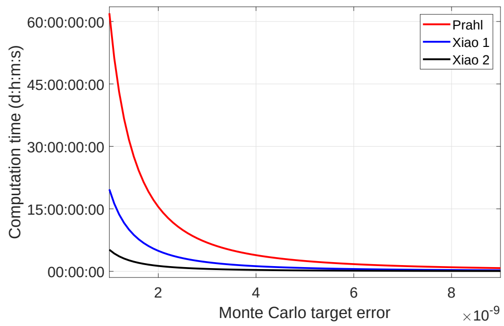
(a) Diameter = 0.5cm



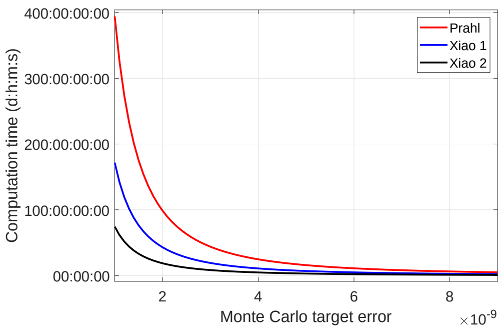
(b) Diameter = 2cm



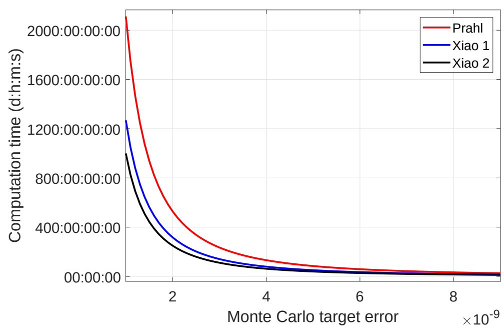
(c) Diameter = 5cm



(d) Diameter = 8cm



(e) Diameter = 20cm



(f) Diameter = 50cm

Figure F.1: Expected computation time varying the target error for Coastal with different receiver's area.

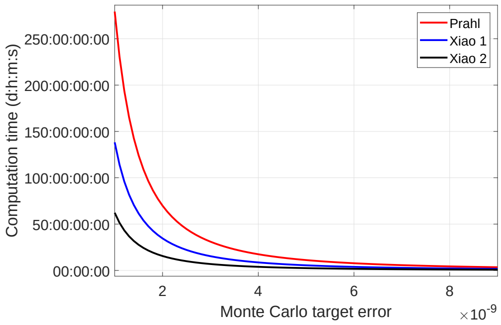
F.2: Changing receiver's FOW

Table F.3: Coastal - unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the receiver's FOV.

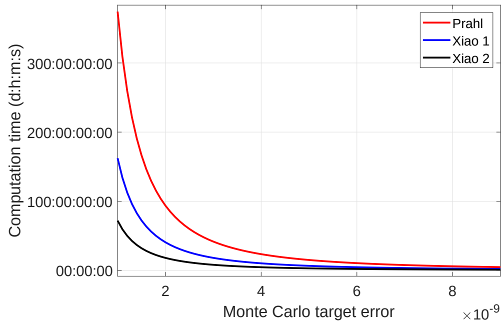
FOV	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
20°	$1.3478E^{-3}$	$8.1582E^{-4}$	$3.0479E^{-4}$	37.36%
40°	$1.4807E^{-3}$	$8.7079E^{-4}$	$3.6152E^{-4}$	41.52%
90°	$1.5276E^{-3}$	$8.8889E^{-4}$	$3.8208E^{-4}$	42.98%
120°	$1.5314E^{-3}$	$8.9034E^{-4}$	$3.8353E^{-4}$	43.08%
150°	$1.5328E^{-3}$	$8.9155E^{-4}$	$3.8424E^{-4}$	43.10%
180°	$1.5327E^{-3}$	$8.9104E^{-4}$	$3.8410E^{-4}$	43.11%

Table F.4: Coastal - unbiased sample variances obtained from algorithm Xiao 2 varying the receiver's FOV.

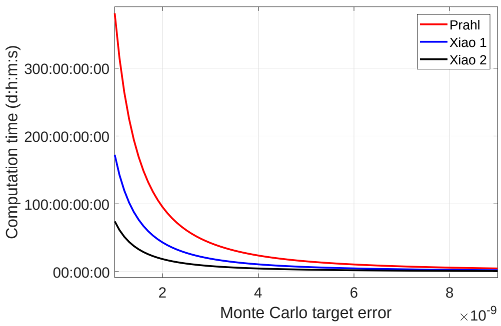
FOV	Algorithm Xiao 2	Ratio 1	Ratio 2
20°	$1.0739E^{-4}$	13.16%	35.24%
40°	$1.2277E^{-4}$	14.10%	33.96%
90°	$1.2729E^{-4}$	14.32%	33.32%
120°	$1.2762E^{-4}$	14.33%	33.28%
150°	$1.2774E^{-4}$	14.33%	33.24%
180°	$1.2772E^{-4}$	14.33%	33.25%



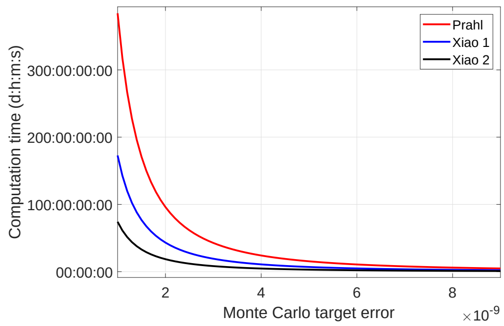
(a) FOV = 20°



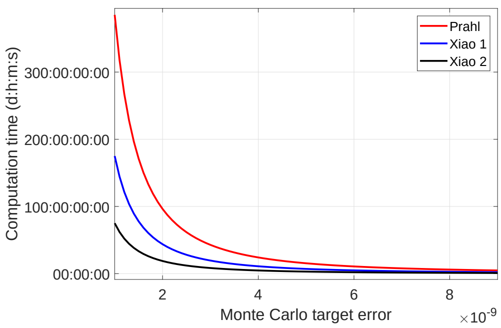
(b) FOV = 40°



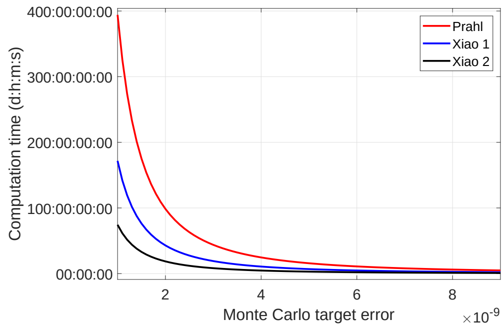
(c) FOV = 90°



(d) FOV = 120°



(e) FOV = 150°



(f) FOV = 180°

Figure F.2: Expected computation time varying the target error for Coastal with different receiver's FOV.

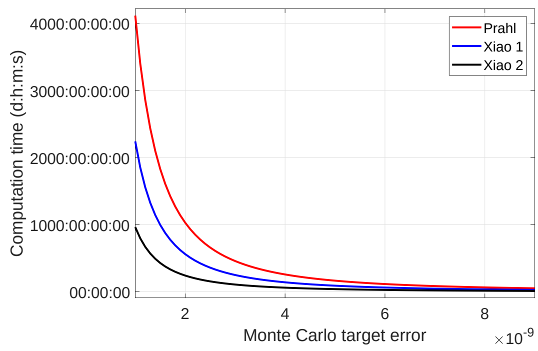
F.3: Changing the distance between T_x and R_x

Table F.5: Coastal - unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the distance between T_x and R_x.

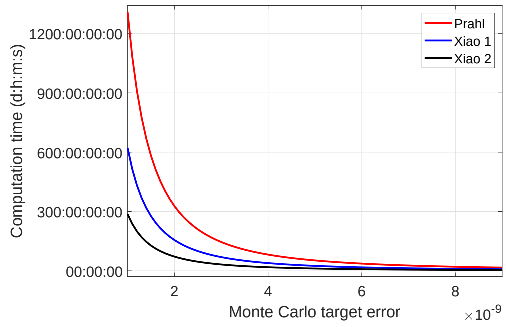
Distance (m)	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
5	$2.4993E^{-2}$	$1.9345E^{-2}$	$1.0265E^{-2}$	53.06%
8	$6.2563E^{-3}$	$4.3293E^{-3}$	$1.9526E^{-3}$	45.10%
12	$1.5327E^{-3}$	$8.9104E^{-4}$	$3.8410E^{-4}$	43.11%
16	$4.7617E^{-4}$	$2.3091E^{-4}$	$9.9770E^{-4}$	43.21%
20	$1.6906E^{-4}$	$6.8171E^{-5}$	$2.9654E^{-5}$	43.50%

Table F.6: Coastal - unbiased sample variances obtained from algorithm Xiao 2 varying the distance between T_x and R_x.

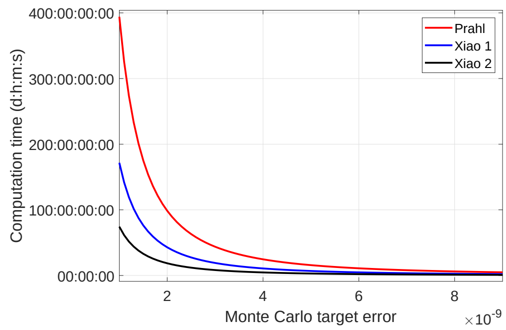
Distance (m)	Algorithm Xiao 2	Ratio 1	Ratio 2
5	$2.6101E^{-3}$	13.49%	25.43%
8	$6.0811E^{-4}$	14.05%	31.14%
12	$1.2772E^{-4}$	14.33%	33.25%
16	$3.3267E^{-5}$	14.41%	33.34%
20	$9.7985E^{-6}$	14.37%	33.04%



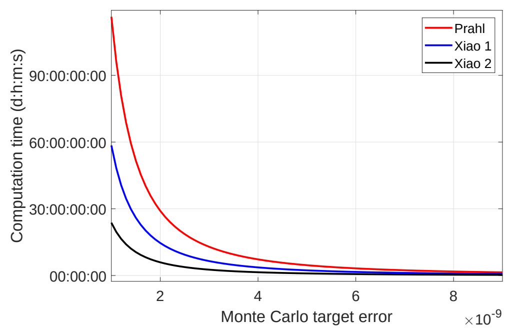
(a) Distance = 5m



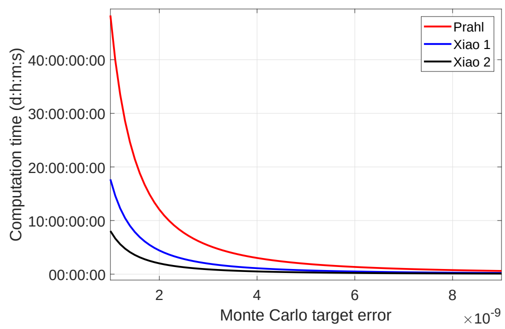
(b) Distance = 8m



(c) Distance = 12m



(d) Distance = 16m



(e) Distance = 20m

Figure F.3: Expected computation time varying the target error for Coastal with different distance between R_x and T_x.

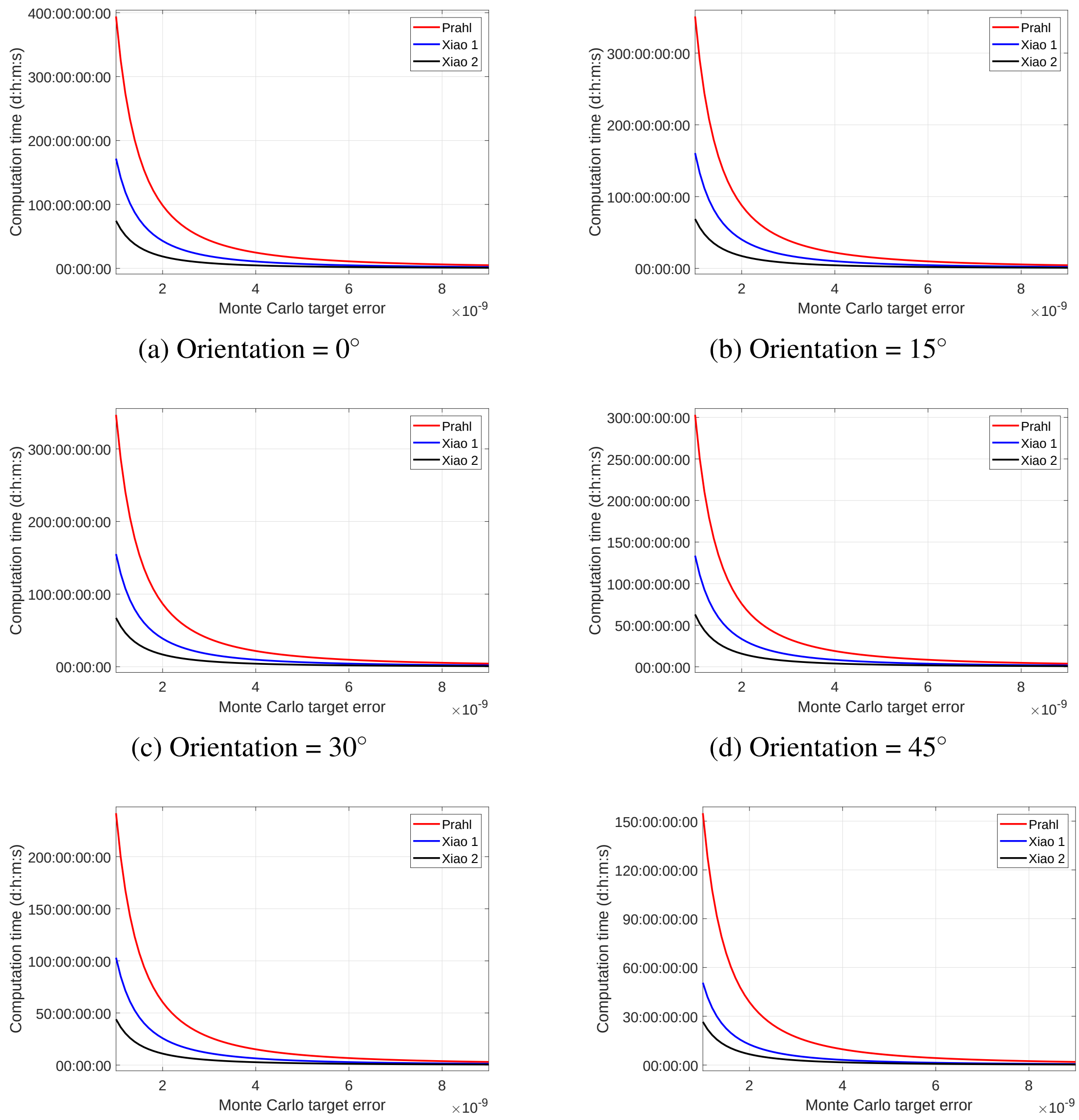
F.4: Changing the receiver's orientation

Table F.7: Coastal - unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the receiver's orientation.

Orientation	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
$0°$	$1.5327E^{-3}$	$8.9104E^{-4}$	$3.8410E^{-4}$	43.11%
$15°$	$1.3628E^{-3}$	$7.9181E^{-4}$	$3.3836E^{-4}$	42.73%
$30°$	$1.3282E^{-3}$	$7.7296E^{-4}$	$3.2862E^{-4}$	42.51%
$45°$	$1.0858E^{-3}$	$6.3184E^{-4}$	$2.6393E^{-4}$	41.77%
$60°$	$7.6973E^{-4}$	$4.4747E^{-4}$	$1.8131E^{-4}$	40.52%
$75°$	$4.0380E^{-4}$	$2.3376E^{-4}$	$9.0614E^{-5}$	38.76%

Table F.8: Coastal - unbiased sample variances obtained from algorithm Xiao 2 varying the receiver's orientation.

Orientation	Algorithm Xiao 2	Ratio 1	Ratio 2
$0°$	$1.2772E^{-4}$	14.33%	33.25%
$15°$	$1.1447E^{-4}$	14.46%	33.83%
$30°$	$1.1172E^{-4}$	14.45%	34.00%
$45°$	$9.1373E^{-5}$	14.46%	34.62%
$60°$	$6.4187E^{-5}$	14.34%	35.40%
$75°$	$3.2923E^{-5}$	14.08%	36.33%



(a) Orientation = 0°

(b) Orientation = 15°

(c) Orientation = 30°

(d) Orientation = 45°

(e) Orientation = 60°

(f) Orientation = 75°

Figure F.4: Expected computation time varying the target error for Coastal with different receiver's orientations.

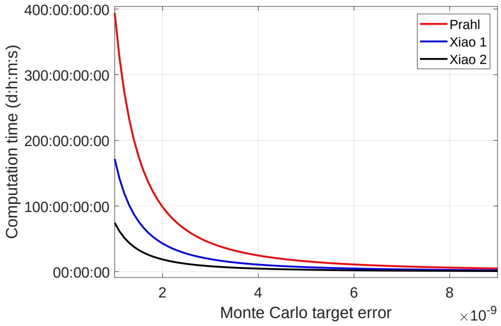
F.5: Changing the receiver's relation position to T_x

Table F.9: Coastal - unbiased sample variances obtained from Prahl's Algorithm [1] and Algorithm Xiao 1 varying the receiver's relative position.

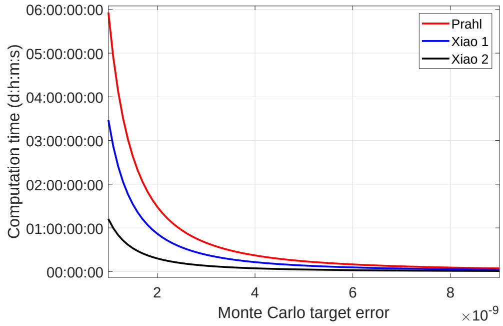
Rotation angle	Channel's Gain	Prahl's Algorithm [1]	Algorithm Xiao 1	Ratio
$0°$	$1.5327E^{-3}$	$8.9104E^{-4}$	$3.8410E^{-4}$	43.11%
$15°$	$4.1586E^{-5}$	$1.4745E^{-5}$	$7.3430E^{-6}$	49.80%
$30°$	$5.4253E^{-6}$	$1.7275E^{-6}$	$8.3354E^{-7}$	48.25%
$45°$	$1.5652E^{-6}$	$4.7319E^{-7}$	$2.2364E^{-7}$	47.26%
$60°$	$6.4351E^{-7}$	$1.8929E^{-7}$	$8.5467E^{-8}$	45.15%
$75°$	$3.2560E^{-7}$	$9.1878E^{-8}$	$4.3179E^{-8}$	47.00%

Table F.10: Coastal - unbiased sample variances obtained from algorithm Xiao 2 varying the receiver's relative position for Coastal.

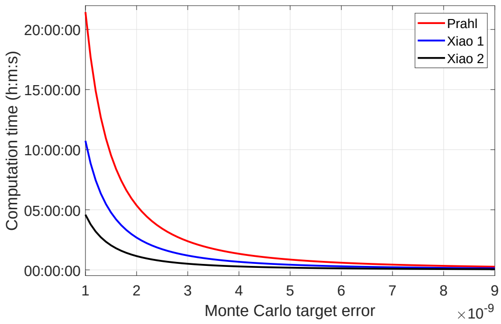
Rotation angle	Algorithm Xiao 2	Ratio 1	Ratio 2
$0°$	$1.2772E^{-4}$	14.33%	33.25%
$15°$	$2.1025E^{-6}$	14.26%	28.63%
$30°$	$2.5335E^{-7}$	14.67%	30.39%
$45°$	$6.8694E^{-8}$	14.52%	30.72%
$60°$	$2.7735E^{-8}$	14.65%	32.45%
$75°$	$1.2371E^{-8}$	13.47%	28.65%



(a) Rotation angle = 0°

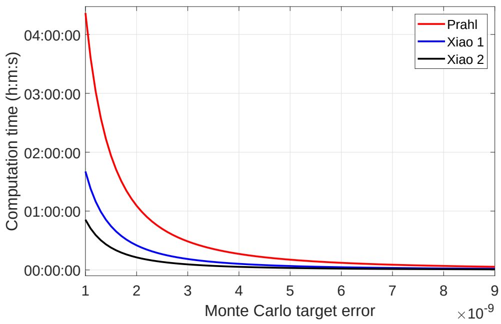


(b) Rotation angle = 15°

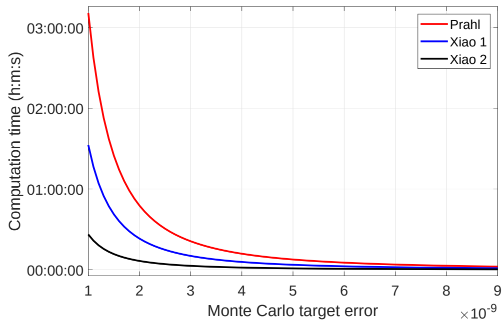


(c) Rotation angle = 30°

(d) Rotation angle = 45°



(e) Rotation angle = 60°



(f) Rotation angle = 75°

Figure F.5: Expected computation time varying the target error for Coastal with different receiver's relative positions to T_x.

F.6: Changing the albedo of sea ground

Table F.11: Coastal - Unbiased sample variance obtained from Prahl-R, Xiao 1-R and Xiao 2-R algorithms varying the albedo of sea ground.

Albedo	Channel's gain	Prahl-R	Xiao 1-R	Xiao 2-R
0.1	$2.1934E^{-5}$	$2.0532E^{-5}$	$9.2021E^{-6}$	$2.9734E^{-6}$
0.2	$2.2476E^{-5}$	$2.0580E^{-5}$	$9.6661E^{-6}$	$2.9617E^{-6}$
0.3	$2.3147E^{-5}$	$2.1186E^{-5}$	$9.7336E^{-6}$	$3.0439E^{-6}$
0.4	$2.3672E^{-5}$	$2.1292E^{-5}$	$1.0015E^{-5}$	$3.0385E^{-6}$
0.5	$2.4316E^{-5}$	$2.2093E^{-5}$	$1.0622E^{-5}$	$3.1313E^{-6}$
0.6	$2.5054E^{-5}$	$2.3149E^{-5}$	$1.1185E^{-5}$	$3.2526E^{-6}$
0.7	$2.5697E^{-5}$	$2.3882E^{-5}$	$1.1907E^{-5}$	$3.3881E^{-6}$
0.8	$2.6333E^{-5}$	$2.5153E^{-5}$	$1.2915E^{-5}$	$3.5017E^{-6}$
0.9	$2.7000E^{-5}$	$2.6460E^{-5}$	$1.3990E^{-5}$	$3.6752E^{-6}$

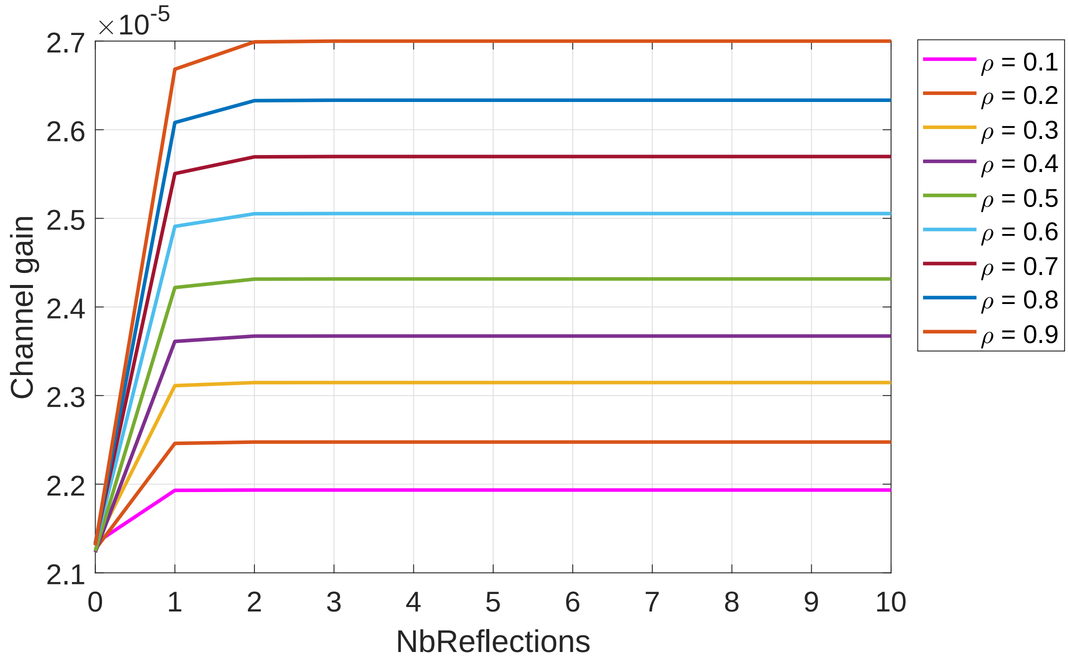


Figure F.6: Channel gain varying $NbReflections$ with different albedos of sea ground for Coastal water

F.7: Changing the distance between sea ground and sensors

Table F.12: Coastal - Unbiased sample variance obtained from Prahl-R, Xiao 1-R and Xiao 2-R algorithms varying the distance between sensors and the sea ground.

Distance (m)	Channel's gain	Prahl-R	Xiao 1-R	Xiao 2-R
0.1	$1.5252E^{-5}$	$1.5502E^{-5}$	$6.6132E^{-6}$	$2.3143E^{-6}$
0.5	$2.1034E^{-5}$	$1.9805E^{-5}$	$9.1600E^{-6}$	$2.7912E^{-6}$
1	$2.3147E^{-5}$	$2.1186E^{-5}$	$9.7336E^{-6}$	$3.0439E^{-6}$
2	$2.4484E^{-5}$	$2.1723E^{-5}$	$1.0136E^{-5}$	$3.0382E^{-6}$
5	$2.3913E^{-5}$	$2.1549E^{-5}$	$9.8048E^{-6}$	$3.0601E^{-6}$
10	$2.2239E^{-5}$	$2.0934E^{-5}$	$9.4608E^{-6}$	$2.9539E^{-6}$
15	$2.2341E^{-5}$	$2.1070E^{-5}$	$9.5710E^{-6}$	$3.0588E^{-6}$
20	$2.2282E^{-5}$	$2.0875E^{-5}$	$9.6227E^{-6}$	$2.9753E^{-6}$
50	$2.2261E^{-5}$	$2.0925E^{-5}$	$9.6562E^{-6}$	$2.9949E^{-6}$

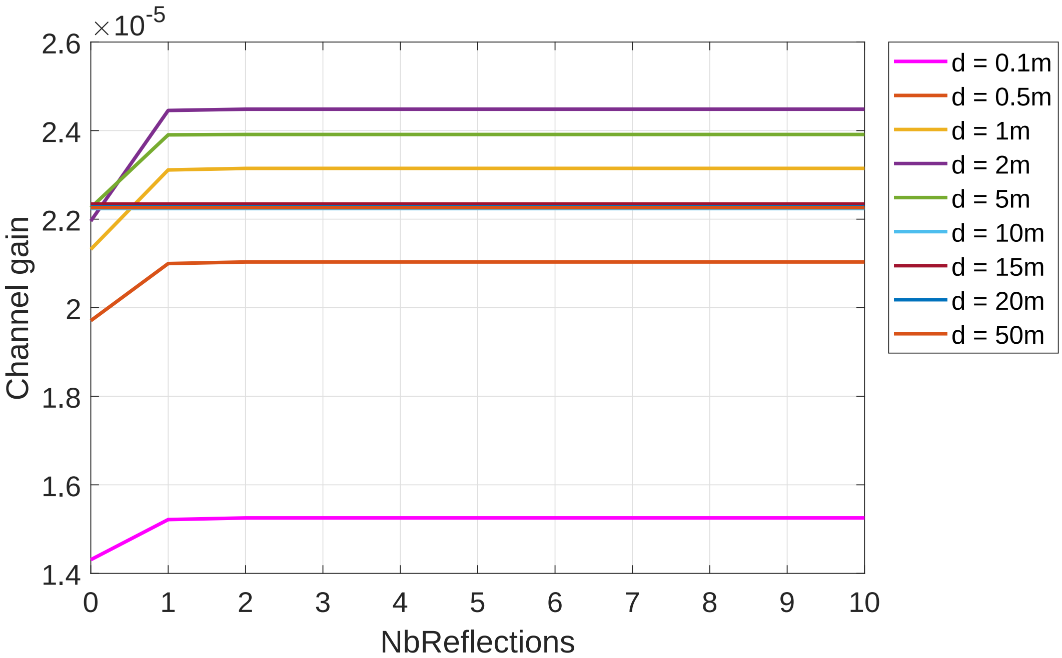


Figure F.7: Channel gain varying $NbReflections$ with different distances between sea ground and sensors for Coastal water