Rendu réaliste de nuages en temps réel

Antoine Bouthors

Real-time realistic rendering of clouds

Rendu réaliste de nuages en temps-réel

Antoine Bouthors

Thèse

présentée pour l'obtention du titre de
Docteur de l'Université Joseph Fourier
spécialité Informatique

préparée au sein du projet **EVASION**, laboratoire Jean Kuntzmann
UMR 5527 (Grenoble Universités, CNRS, INRIA)

dans le cadre de l'école doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique

soutenue le 17 juin 2008.

Composition du jury:

Eric Blayo	Professeur	Université de Grenoble 1	Président
Jerry Tessendorf	Principal Graphics Scientist	Rhythm & Hues	Rapporteur
Mathias Paulin	Professeur	Université Paul Sabatier	Rapporteur
Mark Harris	Senior Software Engineer	NVIDIA	Examinateur
Bernard Péroche	Professeur	Université de Lyon 1	Examinateur
Jérôme Riedi	Maitre de Conférences	Université de Lille 1	Examinateur
Fabrice Neyret	Directeur de Recherches	CNRS	Directeur de thèse

Acknowledgements

I would first like to thank Marie-Paule Cani for welcoming me in her team and Fabrice Neyret for giving me this great opportunity. I don't think I could have found a better advisor. I am grateful for his availability and the freedom he gave me, and sorry for the heart attacks I may have caused him during the deadlines. I also had the chance to visit Nelson Max for six months at UC Davis and LLNL and would like to thank him for confronting my ideas with his knowledge, experience and keenness.

Secondly, a warm thank you to all the members of my jury who traveled a (more or less) long way to listen to my rantings: in order of distance, Jerry Tessendorf (Los Angeles), Mark Harris (London), Jérôme Riedi (Lille), Mathias Paulin (Toulouse), Bernard Péroche (Lyon) and Eric Blayo (Grenoble). Special thanks to Jerry for staying around and explaining me all the radiative transfer physics I ought to have known, and to Jérôme to show me my intuitions were maybe not that stupid from a physicist's point of view.

This work would not have been possible without me exploiting the strengths of my colleagues and of a few gullible interns. So thanks to Sylvain Lefebvre, Eric Bruneton, Cyril Crassin, Benoit Laurent, Damien Rumiano and Guillaume Piolat. Eric Ragusi and Frédéric Saint-Marcel where there to set up and maintain the architecture used for the light transport study, and Laurence Boissieux spent her weekends modeling the cloud shapes. Jamie Wither, Nelson Max and Isabel Faria contributed to this document by proofreading it. Thumbs up to all of them.

I am going to miss all the people at Evasion, Artis and IDAV and all the other people at INRIA and UC Davis I have been hanging out with. It was a great pleasure to work in such friendly, diverse and inspiring places. The space is missing to cite everyone, but I have to give a special thought to Matt, who has been a great officemate, colleague, coauthor, friend, climbing partner and life-savior. I hope our music was not too traumatizing for our officemates and neighbors. Many thanks also to Anne Pierson, who was a very helpful and forgiving team assistant.

A great thanks to my friends for being a haven of (in)sanity outside research: the drama team, the climbing trips in the Sierra with Niels and Charlie or in the Alps with Matt, the nights out at the G-Street pub or the Family's pub, la coloc de Fontaine, the soccer tournaments, the INRIA babyfoot team, the folk music sessions, the ESIEE alumni.

Thanks to my parents for their inconditional support through all these years. Thanks to my brother for introducing me to the joys of coding.

Last but not least, my thoughts go to the women who attempted to share my life through these years. Their moral support helped me greatly in going through this experience. I keep a special place for each of them in my heart.

Copyright notice

Abstract

Video games and visual effects are always in the need for more realistic images of natural scenes, including clouds. Traditional computer graphics methods involve high amounts of memory and computing resources which currently limits their realism and speed. In this thesis we propose new approaches for the realistic rendering of clouds in realtime. These goals are achieved using a methodology relying on both physical and phenomenological approaches.

In the first part of this thesis we study the clouds from the point of view of a human observer, which allows us to identify the important visual features that characterize clouds. These features are the ones we seek to reproduce in order to achieve realism. In the second part we conduct an experimental study of light transport in a slab of cloud using the laws of radiative transfer. This study allows us to better understand the physics of light transport in clouds at a mesoscopic and macroscopic scale, to find the origin of the visual features of clouds, and to derive new, efficient phenomenological models of light transport. In the third part we propose two new methods for the rendering of clouds based on the findings of our study. The first model is aimed at stratiform clouds and achieves realistic rendering in real-time. The second model is aimed at cumuliform clouds and achieves realistic rendering in interactive time. These models are optimized for graphics hardware.

These works can be extended to other applications involving the rendering of participating media such as subsurface scattering.

Les jeux vidéo et les effets spéciaux ont un besoin constant d'images plus réalistes de scènes naturelles, ce qui inclut les nuages. Les méthodes traditionnelles de synthèse d'images exigent d'importantes ressources en mémoire et en calcul, ce qui limite leur réalisme et leur vitesse. Dans cette thèse, nous proposons de nouvelles approches pour le rendu de nuages à la fois réalistes et rapides. Ces objectifs sont atteints via une méthodologie s'appuyant sur des approches à la fois physiques et phénoménologiques.

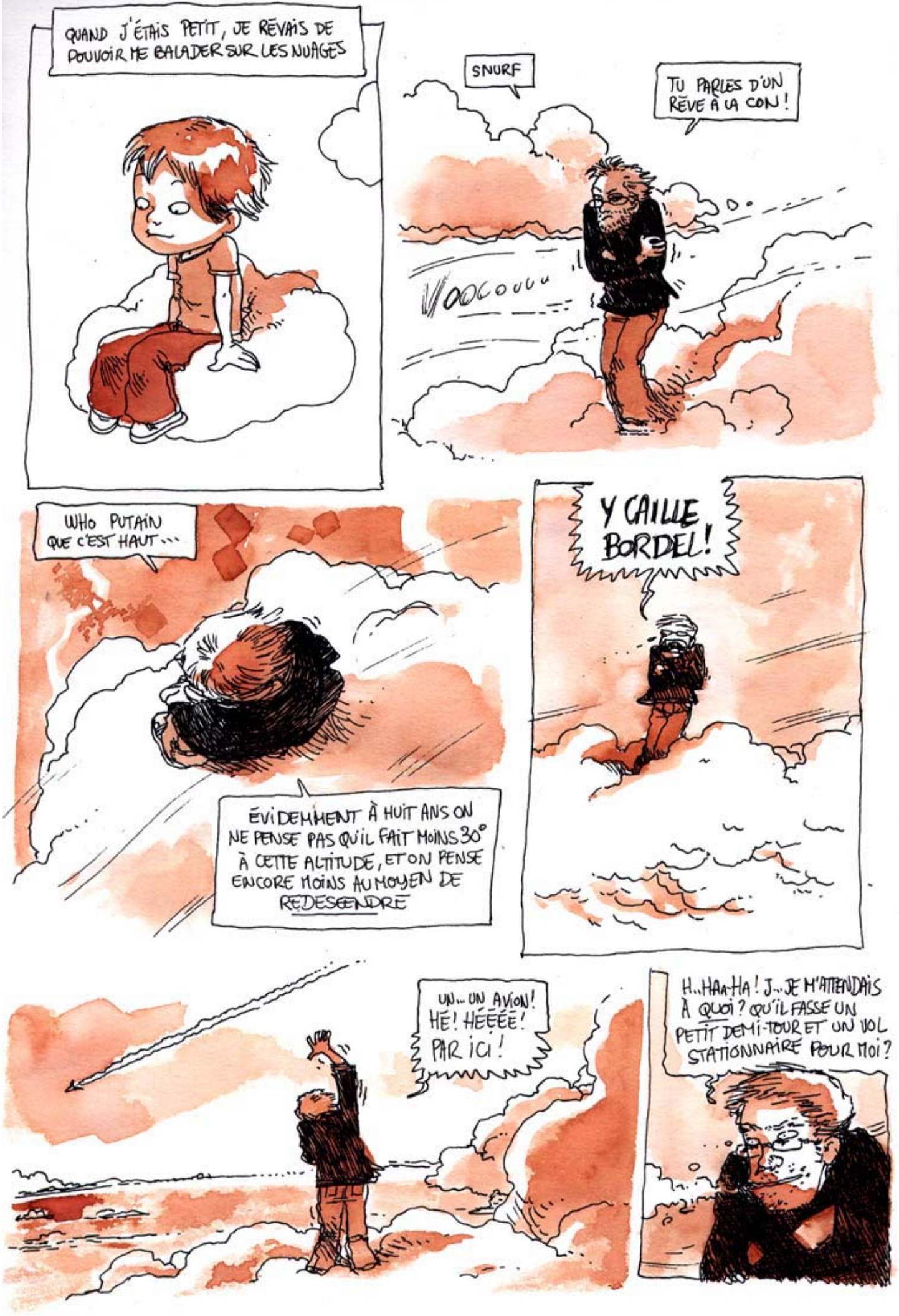
Dans la première partie de cette thèse, nous étudions les nuages du point de vue d'un observateur humain, ce qui nous permet d'identifier les éléments visuels importants caractérisant les nuages. Ces éléments visuels sont ceux que nous cherchons à reproduire dans un but de réalisme. Dans la deuxième partie nous conduisons une étude expérimentale du transport de la lumière dans une dalle de nuage en utilisant les lois du transfert radiatif. Cette étude nous permet de mieux comprendre le transport de la lumière dans les nuages à des échelles mésoscopiques et macroscopiques, de trouver l'origine de ces éléments visuels, et de déduire de nouveaux modèles de transport de la lumière phénoménologiques efficaces. Dans la troisième partie nous proposons deux nouvelles méthodes permettant le rendu de nuages réaliste en temps-réel basées sur les résultats de notre étude. Ces méthodes sont optimisées pour le matériel graphique.

Ces travaux peuvent être étendus à d'autres applications impliquant le rendu de milieux participants telles que la diffusion sous-surfacique.

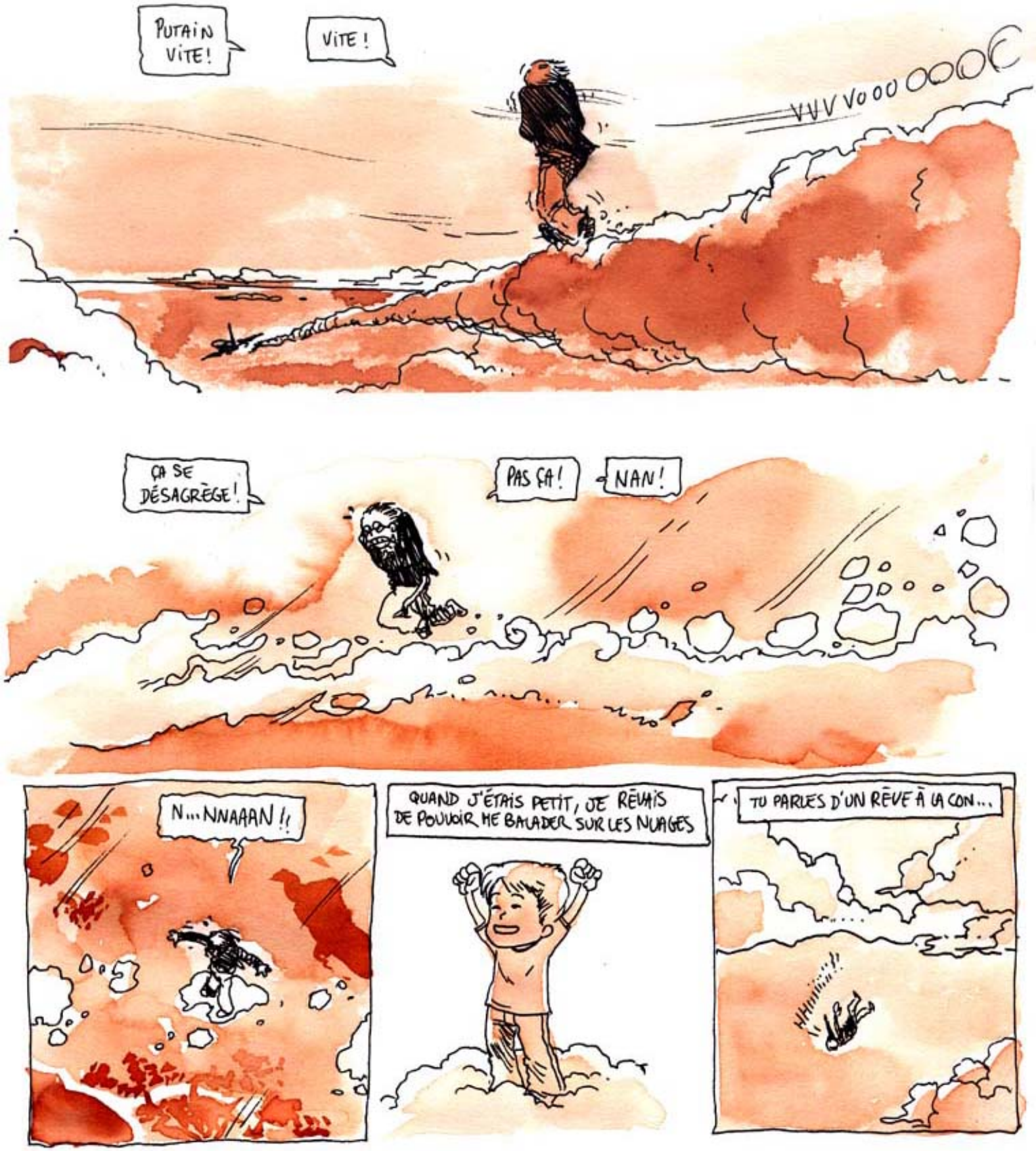
Contents

Part I

Scientific background


QUAND J'ÉTAIS PETIT, JE RÊVAIS DE POUVOIR ME BALADER SUR LES NUAGES
SNURF
TU PARLES D'UN RÊVE À LA CON !
WOOOUUU
WHO PUTAIN QUE C'EST HAUT...
ÉVIDEMMENT À HUIT ANS ON NE PENSE PAS QU'IL FAIT MOINS 30° À CETTE ALTITUDE, ET ON PENSE ENCORE MOINS AU MOYEN DE REDESCENDRE
Y CAILLE BORDEL !
UN... UN AVION ! HÉ ! HÉÉÉÉ ! PAR ICI !
H..HAA-HA ! J... JE M'ATTENDAIS À QUOI ? QU'IL FASSE UN PETIT DEMI-TOUR ET UN VOL STATIONNAIRE POUR MOI ?


VRROOOOSSSHHH
HAAAA
RÉSUMONS-NOUS... J'AI 5 KM DE VIDE SOUS LES PIEDS ET L'ESPÉRANCE DE VIE D'UN CANARI DANS L'ESPACE...
PLUS QU'UNE CHOSE À TENTER!
J'ESPÈRE QUE CET AVION ÉTAIT EN TRAIN DE SE POSER...


PUTAIN VITE!
VITE !
VVV Voo ooooC
ÇA SE DÉSAGRÈGE!
PAS ÇA!
NAN!
N...NNAAAN !!
QUAND J'ÉTAIS PETIT, JE RÊVAIS DE POUVOIR ME BALADER SUR LES NUAGES
TU PARLES D'UN RÊVE À LA CON...
HHHHHH!!

Chapter 1

Introduction

Contents

1.1 Introduction

1.1.1 Motivation

Video games, flight simulators and special effects have always the need for more realistic images of natural scenes. Among the objects they seek to reproduce, clouds are one of the most complex. Traditionally these involve expensive computations in order to be reproduced faithfully, which currently limits their realism. Moreover, classical methods involve solving chaotic physical simulations whose outcome are difficult to predict. This is a problem for computer graphics (CG) artists, who need intuitive control of these objects. As a result, new approaches for the simulation of clouds have to be found. These approaches should be realistic, fast, and controllable.

CG

Moreover, studying clouds from the human eye's point of view may help physicists and meteorologists understand this complex phenomenon.

Figure 1.1: *Photograph of a* cumulus congestus *cloud. This is the kind of image we seek to generate.*

1.1.2 Objectives

Three main problems can be defined relating to the simulation of clouds: simulating their shape (modelling), their dynamics (animating), and their aspect (rendering). Each of these problems is an especially difficult task on its own. This thesis will mainly focus on the rendering of clouds.

Previous approaches have shown that it is possible to achieve quite realistic offline rendering of clouds while realtime rendering of clouds is still somewhat unrealistic. In particular, these techniques fail at reproducing cumulus clouds, which are the most complex, fluffy, detailed, contrasted, spectacular and visually appealing clouds. Thus, our main goal is to design a method capable of achieving the realtime, realistic rendering of cumulus clouds similar those shown on Figure 1.1. The following sections provide more information on the definitions of "realtime" and "realistic".

Since the users of these methods are artists designing clouds for games or visual effect, our methods have to be user-friendly allowing simple controls. If a method produces very realistic clouds but the artist cannot control their general look, shape or trajectory, this method will not be implemented into a special effect company's toolbox. For example, a cloud modelling application that requires the user to input all atmospheric parameters is far less user-friendly than one that lets the user draw a general shape and infers the details. Similarly a user-friendly rendering application would be one that allows the user to easily control the dominant colors of the cloud.

Moreover, we want our methods to be animation-friendly. Indeed, some techniques are able to reproduce complex realistic illumination effects in realtime, such as precomputed radiance transfer (PRT, see §2.6.3), but they prevent the objects from being animated since they rely on heavy precomputations. Clouds are constantly changing objects. Two photographs of the same cloud taken at a ten second interval show visible differences. Thus, we want to be able to handle animated clouds.

1.1.3 Realtime

In CG, *realtime* usually means that the application is capable of rendering at a rate of at least 30 frames per second (FPS). This means that the computation of one image should take less than 30 ms. If the frame rate is lower than this limit but is still at least a few FPS, the application can be labelled "*interactive*". Applications running at these speeds may become realtime as the computing power of hardware increases with time or further optimizations are applied.

realtime

interactive

Slower speeds are called "*offline*". In particular, computer-generated visual effects sometime require several hours of computation per frame, which may be compensated for by massive computing power. Even a large increase of computing power would not allow offline methods to become realtime.

offline

1.1.4 Realism

1.1.4.1 Definition

The problem of the word "realistic" is that there is no standard definition for it in computer graphics. Ferwerda has made efforts to clarify this problem and has proposed several definitions for the word [Fer03].

The first definition is *physical realism* which is to reproduce the exact *visual stimulation* a human eye would receive in a real scene, *i.e.,* the same luminance field, at a resolution matching that of the eye. However, this is not possible mainly because computer screens are not able to reproduce the high ranges of luminance we see in reality. Moreover, and maybe most importantly, human vision has some limitations (examples below) we could take advantage of. As a result, the realism we aim for is *photorealism*, *i.e.,* to produce images that trigger the same *visual response* as the real scene, even if the luminance field is different.

physical realism

photorealism

As an example of the limitation of vision: the human eye is not sensitive to some parts of the light spectrum such as infrared and ultraviolet. This allows us to ignore these spectra (*i.e.,* not compute them). By doing this we can save computation time and optimize our algorithms for the visible spectrum. Another example is that if the light is too intense the eye becomes saturated and does not see the brightness variations. Thus it might not be necessary to compute these variations very faithfully.

A less trivial example is that the human eye is a saliency detector, which implies that it is more important to faithfully reproduce the gradients of an image than its flat tints. In our case, it means we

have to pay special attention to the edges of a cloud, while the shadowed, soft parts may contain more errors. The other implication of the eye being a gradient detector is that an error in the computation of an image will be more visible if it has a high spatial frequency: shifting the colors of a whole image is easily forgiven by the eye, but adding white noise to it, even to a small extent, is very distracting.

1.1.4.2 Measure

The main limitation of the photorealism criterion is that it is very difficult to have a good measure of it. We do not want to compare the resulting luminance field we compute with a real one, we want to compare the resulting visual response, which is located somewhere unknown in the human brain. There exist a few perceptual metrics of realism [Dal92, RPG99] but they are still very low-level, *i.e.,* they can tell if two images will be considered undistinguishable but they are not able to give answers such as "this image looks like it is the picture of a cloud", which is ideally the kind of measure we would like.

Such approaches (comparing a CG image and a reference image using perceptual metrics or user studies) have been used for participating media [SGA+07] and proved their usefulness. To be able to use these approaches we would need to try to reproduce an existing cloud and compare our results against a photograph of this cloud. Unfortunately, current sensing tools are far from being accurate, fast and precise enough to capture a whole, high-resolution, 3D cloud shape (including its interior) at one precise moment. As a result, these metrics are useless for us.

Figure 1.2: *Photograph of a* mammatus *cloud. Is this realistic or unrealistic ?*

Another way of measuring this realism is to do a user study asking a panel of people to decide whether the clouds presented to them are realistic or not. This poses many issues. A picture might be judged unrealistic just because the surroundings of the clouds are unrealistic, or because the shape of the cloud itself looks unrealistic even though the light transport computation is accurate. After all, even some real clouds look unrealistic (see Figure 1.2). Depending on the people surveyed, the answer will be different. A meteorologist would consider a cloud unrealistic simply because it does not fit with the surrounding weather, regardless of the rendering quality. An artist would care more about the aesthetic appeal of the picture than anything else. A video game user might find the clouds from his/her favorite game more realistic than real ones. Even though the target applications are video games and special effects, our target population consists of humans in general and not, say, gamers or movie fans.

As a result, in order to have a metric as objective as possible, we chose the following qualitative criterion. In a first step, we study real clouds and identify the characteristics and *visual features* that are common to clouds (these are detailed in §2.2). Our criterion for realism is that the pictures we create display these visual features. Using this approach we inspire from real clouds pictures and we compare our results to real cloud pictures. This criterion can be easily assessed by the users, readers and reviewers, and does not necessarily need a user study. Obviously, its difficulty lies in clearly identifying all the visual features of clouds.

1.1.5 Methodology

Computer graphics research usually follows two different approaches: the so-called "physical" approach and the "phenomenological" approach.

1.1.5.1 The "physical" methodology

physical approach

phenomenological approach

The *physical approach* consists in reproducing the local laws of physics in a computationally efficient way and use them to render an image. The *phenomenological approach* rather uses an inverse approach. It consists in finding a computational model capable of reproducing an observed phenomenon, regardless of the underlying physics that cause it.

The advantages of the physical approach is that it relies on numerous works and studies on the matter. It has, however, many pitfalls. First, the laws of physics are themselves approximations of real phenomenons[1] that correspond to ideal cases[2]. As an example, geometric optics assume the objects are much larger than the wavelength of light and thus fail completely at determining the optics of a cloud droplet. Second, these physical laws rely on boundary conditions, initial conditions and numerous parameters, which are usually very complex in real cases if not unknown. As a result a method that reproduces very faithfully the laws of physics but relies on simplified boundary conditions is likely to give poor results. Third, the criterion used to measure realism is very often a quantitative measure (*e.g.,* a least squares error) that does not consider any perceptual consequences (see §1.1.4). Finally, it is difficult for a human to predict the results of a method relying on multiple microphysical parameters whereas an artist needs understandable meaningful parameters.

[1] In a common joke among astrophysicists, various people are asked how they might increase milk production on a farm. The expert in animal husbandry says, "Consider the role of the cow's diet." The engineer says, "Consider the design of the milking machines." The astrophysicist says, "Consider a spherical cow..."

[2] In consequence, it is rather inappropriate to label "ground truth" images produced using physical methods.

In consequence, it is very easy to be blinded by the illusion of theoretical realism when using the physical approach. By only caring about whether the equations of physics are correctly computed, one might easily not realize that what is being computed is actually of no visible significance or, worse, that the resolution of the result is so coarse that these equations make no sense anymore (see §2.6.3.4). The problem here is that no other criteria (and especially no perceptual criteria) are generally used to assess the quality of the result other than the faithfulness of the reproduction of the formulas.

Moreover, it is very difficult to obtain realtime speeds using this approach. Indeed, the current state of the art in the physical approach (as seen in CG) usually consists in partial differential equations describing a behavior at microscopic scale. These equations usually require heavy computational methods (*e.g.,* a Monte-Carlo approach or a finite elements method, see §2.6.3.1) to be solved. Note that there exists physical laws describing phenomenons in a more macroscopic or mesoscopic way [Ray83]. These kind of descriptions may lead to more efficient computational models.

1.1.5.2 The "phenomenological" methodology

Phenomenological methods take the opposite approach, which is to aim at a specific output, regardless of microphysical generative models. That is, the goal is to reproduce the phenomenon, not the underlying mechanisms that cause it.

The first drawback of this method is that the more complex the phenomenon is, the more difficult it is to analyze and reproduce. Second, there does not currently exist sufficiently sound criteria for the quality of the result (see §1.1.4). Third, this inductive strategy is also the one used by physicists to establish the aforementioned equations, the current state of the art being the result of centuries of research. Thus, it seems unproductive to start from scratch and try to rediscover in a few months all that was found by physicists.

1.1.5.3 Our methodology

As a consequence, we would like instead to take the best of both worlds. First, we use criteria for realism that are based on human perception rather than on blind quantitative measures. These criteria are described in §1.1.4, and rely on identifying the visual features of clouds. Since these criteria only rely on somewhat subjective descriptions rather than sound scientific measures, the reader is invited to appreciate the quality of the results.

Second, since light transport in scattering media (which include clouds) is a very complex and widely studied phenomenon, we rely on the physics on radiative transfer instead of trying to reinvent the wheel. We use these laws for two purposes. The first is to identify the physical origin of the visual features of clouds. Since we seek to reproduce these visual features, we will focus on finding mechanisms that reproduce them rather than reproducing light transport in general. Thus, the second purpose is to rely on physical simulations to find new, mesoscopic, computationally efficient, humanly understandable, phenomenological models of light transport in clouds that match these simulations. This approach may be called a "physically-based phenomenological" approach.

Summary of §1.1 : The goal of this thesis is to design a method capable of achieving the realtime, realistic rendering of cumulus clouds. The criterion for realtime is the rendering framerate. The criterion for realism is the faithful reproduction of the visual features seen in real clouds. To achieve

this goal we use a physically-based phenomenological methodology. This methodology allows us to understand the physics of light transport in clouds, the origin of their visual features, and to derive sound, efficient, phenomenological models of light transport in clouds.

Le but de cette thèse est de concevoir une méthode capable de rendre des nuages cumuliformes réalistes en temps réel. Le critère du temps réel est défini via la vitesse de rendu. Le critère du réalisme est la reproduction fidèle des caractéristiques visuelles vus dans les nuages réels. Afin d'atteindre ce but nous utilisons une méthodologie phénoménologique basée sur la physique. Cette méthodologie nous permet de comprendre la physique du transport lumineux dans les nuages, l'origine de leurs caractéristiques visuelles, et d'en dériver des modèles phénoménologiques du transport lumineux dans les nuages à la fois solides et efficaces.

1.2 Our contributions

The main contributions of this thesis include

- A study of clouds as see from a human observer, describing the *visual features* that characterize clouds (§2.2). These features are the ones this thesis seeks to reproduce in order to achieve realism.
- A study of light transport in a slab of cloud, analyzing the behavior of light in this slab and seeking the origin of these visual features (Chapter 4).
- New phenomenological models for the transport of light in a slab of clouds (Chapter 5) derived from the results of our study.
- A new method for the realistic rendering of stratiform clouds using these phenomenological models (Chapter 6).
- A realtime implementation of this method on graphics hardware (Chapter 6).
- A new method for the realistic rendering of cumuliform clouds using these models (Chapter 7).
- An interactive implementation of this method on graphics hardware (Chapter 7).

1.3 Overview of the thesis

This thesis is divided into three main parts.

Part I defines the necessary background and is divided into three chapters. Chapter 1 presents our goals and methodology and defines the units used in this document. Chapter 2 presents the visual features of real clouds that we use as criterion for realism, explains the physics of radiative transfer and recalls the previous works done in clouds rendering. Chapter 3 give more details about some of the mathematical and computational tools we use in this thesis.

Part II presents our study of light transport in clouds in two chapters. Chapter 4 introduces our study and its goals, explains its implementation and exhaustively presents our analysis of its results. Chapter 5 relies on this analysis to propose new physically-based phenomenological models of light transport in a slab of cloud.

Finally, Part III makes use of these models to propose two new approaches for the realtime realistic rendering of clouds. The first approach, described in Chapter 6, is simple and aimed at the rendering of stratiform clouds. The second approach, presented in Chapter 7, is more complex and addresses all types of clouds, with an emphasis on cumulus-type clouds.

1.4 Mathematical bases

1.4.1 Notational conventions

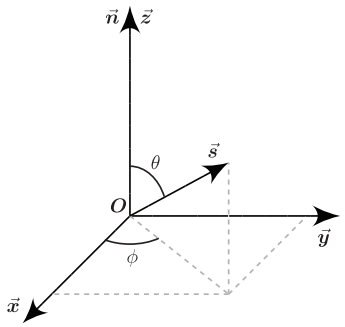


Figure 1.3: *Reference frame for spherical coordinates.*

We use boldface to distinguish vectors ($\boldsymbol{p}$) from scalars (x). We use arrows to distinguish vectors indicating directions ($\vec{\boldsymbol{s}}$) from vectors indicating points in space ($\boldsymbol{p}$). The use of uppercase or lowercase letters has no particular meaning. We write Dirac's delta function centered in x as δ_x. We write the components of a point $\boldsymbol{p}$ or vector $\vec{\boldsymbol{s}}$ as $\boldsymbol{p} = (x_{\boldsymbol{p}}, y_{\boldsymbol{p}}, z_{\boldsymbol{p}})$ and $\vec{\boldsymbol{s}} = (x_{\vec{\boldsymbol{s}}}, y_{\vec{\boldsymbol{s}}}, z_{\vec{\boldsymbol{s}}})$, respectively. The norm of a vector is written $\|\vec{\boldsymbol{s}}\|$.

We use a consistent notation, *i.e.*, x, l are always 1D lengths, $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{p}$ are always points, σ is always a surface, ω is always a solid angle, $\vec{\boldsymbol{s}}$ is always a direction, Θ, θ, ϕ, ψ are always scalar angles, ρ is always a concentration (number density), etc. The symbols that are commonly used throughout the thesis are summarized in the notation index page 257.

We use the reference frame as shown on Figure 1.3 for spherical coordinates. When considering a surface, this frame is aligned with the surface normal $\vec{\boldsymbol{n}}$ with $\vec{\boldsymbol{z}} = \vec{\boldsymbol{n}}$.

1.4.2 Solid angles

solid angle

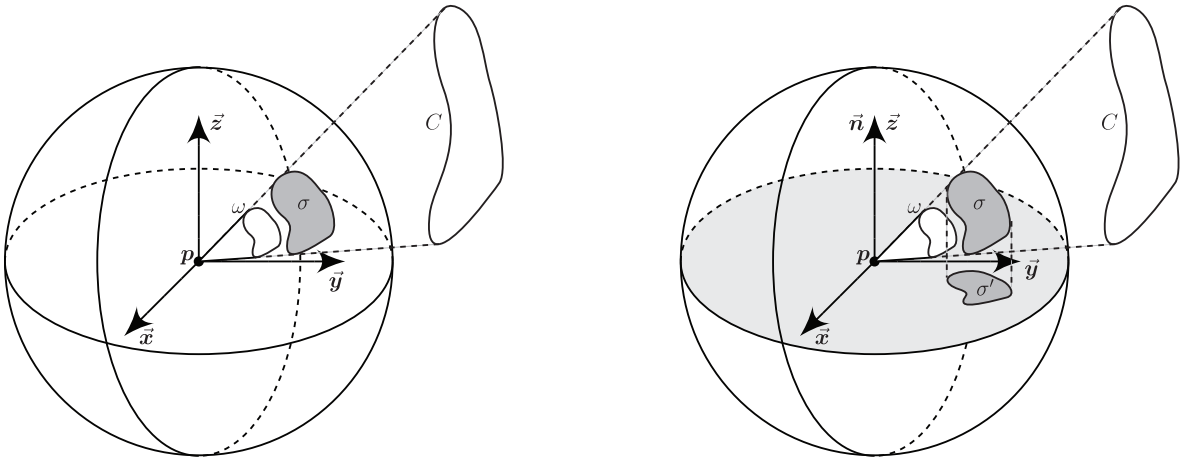
A *solid angle* is the term used to define a set of directions in space, just like an area is the term used to define a set of points in the plane. A solid angle is written as ω in this thesis. Its unit is the steradian, denoted sr. The solid angle ω subtended by a curve C in space as viewed from a point $\boldsymbol{p}$ is equal to the area σ the curve covers when projected onto the unit sphere centered around $\boldsymbol{p}$ (see Figure 1.4(a)). Thus, the solid angle subtended by a complete sphere is 4π sr, and the solid angle subtended by a hemisphere is 2π sr.

projected solid angle

A *projected solid angle* Ω subtended by a curve C on a point $\boldsymbol{p}$ of a plane of normal $\vec{\boldsymbol{n}}$ is the area σ' the curve covers when the projection σ of C on the unit sphere is projected along $-\vec{\boldsymbol{n}}$ onto the plane (see Figure 1.4(b)). For an infinitesimal solid angle $d\omega$, the corresponding projected solid angle is equal to $\Omega' = \cos\theta d\omega$.

If S and S' are sufficiently away from each other (*i.e.*, $\|\boldsymbol{p}' - \boldsymbol{p}\|^2 \gg S'$), the solid angle ω subtended in $\boldsymbol{p}$ by a surface of area S' and normal $\vec{\boldsymbol{n}}'$ centered in $\boldsymbol{p}'$ (see Figure 1.5) is

$$\omega = \frac{S'[\vec{\boldsymbol{n}} \cdot (\boldsymbol{p}' - \boldsymbol{p})]}{\|\boldsymbol{p}' - \boldsymbol{p}\|^2} = \frac{S' \cos\theta'}{\|\boldsymbol{p}' - \boldsymbol{p}\|^2}.$$



(a) Solid angle (b) Projected solid angle

Figure 1.4: *Solid angle and projected solid angle.*

Similarly, the projected solid angle Ω subtended on a surface S of normal $\vec{\boldsymbol{n}}$ in $\boldsymbol{p}$ by a surface of area S' and normal $\vec{\boldsymbol{n}}'$ centered in $\boldsymbol{p}'$ (see Figure 1.5) is

$$\Omega = \frac{S'[\vec{\boldsymbol{n}}' \cdot (\boldsymbol{p} - \boldsymbol{p}')][\vec{\boldsymbol{n}} \cdot (\boldsymbol{p}' - \boldsymbol{p})]}{\|\boldsymbol{p}' - \boldsymbol{p}\|^2} = \frac{S' \cos\theta \cos\theta'}{\|\boldsymbol{p}' - \boldsymbol{p}\|^2}.$$

In spherical coordinates, an element of solid angle $d\omega$ is equal to

$$d\omega = \sin\theta d\theta d\phi,$$

and an element of projected solid angle $d\Omega$ is equal to

$$d\Omega = \cos\theta d\omega = \cos\theta \sin\theta d\theta d\phi.$$

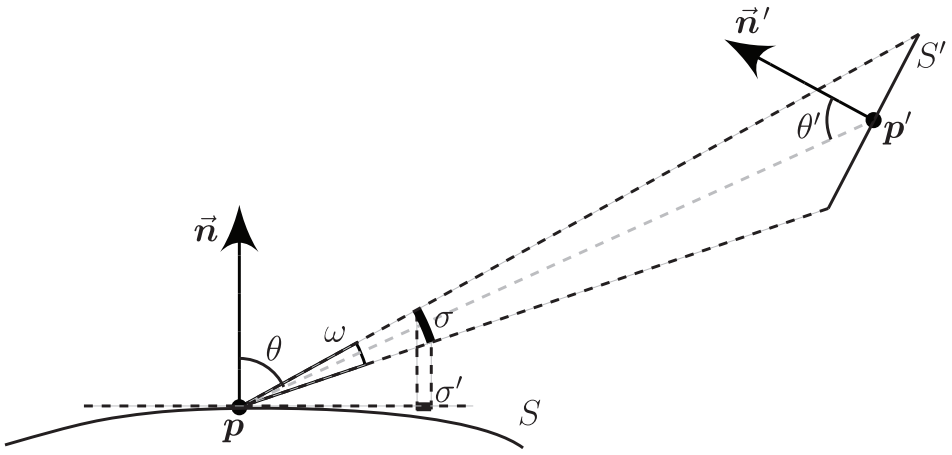


Figure 1.5: *Solid angle $\omega = \sigma$ and projected solid angle $\Omega = \sigma'$ subtended by a surface S' on S in $\boldsymbol{p}$.*

1.5 Radiometric and photometric units

We introduce in this section the basic units we will use throughout this thesis. As we are dealing with rendering and light transport problems, we will manipulate many units related to light [NRH+77]. Radiometry and photometry are two systems describing and measuring light.

1.5.1 Differences between radiometry and photometry

Light is electromagnetic radiation. Radiometry and photometry are two different ways of measuring radiation. The difference is the point of view. Radiometry considers all radiation, while photometry considers only radiation visible by humans.

wavelength

In radiometry, when measuring a radiation at a given *wavelength* λ, the radiation is weighted by the energy a photon has at this wavelength. In photometry, when measuring radiation at a given wavelength, it is weighted by the eye's sensitivity to this wavelength. This weighting function is classically noted $v(\lambda)$. The difference between radiometry and photometry is the use of a different $v(\lambda)$. We are only interested in the weighting function of photometry. The $v(\lambda)$ function for photometry is shown on Figure 1.6(a). This function is also called *luminous efficiency*. It can be split into three different functions $v_R(\lambda)$, $v_G(\lambda)$, $v_B(\lambda)$ corresponding to the weighting functions of each of the three types of light sensors in the human eye and corresponding (roughly) to our sensitivity to red, green and blue colors, respectively (see Figure 1.6(b)).

luminous efficiency

Note that the direct conversion possible between photometric and radiometric quantities is not possible. An invisible flux having a very high radiant power has no luminous power since it is not visible.

In this work, we are interested in photometric units. Since the difference between radiometry and photometry is only the final weighting function, any radiometric theory can also apply to photometry. This includes radiative transfer, which we will make a heavy use of. Most of the background theories and previous work we draw on (including most CG works) have used radiometric units. As a result, we introduce units from both fields and alternate between them within this thesis.

All the units we introduce in this section are monochromatic, *i.e.,* they are defined for a given wavelength (*i.e.,* a given color). As a result, to compute an image one should compute the luminous flux arriving to the camera for all visible wavelengths. Since this may be fairly complex, a very common solution is to compute the luminous flux only for three different channels corresponding to the red, green and blue primary colors. Note that doing this is already an approximation of light transport, although it generally produces visually satisfying images. Proposing new methods for the faithful computation of colors is out of the scope of this thesis, hence we use this classical RGB approach.

1.5.2 Interaction between light and matter

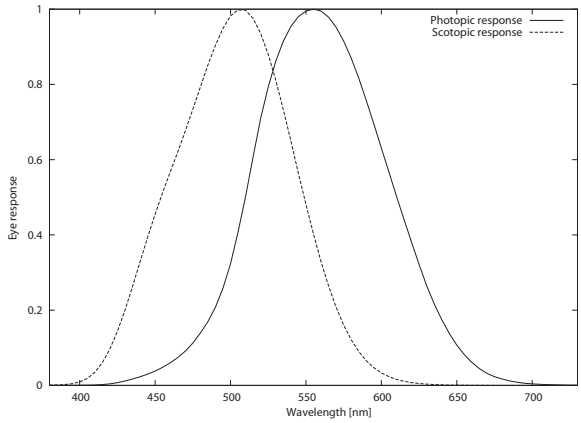
Matter can either *absorb*, *emit* or *scatter* light. Absorption means some of the light is transformed by matter in some other energy (*e.g.,* heat). As an example, black paint absorbs more light than white paint. Emission means that matter emits light by itself from other sources of energy (*e.g.,* melting lava or metal). Scattering means that matter re-emits some of the radiation it receives.

absorption coefficient

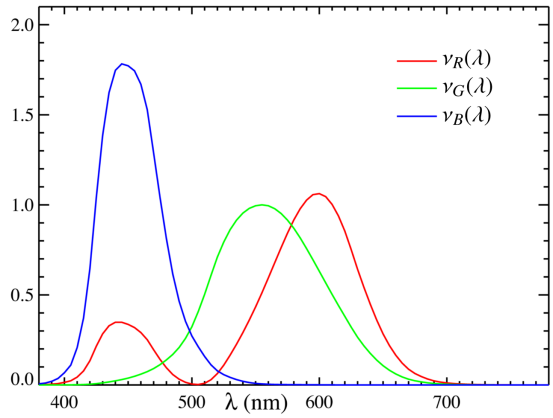
The absorption and scattering properties of a material or particle depend on the *absorption coefficient* κ_a and the *scattering coefficient* κ_s. The *extinction coefficient* κ_e is defined by $\kappa_e = \kappa_s + \kappa_a$. The ratio between scattering and extinction κ_s/κ_e is called the *single scattering albedo* and is noted ϖ_0. For clarity we call it *albedo* in the rest of this document. It varies between 0 and 1. For solid diffusive

scattering coefficient

extinction coefficient



(a) $v(\lambda)$ for the photopic receptor (cone cells, perceiving bright light) and scotopic receptor (rod cells, perceiving dim light) of the human eye.



(b) $v(\lambda)$ for each of the three types of cone cells in the human eye. Rod cells do not distinguish colors.

Figure 1.6: *Luminous efficiency functions in photometry.*

objects it can roughly represent the "brightness" of an object (*e.g.,* the albedo of asphalt is $\varpi_0 \approx 0.05$ while $\varpi_0 \approx 0.4$ for desert sand).

In the case of cloud droplets and for visible light, there is no absorption (more precisely, the absorption coefficient of a cloud droplet is $\kappa_a \approx 10^{-6}$, which is negligible), only scattering. Thus, the albedo of cloud droplets is 1. As we will see in §2.5, this makes a cloud very complex to model in terms of light transport.

It is important to note that only the transport of visible light (*i.e.,* in the wavelength range of $380\,\text{nm} \leq \lambda \leq 750\,\text{nm}$) will be considered in this work. The physics of light transport in clouds is highly dependent on the wavelength. Thus, the approximations and optimizations we use in this thesis may not be directly applicable to other parts of the light spectrum.

1.5.3 Luminous energy - radiant energy

radiant energy

The *radiant energy* represents an energy in the classical sense. It is measured in Joules (J). As an example, the sun holds about 1.28×10^{45} J of radiant energy.

luminous energy

The *luminous energy* is expressed in lumen seconds (lm.s). In measurements luminous energy can be used to describe the amount of light that has been emitted by a light source over a given period. For instance, a light bulb emits two times more luminous energy in two seconds than in one. Roughly speaking, we can see it as the "number of photons" emitted by a source.

1.5.4 Luminous flux - radiant flux

luminous flux

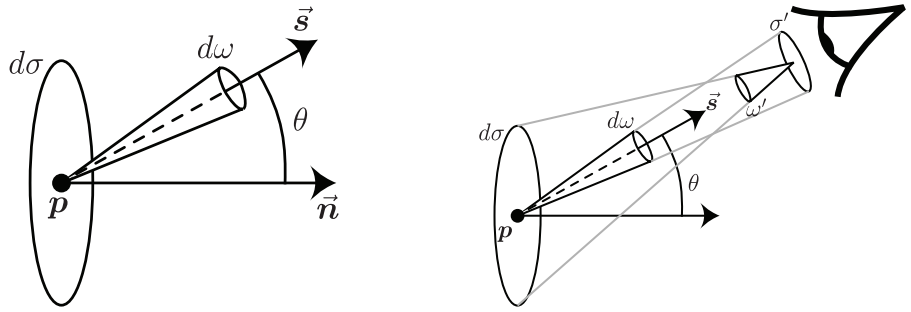
luminous power

radiant flux

radiant power

lumens

The *luminous flux* (or *luminous power*, or *radiant flux*, or *radiant power*) Φ is the unit to measure a variation of energy in time (*i.e.,* a "number of photons per second"). Radiant flux is measured in watts ($1\,\text{W}=1\,\text{J/s}$), and luminous flux in *lumens* (lm). This is a base unit in radiometry and photometry, from which other units are derived. As an example, the sun emits 3.846×10^{26} W, or 3.75×10^{28} lm.



(a) Luminance is the flux emitted from an element of surface $d\sigma$ into an element of projected solid angle $\cos\theta d\omega$ in a direction $\vec{s}$.

(b) When luminance is measured, the element of surface $d\sigma$ is subtended by the solid angle ω' of the receptor, while the solid angle $d\omega$ is subtended by the area σ' of the receptor.

Figure 1.7: *Luminance*

1.5.5 Luminous intensity - radiant intensity

luminous intensity

The *luminous intensity* $I(\boldsymbol{p}, \vec{s})$ is another base unit in photometry. It represents the luminous flux $d\Phi$ flowing from a given point $\boldsymbol{p}$ in a given direction $\vec{s}$ in an element of solid angle $d\omega$:

$$d\Phi = I(\boldsymbol{p}, \vec{s})d\omega.$$

As an example, the sun has a luminous intensity that is constant in every direction, while a flashlight has an high intensity in the forward direction and no intensity in the backward direction. It is measured in *candela* which are lumen per steradian ($\mathrm{lm.sr^{-1}}$). The radiometric equivalent is the *radiant intensity* in $\mathrm{W.sr^{-1}}$.

candela

radiant intensity

1.5.6 Luminance - radiance

luminance

The *luminance* $L(\boldsymbol{p}, \theta, \phi)$ is a luminous flux $d\Phi$ flowing through an element of area $d\sigma$ in directions confined to an element of *projected* solid angle $d\Omega = \cos\theta d\omega$ around a direction $\vec{s}$ (see Figure 1.7(a)). In other words, it is an intensity restricted to an element of surface. This flux is expressed by

$$d\Phi = L(\boldsymbol{p}, \theta, \phi)\cos\theta d\omega d\sigma$$

The luminance is the unit we will work the most with throughout this thesis. It is the unit that represents what an eye (or a camera) sees.

Indeed, consider a human eye (or camera) looking at a given surface S (see Figure 1.7(b) and 1.8). One infinitesimal photoreceptor in the eye (or one photodetector in the image sensor of the camera) of area σ', viewing through an small solid angle ω', sees an infinitesimal area $d\sigma$ of this surface. Every point $\boldsymbol{p}$ on this area $d\sigma$ emits a given luminous intensity $I(\boldsymbol{p}, \vec{s})$ in the direction $\vec{s}$ of the observer over an infinitesimal angle $d\omega$. The luminance $L(\boldsymbol{p}, \theta, \phi)$ represents this luminous flux

$$L(\boldsymbol{p}, \theta, \phi) = \frac{dI(\boldsymbol{p}, \vec{s})}{\cos\theta d\sigma}.$$

nit

It is expressed in $\mathrm{lm.sr^{-1}.m^{-2}}$, or $\mathrm{cd.m^{-2}}$. This unit is also called *nit*. The radiometry equivalent is

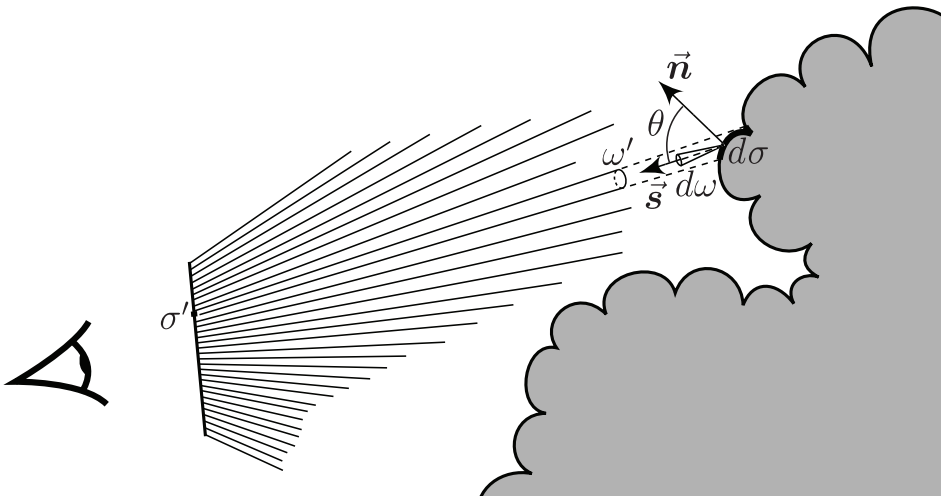


Figure 1.8: *When rendering an image (or taking a photograph), each pixel spans a small solid angle ω' subtending a small surface σ on the visible object in the scene. To render the image, one has to integrate the luminance over σ in the direction $\vec{s}$ of the camera, in the solid angle $d\omega$ subtended by the surface σ' of the captor. This is how light is perceived by a camera and the human eye.*

called *radiance* and is expressed in $\mathrm{W.sr^{-1}.m^{-2}}$.

radiance

It is important to note that luminance is constant with respect to the distance from the detector to the surface. Indeed, as we can see in Figure 1.7(b), as the distance from the viewer increases, $d\sigma$ increases but $d\omega$ decreases as much, thus $d\sigma d\omega$ stays constant. Similarly, luminance is constant with respect to θ. As we can see in Figure 1.7(b), $d\sigma$ is proportional to $1/\cos\theta$, thus $\cos\theta d\sigma$ stays constant as θ varies.

One last thing to be noted is that the element of surface $d\sigma$ does not need to correspond to an actual surface. It is just necessary that there is such a frame of reference. As a result, it is perfectly possible to measure the luminance of something that has no surface, such as the sky.

1.5.7 Illuminance - irradiance

The *illuminance* (or *irradiance*) $E(\boldsymbol{p})$ received by an element of surface $d\sigma$ at $\boldsymbol{p}$ represents the total luminous power *incident* on this surface from all directions. It is computed by

illuminance

irradiance

$$E(\boldsymbol{p}) = \frac{1}{2\pi}\int_0^{2\pi}\int_0^{\pi/2} dE(\boldsymbol{p},\theta,\phi)$$

where $dE(\boldsymbol{p},\theta,\phi)$ is an *element of illuminance* and represents the luminance incident on a surface flowing from an element of solid angle $d\omega$. It is computed by

$$dE(\boldsymbol{p},\theta,\phi) = L(\boldsymbol{p},\pi+\theta,\phi)\cos\theta d\omega.$$

Note that the difference between luminance and an element of illuminance (apart from the inverse direction) is a $\cos\theta$ factor. It comes from the fact that light coming from grazing angles has less power than light coming from normal angles. The unit of illuminance is lumen per square meter ($\mathrm{lm.m^{-2}}$), also called *lux* (lx). The unit of irradiance is Watt per square meter ($\mathrm{W.m^{-2}}$).

lux

1.5.8 Luminous exitance - Radiant exitance

exitance

The luminous or radiant *exitance* $M(\boldsymbol{p})$ of a surface is the luminous flux exiting from the surface at $\boldsymbol{p}$. It is thus defined similarly to irradiance as

$$M(\boldsymbol{p}) = \frac{1}{2\pi} \int_0^{2\pi} \int_0^{\pi/2} dM(\boldsymbol{p}, \theta, \phi)$$

where $dM(\boldsymbol{p}, \theta, \phi)$ is an *element of exitance* and represents the luminance exiting from a surface flowing from an element of solid angle $d\omega$. It is computed by

$$dM(\boldsymbol{p}, \theta, \phi) = L(\boldsymbol{p}, \theta, \phi) \cos\theta d\omega.$$

The unit of exitance is the same as for illuminance, *i.e.,* lux (lx) or Watt per square meter ($W.m^{-2}$).

1.5.9 Reflectance - BRDF

1.5.9.1 Definition

The reflectance R of an element of surface measures ratio between the *exiting radiance* reflected off the surface in direction $\vec{s}$ and the *incoming irradiance* arriving from direction $\vec{s}'$ (see Figure 1.9).

bidirectional reflectance distribution function

The reflectance of a surface at a point is described by a *bidirectional reflectance distribution function (BRDF)* (BRDF) [Nic65]

$$R(\boldsymbol{p}, \theta, \phi, \theta', \phi') = \frac{dL(\boldsymbol{p}, \theta, \phi)}{dE(\boldsymbol{p}, \theta', \phi')}.$$

Note that it is the ratio between a radiance and an irradiance, and not between radiances. Therefore, a BRDF is expressed in sr^{-1}. It is defined over the hemisphere with $0 \leq \phi \leq 2\pi$, $0 \leq \phi' \leq 2\pi$, $0 \leq \theta \leq \pi/2$ and $0 \leq \theta' \leq \pi/2$. Knowing the BRDF and the illuminance of a surface, its luminance can simply be computed by

$$L(\boldsymbol{p}, \theta, \phi) = \int_0^{2\pi} \int_0^{\pi/2} R(\boldsymbol{p}, \theta, \phi, \theta', \phi') dE(\boldsymbol{p}, \theta', \phi') \tag{1.1}$$

$$= \int_0^{2\pi} \int_0^{\pi/2} R(\boldsymbol{p}, \theta, \phi, \theta', \phi') L(\boldsymbol{p}, \pi + \theta', \phi') \cos\theta' \sin\theta' d\theta' d\phi'. \tag{1.2}$$

If a surface is illuminated by a directional source, *i.e.,* from a single direction (θ', ϕ'), Equation 1.2 simplifies to

$$L(\boldsymbol{p}, \theta, \phi) = R(\boldsymbol{p}, \theta, \phi, \theta', \phi') L(\boldsymbol{p}, \pi + \theta', \phi') \cos\theta'. \tag{1.3}$$

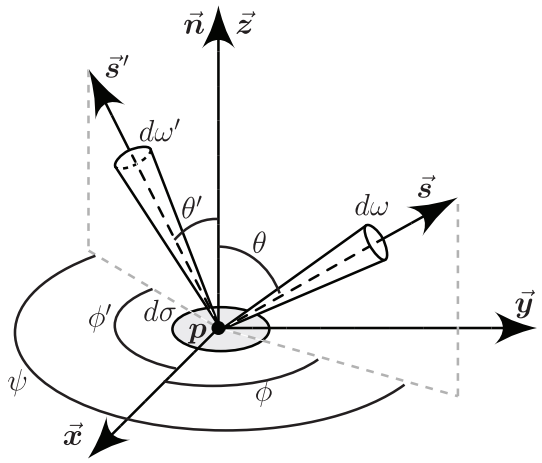


Figure 1.9: *Reference frame for a BRDF. A BRDF encodes the reflectance of the element of surface $d\sigma$ located at $\boldsymbol{p}$ with normal $\vec{\boldsymbol{n}}$. It is the ratio between the element of illuminance coming from $\vec{\boldsymbol{s}}'$ in the element of solid angle $d\omega'$ and the luminance towards $\vec{\boldsymbol{s}}$ in the element of solid angle $d\omega$.*

1.5.9.2 Properties of BRDFs

A BRDF is often defined as constant across the whole surface, in which case it depends only on the angles between the normal and the directions as $R(\theta, \phi, \theta', \phi')$. If it is spatially varying, it is usually called a bidirectional texture function (BTF). BRDF models have some properties:

- A physically-correct BRDF should respect the law of the conservation of energy

$$\int_0^{2\pi} \int_0^{-\pi/2} R(\boldsymbol{p}, \theta, \phi, \theta', \phi') \cos\theta' \sin\theta' d\theta' d\phi' \leq 1,$$

  be non-negative

$$R(\boldsymbol{p}, \theta, \phi, \theta', \phi') \geq 0,$$

  and respect the *reciprocity invariance*

reciprocity invariance

$$R(\theta, \phi, \theta', \phi') = R(\theta', \phi', \theta, \phi).$$

- The *rotational invariance* translates to

rotational invariance

$$R(\theta, \phi, \theta', \phi') = R(\theta, \phi'', \theta', \phi'''), \phi - \phi' = \phi'' - \phi'''.$$

  It this case, the BRDF can be written as

$$R(\theta, \psi, \theta'),$$

  with $\psi = \phi' - \phi$. A surface having a rotationally invariant BRDF is also said to be *isotropic*.

- A BRDF that is *bilaterally symmetric* means that

bilaterally symmetric BRDF

$$R(\theta, \phi, \theta', \phi') = R(\theta, -\phi, \theta', \phi').$$

Usually, a bilaterally symmetric BRDF is also rotationally invariant, which means it can be written as

$$R(\theta, \psi, \theta'),$$

with $\psi = |\phi' - \phi|$.

axially symmetric BRDF

- An *axially symmetric* depends only on θ and θ' and can thus be written

$$R(\theta, \theta').$$

isotropic BRDF

- Finally, an *isotropic BRDF* means that it is constant, *i.e.,* it has the same value for all directions.

The confusion should not be made between an *isotropic surface* and an *isotropic BRDF*. An *isotropic surface* implies that its BRDF is rotationally invariant, but not necessarily isotropic.

isotropic surface

1.5.9.3 BRDF Models

A BRDF can be measured from real surfaces using a gonioreflectometer [War92, Cor01], or can be modeled using physical or phenomenological models. Much work has been devoted to reproducing realistic BRDFs of real-world materials. The oldest, simplest existing BRDF model is that of a perfectly diffuse surface (or *lambertian* surface, from [Lam60]), where R is isotropic, *i.e.,*

diffuse surface

lambertian surface

$$R(\theta, \phi, \theta', \phi') = \varpi_0(\boldsymbol{p}),$$

where ϖ_0 is the albedo of the surface at $\boldsymbol{p}$ (see §1.5.2).

Other common BRDF models include the Phong model [Pho75], the Blinn-Phong model [Bli77], the microfacet (or Torrance-Sparrow) model [TS67], the Cook-Torrance model [CT82], the He-Torrance-Sillon-Greenberg model [HTSG91], and many others. These models aim at better reproduction of *specular* surfaces, *i.e.,* glossy surfaces such as metal or plastic. A mirror is a perfectly specular surface. For example, the Phong model is

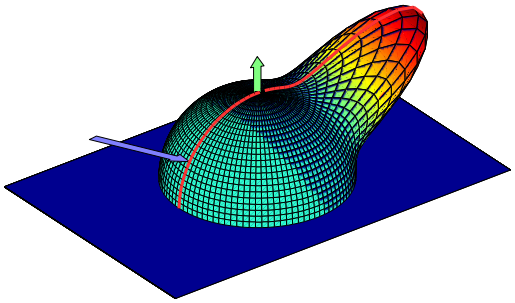
specular surface

$$R(\vec{s}, \vec{s}') = a + \frac{b}{\cos\theta'}(\vec{r} \cdot \vec{s})^c$$

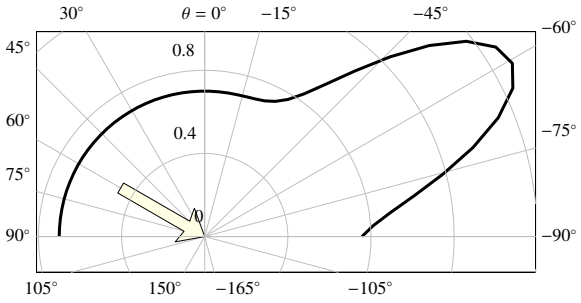
where $\vec{r}$ is the reflection of $\vec{s}'$ on the surface, *i.e.,* $\theta_{\vec{r}} = \theta'$ and $\phi_{\vec{r}} = \pi + \phi'$. The parameters of the model a, b and c define the aspect of the material.

1.5.9.4 Visualizing a BRDF

Since a BRDF is a 4-dimensional function, it is quite difficult to visualize. A common way of displaying such a function is to plot $R(\vec{s}, \vec{s}')$ in spherical coordinates with $\vec{s}'$ constant, which results in a hemispherical surface plot (see Figure 1.10(a)). In this case, $\vec{s}'$ should be chosen so as to be relevant and should be shown on the plot. A simpler, more readable way is to plot only $R(\theta, \psi, \theta')$ with θ' constant, $\psi = 0$ and $-\pi \leq \theta \leq \pi$ in polar coordinates (see Figure 1.10(b)). Although it is a very limited 1D view of a 4D function, it shows the most important aspects of the reflective aspect of a material.



(a) Plot of a Phong BRDF in 2 dimensions. The surface represents the BRDF for $\phi' = 0$ and $\theta' = \pi/3$. The green arrow represents the surface normal. The blue arrow represents the incident light direction.



(b) Plot of a BRDF in 1 dimension. The line represents the BRDF for $\phi = 0$, $\phi' = 0$ and $\theta' = \pi/3$, *i.e.,* the red line in Figure 1.10(a). The white arrow represents the incident light direction.

Figure 1.10: *Two ways of visualizing a BRDF.*

1.5.10 Subsurface scattering - BSSRDF

The BRDF model assumes that the luminance of a surface at a point $\boldsymbol{p}$ depends only on the illuminance at $\boldsymbol{p}$. This assumption may be valid for various type of opaque surfaces but does not always hold. For material such as marble or human skin, light passes through the surface, is absorbed and scattered, and eventually exits the material in another location and another direction. This behavior is called *subsurface scattering*. It can be modeled through as *bidirectional subsurface reflectance distribution function (BSSRDF)* (BSSRDF) [NRH+77, Kor69] that describes light transport between two elements surface of an object and two directions by

subsurface scattering

bidirectional subsurface reflectance distribution function

$$S(\boldsymbol{p}, \vec{\boldsymbol{s}}, \boldsymbol{p}', \vec{\boldsymbol{s}}') = \frac{dL(\boldsymbol{p}, \vec{\boldsymbol{s}})}{dE(\boldsymbol{p}', \vec{\boldsymbol{s}}')d\sigma'}.$$

It is expressed in $\text{sr}^{-1}.\text{m}^{-2}$. If S and E are known, then L can be computed as

$$L(\boldsymbol{p}, \theta, \phi) = \int_A \int_0^{2\pi} \int_0^{\pi/2} S(\boldsymbol{p}, \theta, \phi, \boldsymbol{p}', \theta', \phi') dE(\boldsymbol{p}', \theta', \phi') d\sigma'$$

where A is the lit surface of the object (see Figure 1.11).

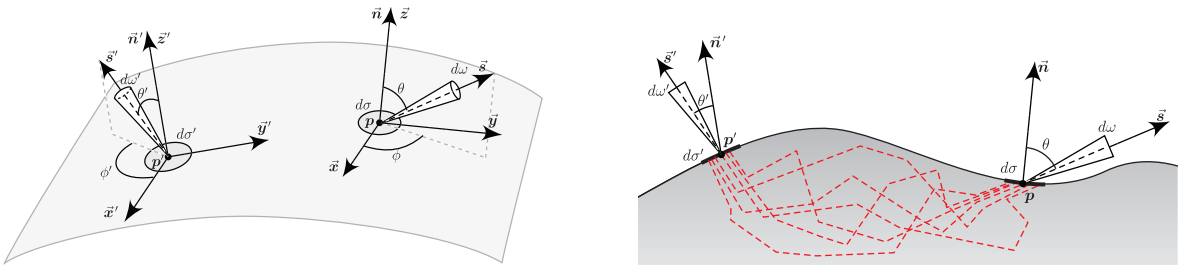


Figure 1.11: *A BSSRDF encodes the luminance transfer between all points and directions on a surface.*

Due to the high-dimensionality (8 degrees of freedom) of BSSRDFs, they are difficult to capture. Existing models for CG hence rely on numerical models. An accurate BSSRDF model would require to solve the radiative transfer equation, which is extremely time consuming, as we will see in Chapter 2. Computer graphics BSSRDF models rely on approximations such as single scattering [HK93] or diffusion approximation [JMLH01]. Further information on this subject will be given in §2.6.3.3.

1.5.11 Transmittance

transmittance

The term *transmittance* is sometimes used in the literature. It represents the amount of light that passes through a surface for which one can clearly define a lit side and an unlit side. It can be seen as a BRDF for transmitted light. It corresponds to what Chandrasekhar [Cha60] calls the *diffuse transmission function* in §2.5.5.

1.5.12 Typical values for atmospheric objects

Table 1.1 shows typical values that can be observed in reality. In particular it is important to note the difference between the luminance of the sun and that of the sky. The ratio between these two values is roughly 100,000 on a clear day. However, the sky spans a very wide solid angle (the whole hemisphere, *i.e.,* 2π sr). The sun on the other hand, with a radius of 1.392×10^9 m and a distance of 149.6×10^9 m from the earth, only spans 2.72×10^{-4} sr. As a result, the ratio $E_{\text{sky}}/E_{\text{sun}}$ between the illuminance due to the sky and the illuminance due to the sun is actually about 0.25 when the sun is at the zenith. That is, both values are of the same order. This observation is very important because it means that skylight is a significant light source in natural scenes, which shines on objects with its own color (blue). Moreover, this ratio $E_{\text{sky}}/E_{\text{sun}}$ increases with the sun angle, which means the sky can even become *the most important* light source for low elevation angles of the sun or when the sun has set (see Figure 1.12).

Similarly, the earth spans a 2π sr solid angle when viewed from the base of a cloud. As a result, even if its luminance is low, its large surface makes it a potentially important light source. Assuming it has a diffuse reflectance, its luminance is $E_{\text{sun}}\varpi_0/2\pi$. Thus, the ratio between the illuminance due to the ground and the illuminance due to the sun $E_{\text{ground}}/E_{\text{sun}}$ is exactly the average albedo ϖ_0. As a result, when ϖ_0 is close to 1 (*e.g.,* in the case of ice or snow), it makes the ground a light source as important as the sun, which should not be neglected. We discuss this further in §2.2.2.1

Table 1.1 Typical photometric values in natural scenes, retrieved from [Hal93, Con00, CRB+06, Wik08, BMS+02].

Measurement	**Range of observed values**
Luminance (§1.5.6) of the sun	900,000,000 nits – 1,600,000,000 nits
Luminance (§1.5.6) of the sky in the daytime	500 nits – 1,500 nits
Illuminance (§1.5.7) on the ground on a sunny day	50,000 lx – 100,000 lx
Illuminance (§1.5.7) on the ground due to the sun	40,000 lx – 80,000 lx
Illuminance (§1.5.7) on the ground due to the sky	10,000 lx – 20,000 lx
Luminance (§1.5.6) of the moon	2,000 nits – 4,000 nits
Luminance (§1.5.6) of the sky at night	≈ 0.0001 nits
Luminance (§1.5.6) of a cloudy sky in the daytime, measured from below	300 nits – 35,000 nits
Luminance (§1.5.6) of thick clouds in the daytime, measured from above	20,000,000 nits – 150,000,000 nits
Albedo (§1.5.2) of asphalt	0.04 – 0.12
Albedo (§1.5.2) of water	≈ 0.05
Albedo (§1.5.2) of trees	0.1 – 0.3
Albedo (§1.5.2) of green grass	≈ 0.25
Albedo (§1.5.2) of desert sand	≈ 0.4
Albedo (§1.5.2) of ice	0.7 – 0.8
Albedo (§1.5.2) of snow	0.7 – 0.95

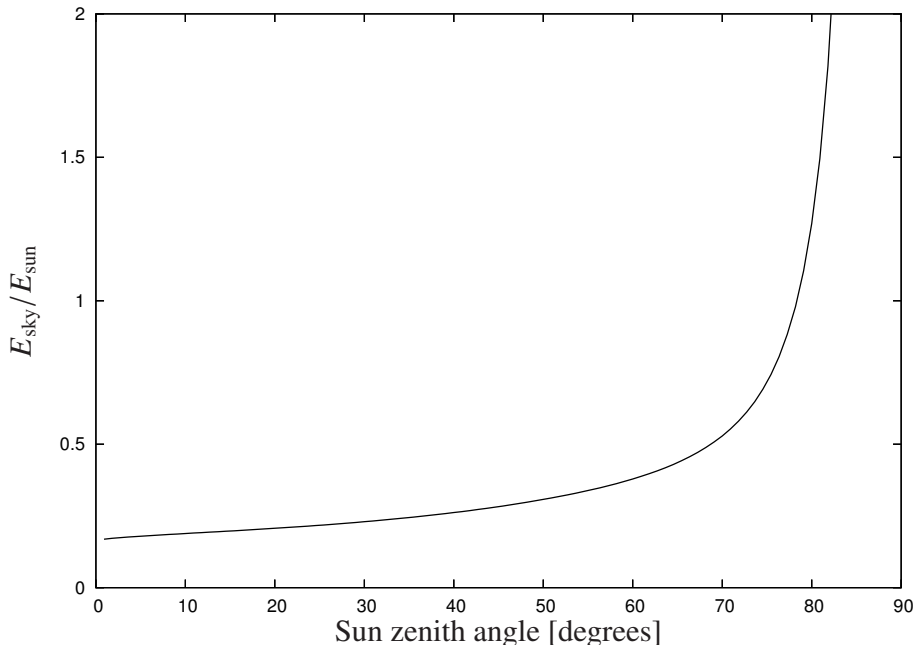


Figure 1.12: *Ratio of the illuminance E_{sky} due to the sky over the illuminance E_{sun} due to the sun on a clear day with respect to the sun's zenith angle, according to the CIE sky model #12 [DK02].*

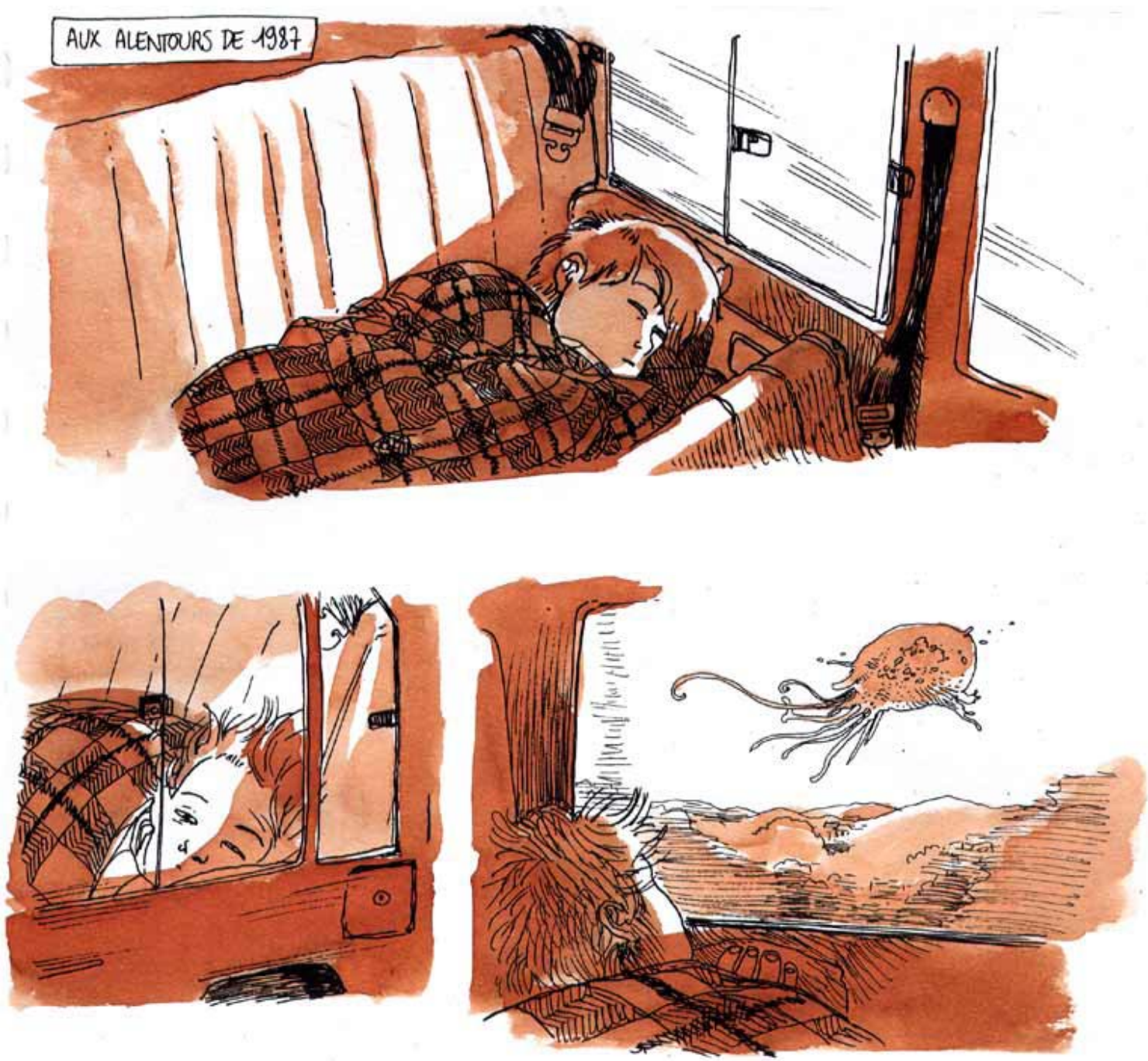
Summary of §1.5 : We introduced basic radiometry and photometry units that will be used in this work. Albedo, luminance, illuminance, exitance, reflectance (BRDF), transmittance (BTDF) and BSSRDF will be the most extensively used. We provided some values to give a sense of importance of the different elements in a cloud's environment. We have seen that the illuminance of the sky and of the ground are of importance comparable to the illuminance of the sun.

*Nous avons introduit les unités radiométriques et photométriques de base que seront utilisées dans ces travaux. L'albédo, la luminance, l'éclairement (*illuminance*), l'exitance, la réflectance (BRDF), la transmittance (BTDF) et la BSSRDF seront les plus utilisées. Nous avons fourni quelques valeurs d'exemple afin de donner une idée de l'importance des différents éléments autour d'un nuage. Nous avons vu que l'éclairement du ciel et du sol sont d'importance comparable à celle de l'éclairement du soleil.*

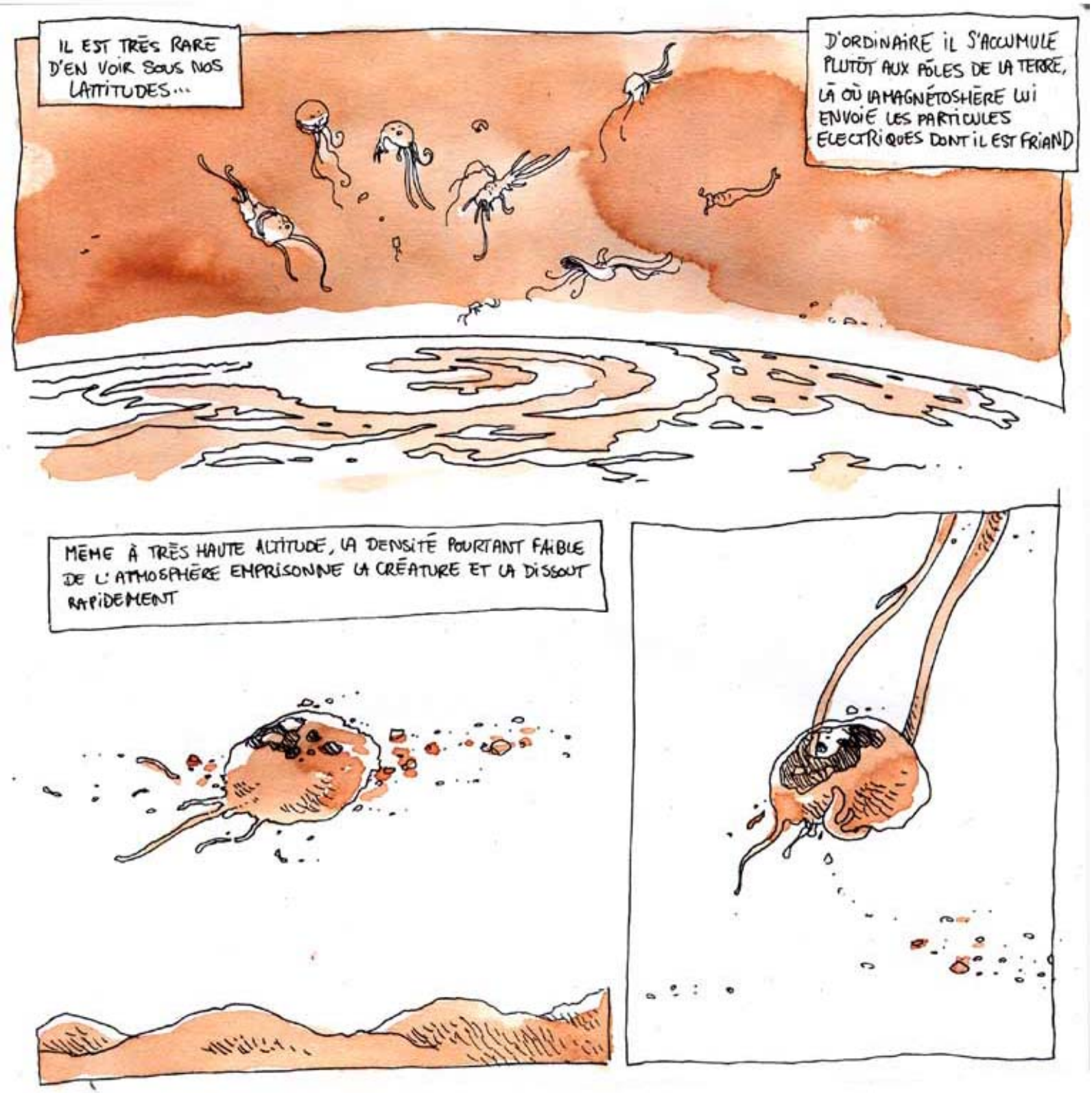
1.6 Summary of Chapter 1

In this chapter, we have introduced our objective and our methodology (§1.1). Our objective is the realtime realistic rendering of clouds. We have defined the criterion for realtime, which is the speed of rendering. We have defined the criterion for realism, which is the reproduction of a series of visual features that are typical of clouds. Our methodology consists in first identify these visual features (§2.2) then study the light transport of clouds to understand the origin of these visual features (Chapter 4) and find models reproducing them (Chapter 5) in order to propose new methods for the realtime realistic rendering of clouds (Chapters 6 and 7). We also have introduced in this chapter the necessary mathematical bases (§1.4) and units (§1.5) that will be used in the rest of this document.

Dans ce chapitre, nous avons présenté notre objectif et notre méthodologie (§1.1). Notre objectif est le rendu réaliste de nuages en temps réel. Le temps réel est défini par la rapidité de rendu. Le réalisme est défini par la reproduction d'un ensemble d'éléments visuels caractéristiques des nuages. Notre méthodologie consiste à identifier dans un premier temps ces caractéristiques visuelles (§2.2), puis étudier le transport de la lumière dans les nuages pour comprendre l'origine de ces caractéristiques visuelles (chapitre 4) et de trouver des modèles les reproduisant (Chapitre 5) afin de proposer de nouvelles méthodes de rendu réaliste de nuages en temps réel (Chapitres 6 et 7). Nous avons également présenté dans ce chapitre les bases mathématiques nécessaires (§1.4) et les unités (§1.5) utilisées dans le reste de ce document.

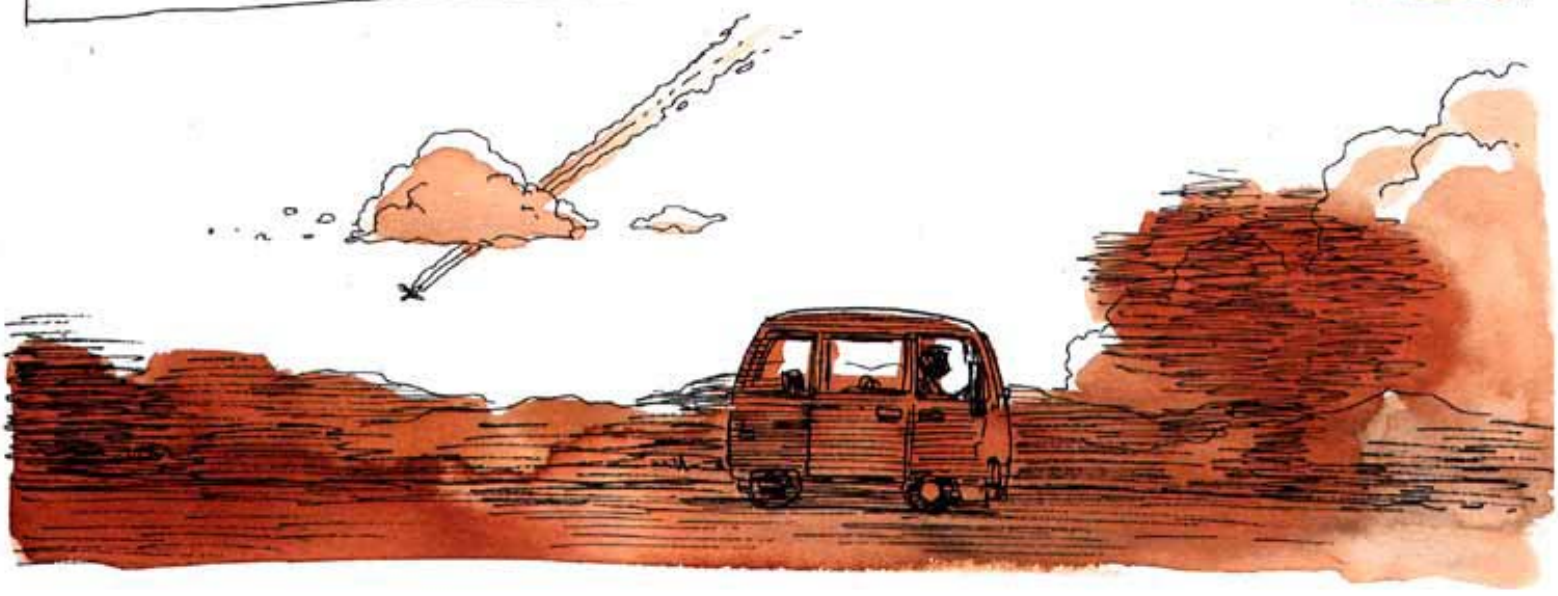


IL ARRIVE PARFOIS QU'UN PEU DE PLANCTON SPATIAL S'ÉGARE DANS L'ATHOSPHÈRE...


IL EST TRÈS RARE D'EN VOIR SOUS NOS LATITUDES...
D'ORDINAIRE IL S'ACCUMULE PLUTÔT AUX PÔLES DE LA TERRE, LÀ OÙ LA MAGNÉTOSHÈRE LUI ENVOIE LES PARTICULES ÉLECTRIQUES DONT IL EST FRIAND
MÊME À TRÈS HAUTE ALTITUDE, LA DENSITÉ POURTANT FAIBLE DE L'ATMOSPHÈRE EMPRISONNE LA CRÉATURE ET LA DISSOUT RAPIDEMENT


...À MOINS QU'ELLE NE SOIT HAPPÉE AVANT PAR UN DES PRÉDATEURS GÉANTS QUI PULLULENT AUTOUR DES PLANÈTES EN QUÊTE DE NOURRITURE


... TRÈS LÉGÈRES (CAR TRÈS PEU DENSES) ET INVISIBLES À L'ŒIL NU, LES CRÉATURES COSMIQUES N'ONT ÉTÉ DÉCOUVERTES QU'EN... QU'EN...
HUUUHOUAAAAAA...
PFFF...
GRAND LIVRE DE ASTRONOMIE



Chapter 2

What is a cloud ?

Contents

2.1 Introduction

Clouds have been studied extensively in a number of different areas. This chapter is devoted to the works that are useful in our research. The visual features we need to reproduce are identified in §2.2. The content and structure of clouds are defined in §2.4. The physical laws of radiative transfer, which describe how light interacts with clouds, is explained in §2.5. We present in §2.6 how previous research in computer graphics clouds have been modeled in previous studies.

2.2 The human spectator point of view

Analyzing photographs and artworks helps us identify the visual features we need to reproduce in clouds in order to fulfill our goal of realism. This section describes these features, starting from the most evident and trivial.

To reproduce real clouds, we first need to answer two questions that will help us identify their visual features: what is the shape of a cloud, and what is the color of a cloud ?

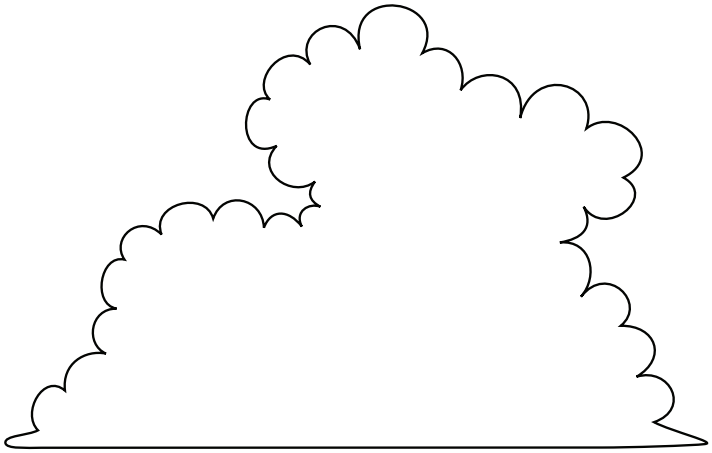
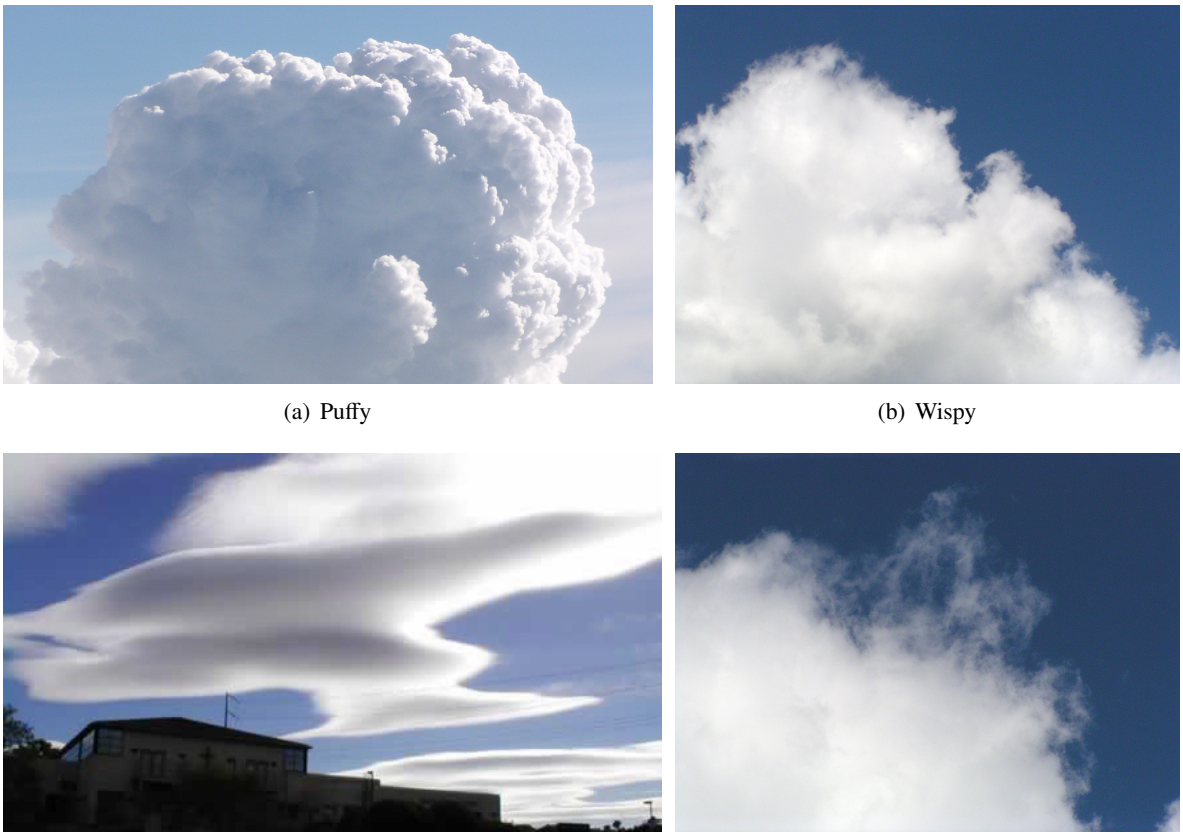


Figure 2.1: *A very abstract cloud.*

2.2.1 What is the shape of a cloud ?

2.2.1.1 Edges of a cloud

A simple sketch made of abstract blobby lines as shown on Figure 2.1 is enough for a human to recognize it as a cloud. In this sketch, the lines represent the edges and outlines of the cloud. This information tells us that the edges and outlines are very important in reproducing a cloud, and are one of the major factors telling the viewer that he/she is looking at a cloud. When observing these lines, we also notice that their shape is particular. It is made of circular curved paths, with smooth convex parts and sharp concave angles. These shapes reproduce the fluffiness of convective clouds, emphasizing their cotton-ball appearance. This observation tells us that the puffs are a strong visual feature of convective clouds (see Figure 2.2(a)).



(a) Puffy (b) Wispy

(c) Soft (d) Close-up on a wisp

Figure 2.2: *Some key aspects of the edges of a cloud.*

When taking a closer look at a real cloud, we can see that their shape is actually more complex than just a collection of convex puffs. When the boundary is not puffy and sharp, it often shows an unstructured and chaotic aspect. These stretched threads of clouds are called *wisps* (see Figure 2.2(b)) and are another strong visual feature of clouds. Moreover, these puffs and wisps have a fractal behavior. We can clearly see on Figure 2.2 that we can define puffs and wisps at various scale levels. Finally, in calm conditions a clouds can have a more vaporous, thin, soft aspect where its edges do not look wispy (see Figure 2.2(c)).

wisp

Thus, the edges of a cloud can have three distinct main aspects: either sharp and fluffy, curly and wispy, or soft and smooth.

2.2.1.2 Types of clouds

The aspect of a cloud's edges are strongly linked with the type of the cloud. Meteorologists have categorized clouds into different types according to their appearance [Wor87]. Figures 2.4, 2.11, 2.12 show typical examples of clouds. These types each have a Latin name, such as "stratocumulus". These names are formed from the following roots.

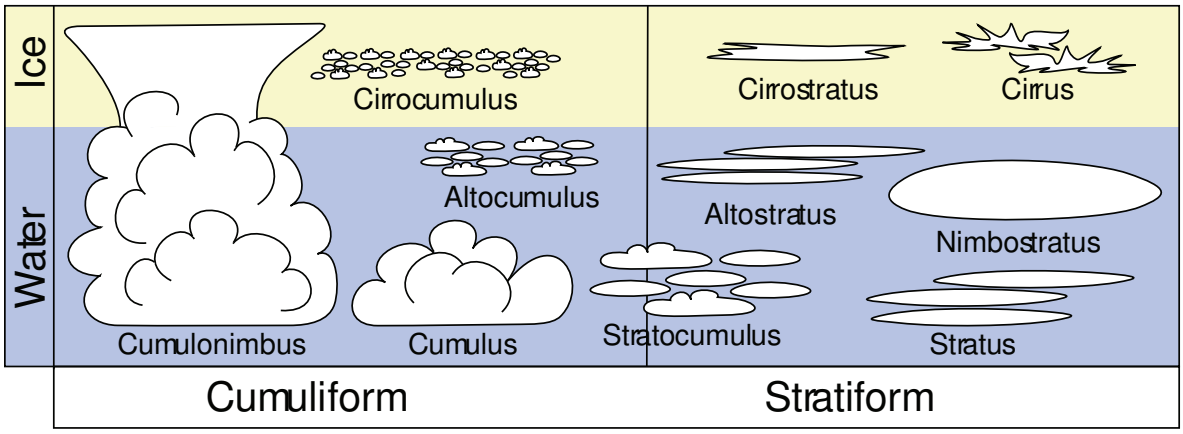


Figure 2.3: *Common cloud types.*

- *cumulus* or *cumulo-* means that the cloud has a puffy shape (*cumulus* is the Latin for "heap"). These clouds usually display hard edges, especially the edges of their puffs. The other parts of the clouds (*e.g.,* the base and the shadowed parts) can have a wispy or soft aspect.
- *stratus* or *strato-* means that the cloud has a stratiform shape, *i.e.,* that of a layer (*stratus* is the Latin for "layer"). These clouds are the ones seen in grey, smooth, overcast skies. It is not often possible to see their edges, but they are usually soft, sometimes wispy.
- *cirrus* or *cirro-* means that the whole cloud (not just its edges) has a wispy appearance (*cirrus* is the Latin for "curl of hair"). This is the case for very high altitude clouds formed of ice only.
- *nimbus* or *nimbo-* means that the cloud brings rain (*nimbus* is the Latin for "rain"). It is always associated with cumulus or stratus.
- The prefix *alto-* indicates that the cloud is at high altitude, but lower than a cirrus-type cloud. *Altum* means height in Latin.

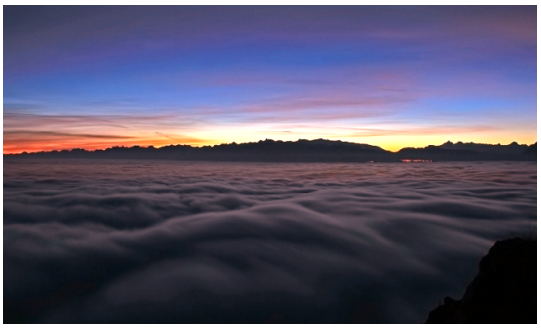
These types have further more precise subtypes. For example, a *cumulus humilis* means a small newly created cumulus (Figure 2.4(f)), while a *cumulus congestus* has grown to the maximum size for a cumulus (Figure 2.12(b)). There also exist very particular, rare types of clouds. A *lenticularis* (Figure 2.2(c)) is a cloud that has a lens-like appearance (consequence of updrafts from nearby hills). A *mammatus* (Figure 1.2) looks like it has breasts dangling on its underside (consequence of the raindrops evaporating as they fall).

For example, an *altostratus* cloud is a high, grey, layer-shaped cloud (Figure 2.12(d)). A *stratus* has the same look but is at lower altitude, while a *nimbostratus* is thicker and brings rain. A *cumulus* is a puffy cloud (Figure 2.4(e)), while a *cumulonimbus* is a *cumulus* that has grown, developed the classical anvil shape at its top, and brings rain and thunderstorms (Figure 2.7). *Stratocumulus*, *altocumulus* and *cirrocumulus* are small, puffy clouds that form in groups and display a regular pattern, causing the "mackerel skies" (Figures 2.4(a) – 2.4(b)) or "seas of clouds" (Figure 2.4(c)). *Cirrus* clouds have a thin, wispy shape while *cirrostratus* have softer edges (Figure 2.12(c)). Clouds generally have a rather flat base. Figure 2.3 summarizes the main types of clouds.

(a) Altocumulus clouds ("mackerel sky") viewed from below. Note the repetitive pattern in the shape, the general layer shape and the high contrast between thin and thick parts.

(b) Altocumulus clouds("mackerel sky") viewed from below.

(c) Stratocumulus clouds ("sea of clouds") viewed from above. Note how the clouds are brighter in the direction of the sun.



(d) Stratocumulus clouds ("sea of clouds") viewed from above at night. The clouds are lit by the night sky and the moon, giving them a dark blue tint. The city lights shine on the clouds from underneath, giving an orange glow to the clouds.

(e) Cumulus humilis ("fair-weather" cumulus) clouds. Note the typical flat and dark base in contrast with the very bright and sharp cloud top.

(f) Cumulus humilis ("fair-weather" cumulus) clouds.

Figure 2.4: *Various real clouds*

While this thesis focuses on light transport and not cloud shape modeling, it is important to note that the edges and outlines of a cloud are a strong visual feature of clouds. First, this is important because the application of our light transport model to just any shape will not by itself make that shape look like a cloud. Some of the visual features of a cloud are bound to its shape. Thus, we have to apply our shading algorithms on shapes that should mimic these most important features, *i.e.,* puffy, wispy or soft edges. Second, this information tells us where we should emphasize the quality of our rendering. Since edges are a key element in clouds, it is more important to reproduce well their color and contrast than that of flat-tint cloud parts.

2.2.2 What is the color of a cloud ?

2.2.2.1 What lights a cloud

"Sky is the source of all light."
– John Constable.

A cloud does not emit light, it only reflects the incident light. Since it does not absorb any part of the visible spectrum, its colors are those of the light it receives, which is not only the sunlight, as we will see in this section.

Sunlight The majority of light that a cloud receives during the day comes from the sun. As a result, the color of a cloud is mainly that of the sun. That is why we see white clouds at noon and red clouds at twilight.

Skylight However, the sun is not the only source of light for a cloud. As we have seen in §1.5.12, the illuminance on a cloud due to the sky is at least a fourth of that of the sun and becomes much more important as the sun sets. As a result, the sky is sometimes the *main* source of light for a cloud, hence the colors of a cloud are also composed of the colors of the sky (see Figure 2.5(a)). Since the sky is a very broad source, the colors it gives to the clouds are softer than the highly contrasted sun colors. As a result, it does not strike the eye that the sky is shining on the clouds when we look at them. However, a human will definitely think that something is wrong (without necessarily knowing what exactly is missing) when looking at a picture of clouds generated by a method that does not account for skylight.

Groundlight Like the sky, the ground is a significant light source for the clouds, as we have seen in §1.5.12, especially when its albedo is high. As a result, the color of a cloud is not just plain white, but also shows tints of blue, green or brown, as seen on Figures 2.5(b) and 2.11(a).

Even though we enhanced the saturation of the picture to better show the effect of the groundlight on the clouds, its importance is not negligible [NND96]. Indeed, clouds appear unrealistic when these effects are not reproduced in computer generated images, as we will show in our results in Chapter 6 and 7.

In addition to reflecting the sky and sun light, the ground also reflects the light coming from the bottom of the clouds. In return, the clouds reflect some of the ground light, and so forth. In some cases (*e.g.,* the clouds are thick and the ground has a high albedo), these inter-reflections are far from negligible and cause the *ice blink* or *water sky* effects described on Figures 2.6(d) and 2.6(c).

ice blink

water sky

(a) Cumulus congestus cloud at altitude. The sky is an important light source for this cloud.

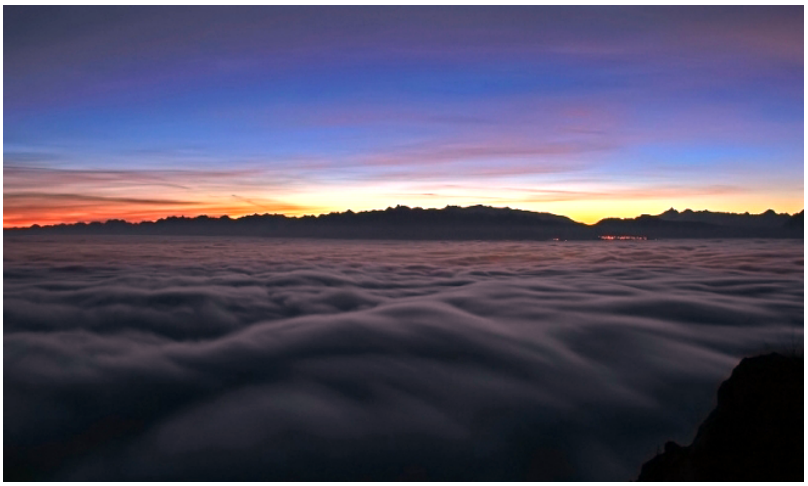
(b) Cumulus humilis clouds above meadows. The grassy ground reflects the sunlight and shines on the clouds, giving their base green tints.

Figure 2.5: Left: *photographs of clouds.* Right: *same photographs with enhanced saturation of colors. Note how the clouds show blue and green tints.*

Other clouds A cloud is usually not only surrounded by the sky and the ground, but also by other clouds. Since clouds can be very bright, it is certain that they act as light sources on each other and that complex inter-reflections happen between them, as for the clouds-ice inter-reflections. However, we do not know of any data or study evaluating the importance of this phenomenon. We believe it is the same phenomenon, arising at lower scales between neighboring parts of clouds and neighboring lobes, that yields the bright creases of clouds (see §2.2.2.2).

Night time At night the moon takes the role of the sun, when it is visible. Because it is much less bright, the luminance of the clouds is mainly in the scotopic range. The human eye does not saturate when looking at the clouds and it sees the dimmer parts of the clouds as black. Lights from urban areas are also an important light source (see Figures 2.11(c), 2.6(a) and 2.6(b)).

(a) Paris by night. The Eiffel Tower in the background illuminates the clouds above it (left part of the image).



(b) Clouds above Grenoble at night. The clouds have an orange tint above the city.

(c) Ice blink

(d) Water sky

Figure 2.6: *Effects of the ground on the colors of a cloud.* Top: *Urban areas shine on the clouds, giving them an orange color. It is even possible to locate a city from far away by locating its glow on the clouds.* Bottom: *Contrary to water, ice has a high albedo, reflecting much of the light it receives back to the clouds above it. These dark zones ("water sky") or bright glares ("ice blink") on the underside of clouds can assist travelers in navigating the ice of the polar seas by helping them detect ice or water beyond the horizon [Nat].*

2.2.2.2 Cloud-specific visual features

View-independent visual features When looking at a smaller scale than the whole cloud itself, we can identify a list of important visual features. The most evident is that clouds are brighter on their lit side and darker on their unlit side (see Figures 2.4(e), 2.7). We can also see that light variations show a lot more high frequencies in lit parts while almost not in shadowed areas: clouds appear detailed and contrasted on the silhouettes and lit sides and show softer, flat tints in the unlit parts (see Figures 2.12(d), 2.7).

When considering the lit side of a cloud, we observe that its creases are brighter than the rest of the cloud. This phenomenon happens at multiple scales: broad creases are bright, smaller creases inside these broad creases are even brighter (see Figures 2.11(a), 2.11(b), 2.12(b)). We believe this is due

Figure 2.7: *A forming cumulonimbus ("thunderhead"). Note the strong contrast between the bright and crisp top and the dark and soft base. This contrast is also present in smaller clouds.*

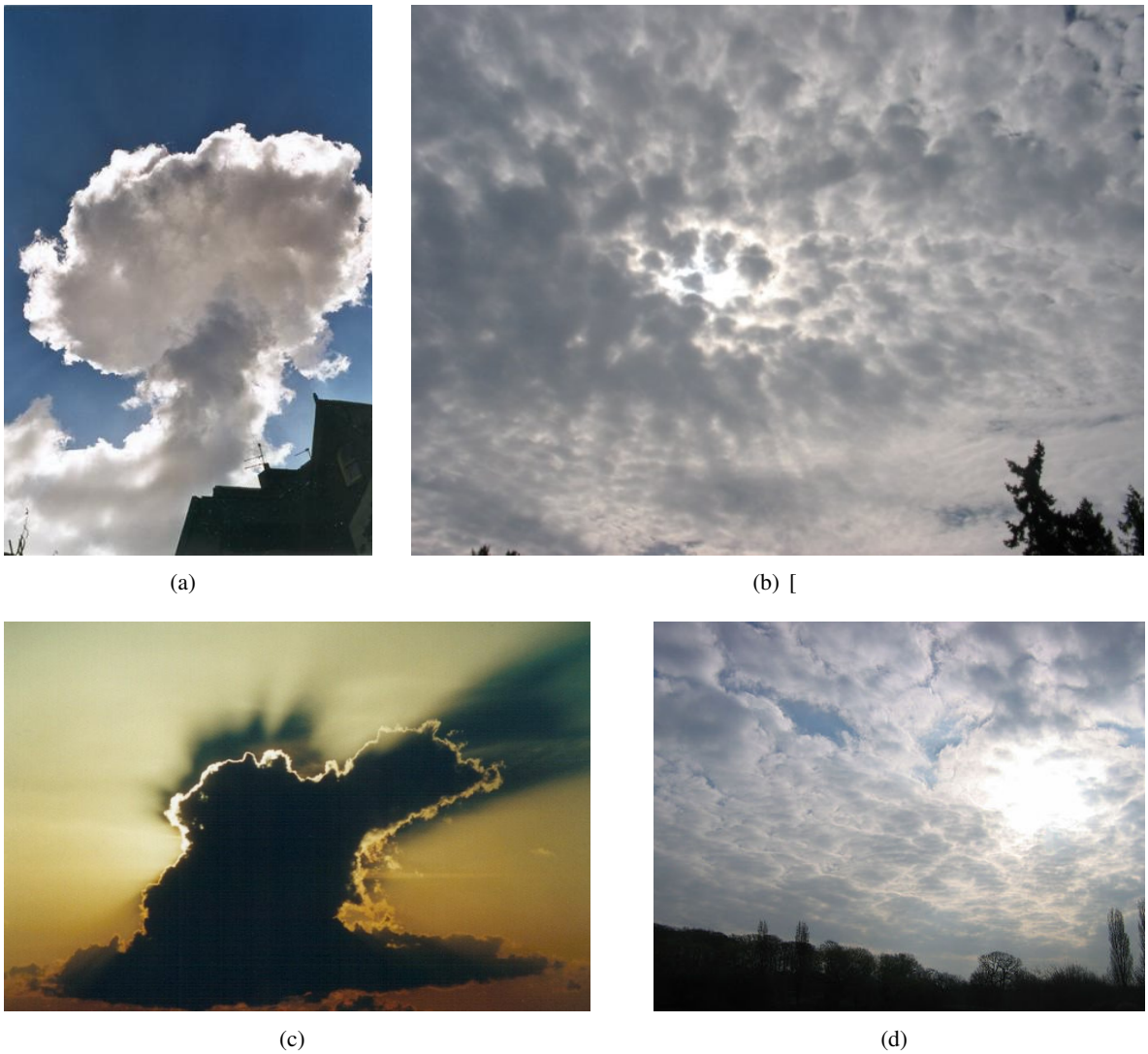
to inter-reflections between different parts of the cloud. These inter-reflections arise also at an even higher scale: a cloud can serve as a light source for a neighboring one.

View-dependent visual features Clouds also display view-dependent visual features. That is, the same part of a cloud will appear differently depending on where it is viewed from. When considering the silhouettes of a cloud, we notice that the silhouette is much brighter than the core when the sun is behind the cloud (see Figure 2.12(d) and 2.8). We call this bright edges effect the "*silver lining*". On the other hand, the edges of a cloud look darker when the sun is behind the observer (see Figures 2.11(b), 2.12(a) and 2.12(b)). This phenomenon also appears in thin parts of clouds and on smaller clouds. It is a very important visual feature of clouds. This phenomenon gives details to the cloud, making it look contrasted instead of just plain white everywhere. Moreover, it outlines the contour and edges of the clouds, making them striking to the human eye. This is why the aspect of the edges are so important in the look of a cloud.

silver lining

Another of these view-dependent effects is what we name the *"pseudo-specular" effect*. When looking at a sea of clouds from above, an observer can see the clouds are brighter in the sun direction. The sea of clouds seems to reflect the sunlight off like a specular surface (*e.g.,* the ocean) would do (see Figure 2.9). Actually, this reflection has a much wider solid angle, and its direction is actually not quite specular. We investigate this effect in Chapter 4.

pseudo-specular effect



(a) (b) [

(c) (d)

Figure 2.8: *The "silver lining" effect: the edges and thin parts of clouds are much brighter than their core when looking towards the sun.*

glory

fogbow

antisolar point

Glory and fogbow Clouds also display specific visual phenomena called *glory* and *fogbow*. The glory is visible at the *antisolar point* (*i.e.,* in the direction opposite to that of the sun). It is a small rainbow-colored circle centered around the antisolar point with an angular size of about 2˚(see Figure 2.10(a)). The fogbow is a colorless arc or circle, about the size of a rainbow, *i.e.,* 40˚around the antisolar point (see Figure 2.10(b)). It is also named "white rainbow", "cloudbow", or "sea dog".

Inside a cloud A cloud as seen from the inside is much more simple. It is of a uniform gray-to-white color, more or less bright depending on whether the observer is closer to the top or to the base of the cloud. It is also brighter when looking up than when looking down when the observer is close to the cloud top. When the viewer is deep inside the cloud, the color is uniform in all directions. According to paragliders, in these conditions there is no way to tell where the vertical direction is.

(a) (b)

Figure 2.9: *The pseudo-specular effect of stratiform clouds. When seen from above, these clouds look much brighter in the direction of the sun, as if they were reflecting the incident light like a specular surface.*

(a) Glory (b) Fogbow and glory

Figure 2.10: *Glory and fogbow.*

(a) Cumulus congestus cloud about to turn into a cumulonimbus. Note how the bottom displays red tints (due to ground illumination) while the top is more white. Bright creases and dark edges are clearly visible.

(b) Cumulus congestus. The bright creases and dark edges are clearly visible.

(c) A cloud at night. The thick parts reflect the orange light coming from the city. The moon shines through the edges that hide it. The rest of the edges have dark blue colors, from the night sky illumination.

Figure 2.11: *Various real clouds*

(a) Cumulus congestus cloud. Note the very bright overall look and the darker, sharp, fluffy edges.

(b) Cumulus congestus cloud. Note the very bright overall look, the darker, sharp, fluffy edges and the even brighter creases.

(c) Cirrus cloud. Note the curly and wispy appearance.

(d) Altostratus cloud. Note the silver lining and very dark base.

Figure 2.12: *Various real clouds*

2.2.2.3 Atmospheric phenomena

The atmosphere surrounding the clouds also affect its apparent color. The atmosphere makes distant objects appear paler and bluer than closer objects. As a result, distant clouds look more blue and less contrasted than close clouds (see Figure 2.13). This phenomenon is called *aerial perspective*.

aerial perspective

Another atmospheric phenomenon associated with clouds is the shafts of light, also called "god rays" or "crepuscular rays". They can be seen in more or less hazy conditions when the sun is behind the clouds. They are caused by the shadows that the clouds cast on the atmosphere (see Figure 2.8(c)).

Finally, atmospheric conditions in general also affect the color the clouds receive since the sun color (red at sunset, yellow in hazy conditions, etc.) and the sky color (blue from the scattering of light by atmospheric particles) are determined by the composition of the atmosphere.

Note that some atmospheric phenomena commonly credited to clouds are actually caused by other sources. Rainbows are the result of scattering of light by raindrops. Halos (or "icebows") and parhe-

Figure 2.13: *Aerial perspective. The atmosphere makes distant objects look less contrasted and blueish.*

lias (or "sun dogs") are caused by ice crystals suspended in the air or in ice clouds. Since our goal is to render water clouds, these effects are not part of our goals.

Summary of §2.2 : We have identified the important aspects of the appearance of a cloud from a human perception point of view. To ensure realism, we should reproduce these visual features, *i.e.,*

- Take into account the *full* light environment of a cloud, *i.e.,* not only the sun but also sky and ground colors.
- Reproduce the most important aspects of the shape of the cloud, *i.e.,* the aspect of its edges (whereas the exact content of the core is less important).
- Reproduce the visual features inherent to clouds: bright, contrasted, detailed lit side, dark, soft unlit side, bright and contrasted edges when seen from below, dark edges when seen from above, bright creases, pseudo-specular effect, glory, fogbow, shafts of light, aerial perspective.
- Display the clouds in a decent environment (*i.e.,* within a realistic sky and landscape)

Nous avons identifié les aspects importants de l'apparence d'un nuage du point de vue de la perception humaine. Pour être réaliste, nos méthodes doivent reproduire ces éléments visuels, i.e.,

- *Prendre en considération l'environnement lumineux* complet *d'un nuage, i.e., non seulement le soleil mais également le ciel et le sol.*

- *Reproduire les aspect les plus importants de la forme des nuages, i.e., l'aspect de ses contours (tandis que le contenu exact de son coeur est moins important).*
- *Reproduire les caractéristiques visuelles inhérentes aux nuages: un côté éclairé détaillé, contrasté et lumineux, un côté non éclairé sombre et doux, des bords lumineux et contrasté lorsqu'ils sont vus du dessous et sombres lorsqu'ils sont vus du dessus, des creux lumineux, l'effet pseudo-spéculaire, la gloire, l'arc-en-nuage, les cônes de lumière, la perspective atmosphérique.*
- *Afficher les nuages dans un environnement décent (i.e., au sein par un paysage et un ciel réaliste).*

City Clouds, *Bruce Cody, oil.* Even on an overcast day, the sky is still the source of light. With the emphasis on cool blue grays, this light has little warmth in it.

Figure 2.14: *Excerpt from [Sel92]. Note that the grays are actually blue.*

2.3 The artistic point of view

Artists depict the world and reproduce the features that are aesthetically and visually important. Taking a look at what they draw and how they draw it can give us hints about what are the important visual features from a human's point of view. Some drawings and paintings are included in this document to illustrate how artists handle clouds (see pages 12–14, 35–37, 104–107, 180, 201–202, 244–247).

Here are a few excerpts from a book explaining how to paint clouds [Sel92].

- On the difference between lit and unlit parts:
  - "*Even where clouds obscure the sun, light still shines through them*".
  - "*Note how crisp the highlight top edge is compared to the shader underside*".
  - "*Sunlit tops of the clouds have a hard edge while those parts in shadow have a softer outline*".

  This supports our observations that there is a strong difference of aspect between the crisp top and the soft base of clouds (see Figure 2.16).

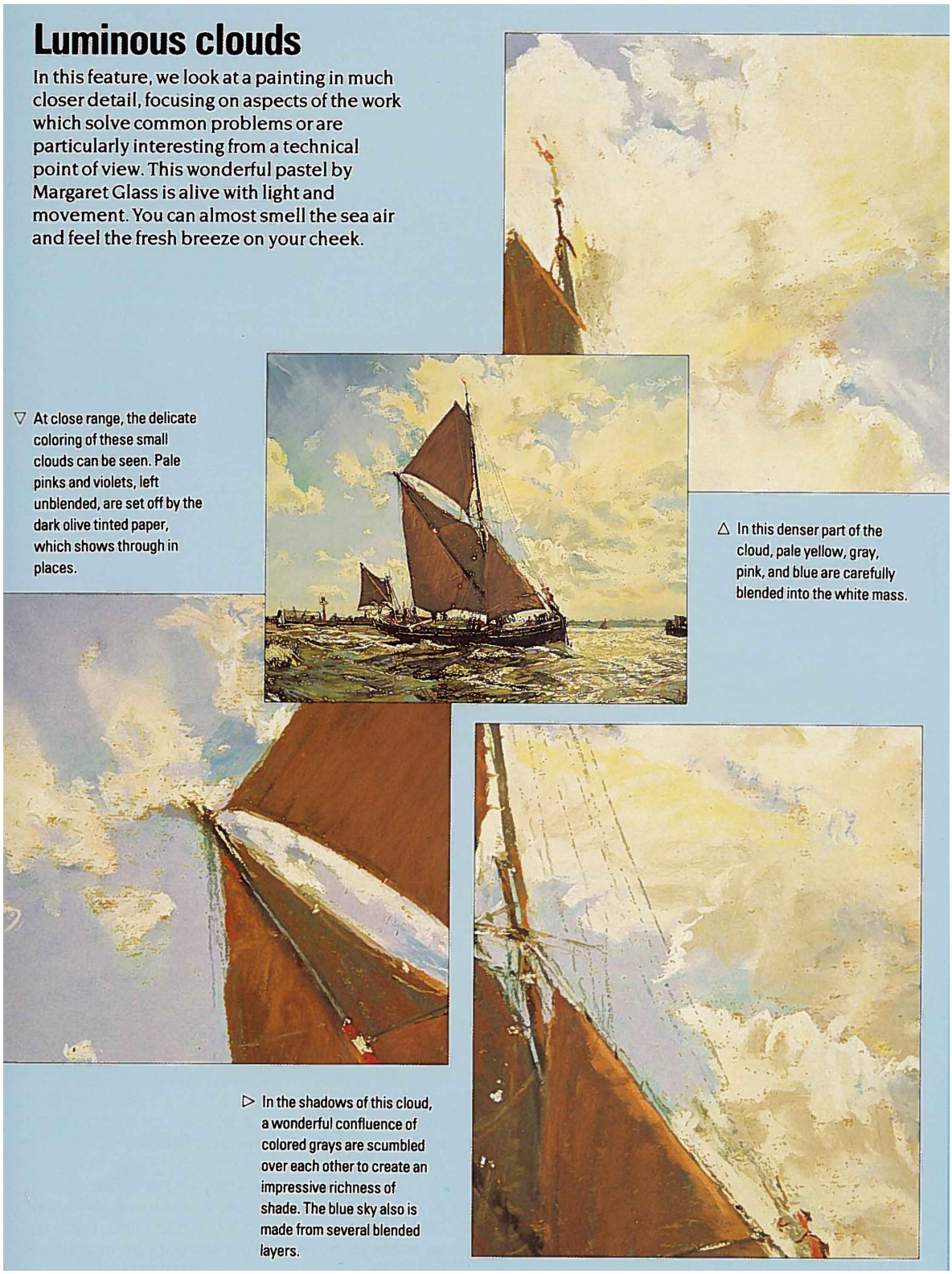


Figure 2.15: *Excerpt from [Sel92] pointing out the various colors used to paint clouds.*

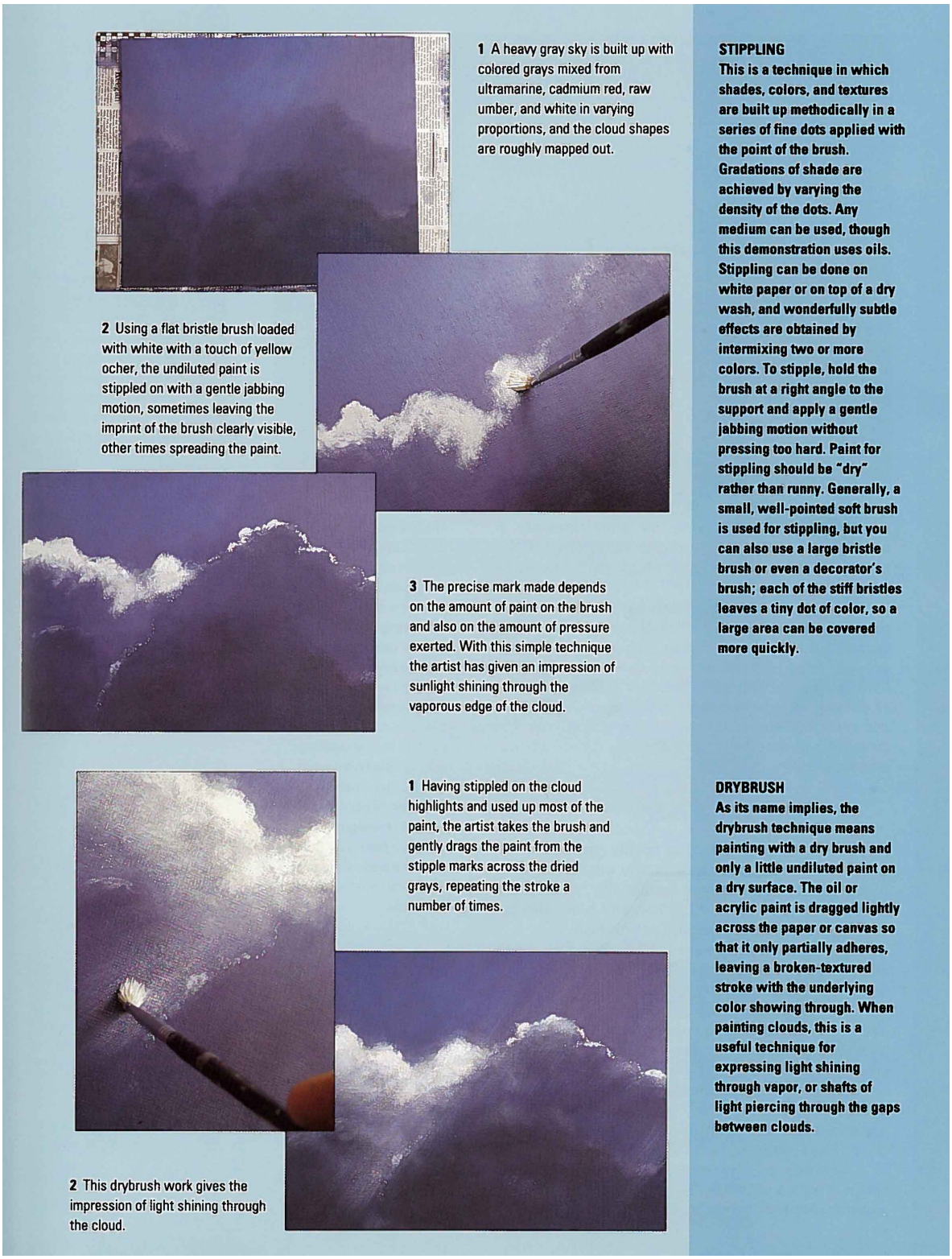


Figure 2.16: *Excerpt from [Sel92] showing how detailed the silver lining is compared to the shadowed parts. Note also the strong contrast.*

- On the aspect of the shape: "*Each cloud is different, made up of an unpredictable, uneven mass of crags and eddies which reflect the light in different strength*". This underlines chaotic and fractal nature of cloud shapes.
- On their colors
  - "*It is easy to think of clouds as simply gray and white, but if you look at them carefully, you will be surprised by the variety of colors they contain: the 'white' areas may contain hints of pink, yellow, and blue, and the gray areas may appear purplish or brownish*".
  - "*Highlights are yellow, reflecting the Sun, and the shadows underneath have a blueish tinge, reflecting the sky*".

  This is a perfect illustration of the fact that the sky and the ground are significant light sources for clouds (see Figures 2.15 – 2.16).
- On atmospheric phenomena: "*Atmospheric haze causes the colors of the clouds to appear cooler, grayer, and less distinct as they recede from our view – hence the term 'atmospheric perspective.*"'. This emphasizes the role of aerial perspective.

Summary of §2.3 : Artists also provide us with some hints about what is important in a cloud from a visual and aesthetic point of view:

- Edges, wisps and shadows are the most important.
- Colors range from dark blue to bright yellow, through pinks and browns. They are not just white, gray or black.
- A cloud should blend with its surroundings.

This comforts us in the analysis we did in §2.2.

Les artistes nous fournissent également des indices sur ce qui est important dans un nuage d'un point de vue visuel et esthétique:

- *Les contours, les barbules et les ombres sont les plus importants.*
- *Les couleurs vont du bleu sombre au jaune lumineux, en passant par le rose et le marron. Elles ne sont pas simplement blanches, grises ou noires.*
- *Un nuage doit se marier avec son environnement.*

Ceci nous conforte dans l'analyse que nous avons faite en §2.2.

2.4 The atmospheric physics point of view

2.4.1 Cloud formation

The way clouds are formed is described by the laws of fluid dynamics and thermodynamics, and can be summarized as follows. The sun heats the ground and sea by radiation. Hot air containing water in form of gas (vapor) rises. As it goes up, the pressure lowers and the air parcel expands, which makes

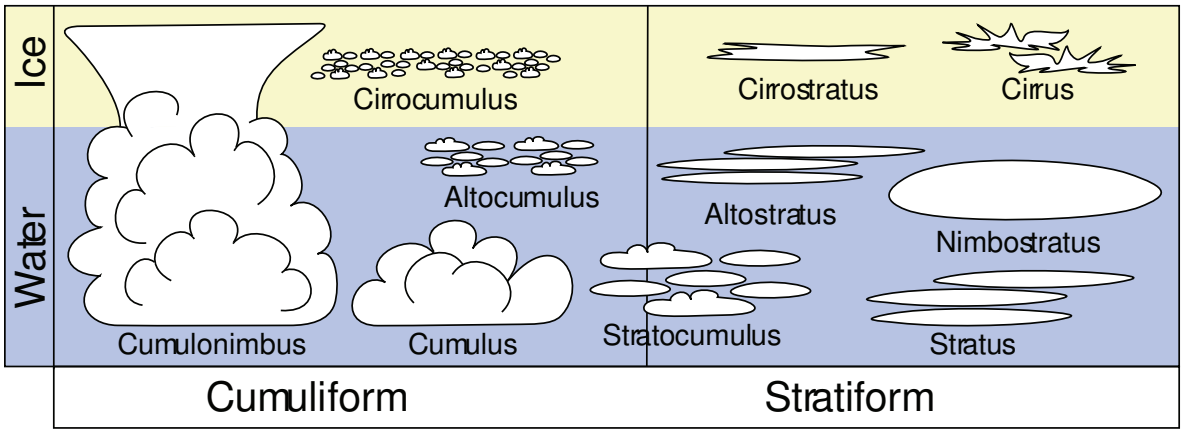


Figure 2.17: *Common cloud types.*

it cool down. If it contains enough vapor, there is one point (the "dew point") at which the temperature decrease makes this vapor condense into spherical water droplets or ice crystals[1]. It is these droplets and crystals that we see and that form the clouds [Cha02]. Note that these droplets are *not* raindrops[2]. Droplets are much smaller than raindrops. The radius of a cloud droplet equals several micrometers, while the radius of a raindrop equals several millimeters. That is, it takes about *one million* droplets to create one raindrop.

Depending on the weather conditions, these clouds can take different shapes, as we have seen in §2.2.1. The main types of clouds are summarized on Figure 2.17. Convective currents create the puffy, heaped, billowing *cumuliform* clouds (*cumulus humilis*, *altocumulus*, *cumulonimbus*, etc.). In the absence of such convective currents, clouds form in layers. These kinds of clouds are called *stratiform* clouds (*stratus*, *nimbostratus*, etc.). In some conditions, clouds can form at ground level, in which case we call them fog. The highest clouds are made of ice particles instead of water droplets and are called *cirrus*, *cirrocumulus* or *cirrostratus*.

It is common to hear that a cloud weighs several million tons. While this is true, the weight of a cloud (i.e. of the air+condensed water mass) is roughly the same as the equivalent volume of clear air. Their densities are almost equal. Thus, it just doesn't tell us anything. It is hot air !

cumuliform

stratiform

What the eye sees when looking at a cloud is the light scattered by water droplets or ice crystals in the direction of the eye. We describe this scattering phenomenon in §2.5. It depends heavily on the shape and distribution of these droplets or ice crystals. In the case of ice crystals it also depends on their spatial orientation. Since our main goal is to reproduce cumulus-type clouds, which are made of droplets, this thesis will not address the specifics of ice clouds.

[1] This dew point corresponds to the altitude of the base of clouds.

[2] Raindrops are created by the coalescence of droplets, which happens in specific conditions, inside the core of rain clouds (*e.g.,* cumulonimbus), in a manner that is not yet fully understood.

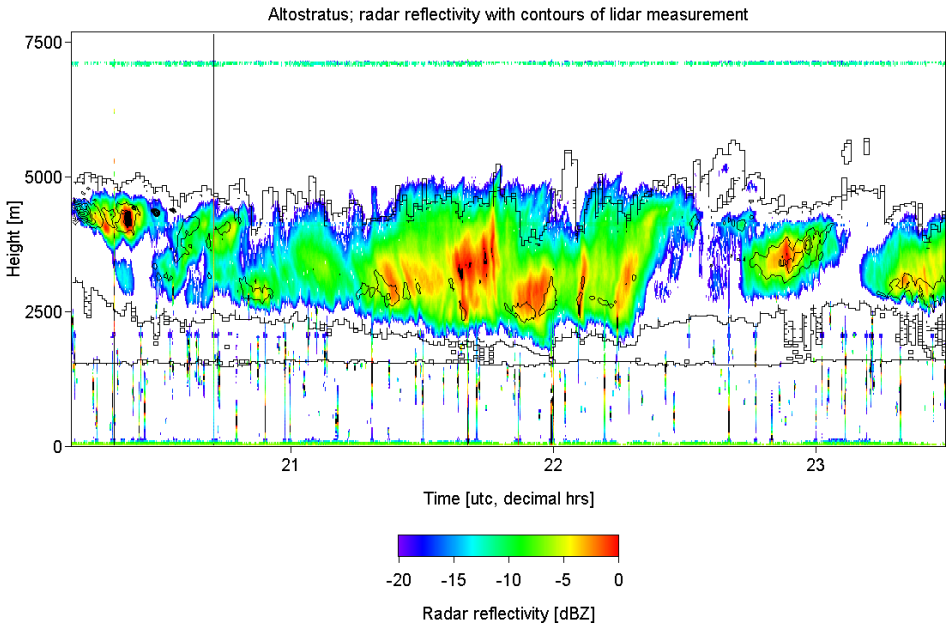


Figure 2.18: *Observation of clouds using radar and lidar [RVL97].*

2.4.2 Cloud contents

liquid water content

Several key values are used to describe the contents of a cloud. The *liquid water content (LWC)* (LWC) $L_0(\boldsymbol{p})$, expressed in g.m^{-3}, is the mass of water per unit of volume at a point $\boldsymbol{p}$. Usual values for water clouds range between 0.1 g.m^{-3} and 1 g.m^{-3}. The *droplet concentration* $\rho(\boldsymbol{p})$, expressed in m^{-3}, is the

droplet concentration

number of droplets per unit of volume at a point $\boldsymbol{p}$. It varies between 50 cm^{-3} and 1000 cm^{-3} [Mas71]. It may be retrieved by measurements from radar, satellite and lidar[3] (see Figure 2.18), although these measurements are currently not precise enough for CG applications.

If all droplets in a cloud were to have the same size, then the radius r of the droplets could simply be determined by $r = \sqrt[3]{\frac{3}{4\pi}\frac{L_0}{N}}$. However, a cloud contains droplets of varying sizes. The mechanism that determines the size of the droplets and their evolution is not fully understood. However, there is experimental data describing the statistics of cloud droplets [Mas71, Dra88]. These statistics are described

droplet size distribution

through a *droplet size distribution (DSD)* (DSD). The DSD $N(\boldsymbol{p}, r)$ represents the concentration of droplets with respect to droplet radius. It is often expressed as

$$N(\boldsymbol{p}, r) = \rho(\boldsymbol{p}).n(r)$$

normalized droplet size distribution

with $n(\boldsymbol{p}, r)$ the *normalized droplet size distribution (nDSD)* (nDSD) (*i.e.*, $\int n(r)dr = 1$). Figure 2.19 shows an example of such a DSD. For non-raining clouds, the DSD usually resembles a lognormal distribution with a single identifiable peak.

mean radius

Several key values can be extracted from the DSD. The *mean radius*

$$r_m = \int_0^\infty r.n(r)dr \tag{2.1}$$

[3] A LIDAR is a type of radar using light instead of radio waves.

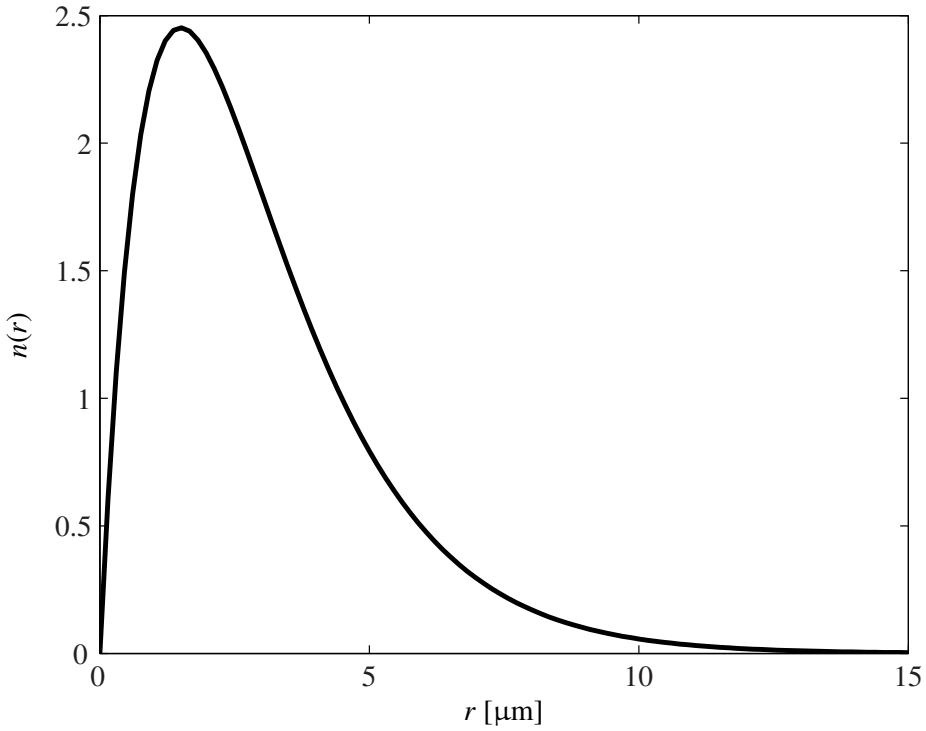


Figure 2.19: *Droplet size distribution of a typical cumulus cloud, generated using the modified gamma distribution formula.*

gives a rough approximation of the droplet size. Due to their characteristic shape, DSDs have been modeled by classical probability distribution functions such as the gaussian distribution [BH98] or the lognormal distribution [PK97]. In this thesis we use the *modified gamma distribution* [Lev58, Dei69, Cla74, PK97], which has been shown to model fairly well the DSD observed in clouds, defined by

modified gamma distribution

$$n(r) = \frac{1}{\Gamma(\gamma) r_n} \left(\frac{r}{r_n}\right)^{\gamma-1} e^{-\frac{r}{r_n}}, \tag{2.2}$$

with $\Gamma(x)$ the gamma function (see Figure 2.19). This distribution is characterized by two parameters: r_n is called the *characteristic radius* of the distribution and γ represents its broadness. For this distribution, the mean radius r_m can be simply computed as $r_m = r_n \gamma$ and the standard deviation is exactly $r_n \sqrt{\gamma}$. We can thus describe the DSD of a cloud compactly via the three parameters N_0, r_n and γ. As we will see in §2.5, the DSD has a huge impact on the visual appearance of clouds.

characteristic radius

All of the variables introduced in this section (*e.g.,* L_0, ρ, etc.) are defined for a given volume of cloud. As a result, they can be defined for an infinitesimal volume and vary through space, or be defined as constant throughout a whole cloud, depending on the uses. Since measuring a DSD across a whole volume of cloud is very difficult, it is common to see only ρ spatially varying while $n(r)$ is assumed constant within the whole cloud volume. Usual values for r_m are between 2 µm and 8 µm. Usual values for γ range rom 1 to 20. N_0 goes from $0\,\mathrm{cm}^{-3}$ (clear air) to $1500\,\mathrm{cm}^{-3}$ (very dense cumulonimbus core). The size of the cloud itself can go up to several kilometers in each direction. A cumulonimbus cloud can have a vertical height of 15 km – 18 km.

Summary of §2.4 : A cloud consists of millions of microscopic water droplets spanning several km. The concentration (or density) and size distribution (DSD) of these droplets are complex and the mechanisms that govern them are poorly understood. It is common for physicists to rely on approximations deduced from measurements to describe the DSD.

Un nuage consiste de millions de gouttelettes d'eau microscopique s'étendant sur plusieurs kilomètres. La concentration (ou densité) et la distribution de tailles (DSD) de ces gouttelettes sont complexes et les mécanismes qui les gouvernent sont encore mal compris. Il est classique chez les physiciens de se reposer sur des approximations déduites de mesures afin de décrire la DSD.

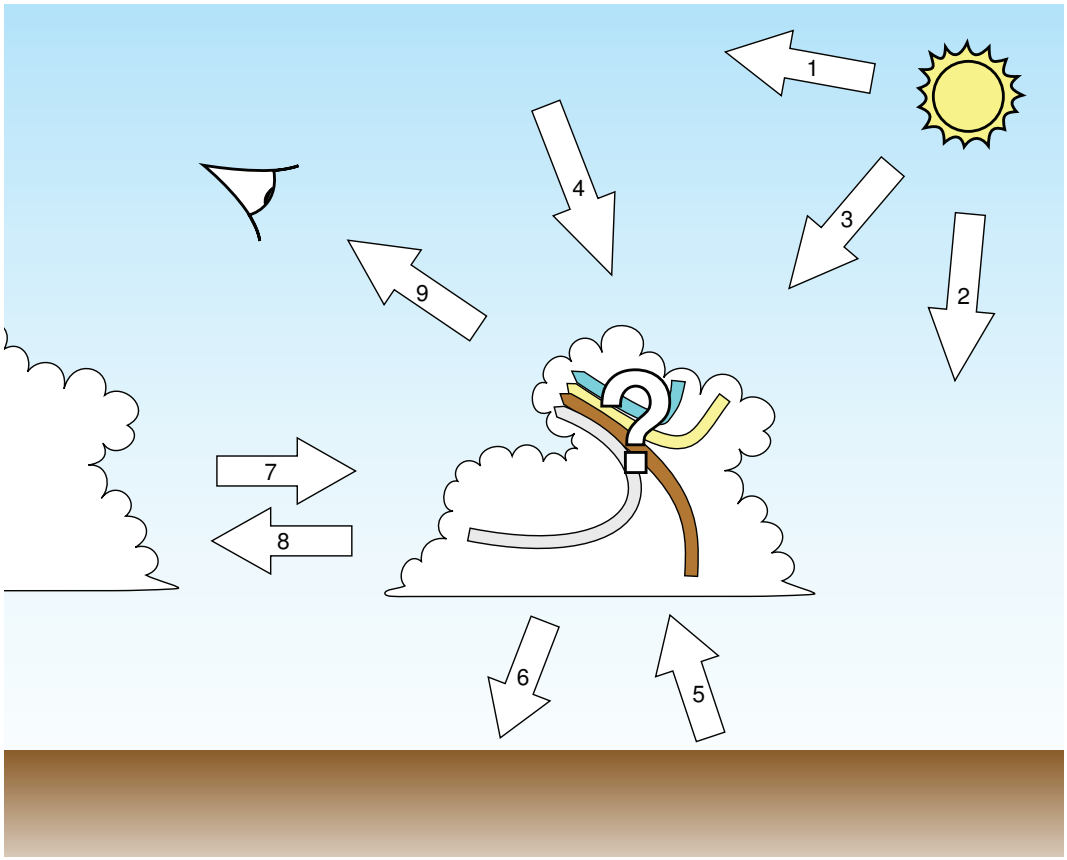


Figure 2.20: *The big picture. The sun lights the atmosphere (1), the ground (2), and the cloud (3). The cloud is also lit by other sources such as the sky (4), the ground (5) or other clouds (7). Inter-reflections arise between the ground and the cloud (5&6) and between neighboring clouds (7&8). All this incoming light is transported in a complex way inside the cloud before finally exiting the cloud and reaching the viewer through the atmosphere (9). The question we need to solve is: how is this light transported ?*

2.5 The radiative transfer point of view

The way light interacts with a cloud (see Figure 2.20) is described by two complementary theories. The Mie theory [Mie08] describes the way light interacts with a single droplet. The radiative transfer theory [Cha60] describes how light interacts at a higher scale with a medium composed of many particles such as droplets. The physics of light transport in clouds are complex enough to fill a few good books [BC06, Cha60, BH98, Dei69, vdH81, Len85, LL95, Min54]. This section will only summarize the notions important to understanding the rest of the thesis. The reader is referred to these books for in-depth information.

When a photon hits a droplet it is scattered in a complex manner described by the Mie theory which we describe in §2.5.1. Since a cloud contains millions of droplets, the chances are that a photon will be scattered multiple times before exiting the cloud. This phenomenon is called multiple scattering and we give an overview of it in §2.5.3. It is formally described by the laws of radiative transfer which we describe in §2.5.4. In §2.5.5 we show that these complex laws admit simpler formulations and sometimes analytical solutions when applied to very simple shapes such as slabs.

2.5.1 Phase function and Mie scattering

2.5.1.1 Phase function

As we wrote in §1.5.2, when a photon hits a cloud droplet the droplet does not absorb it but scatters it. The way light is scattered by cloud droplets is complex. For a photon hitting a droplet from incident direction $\vec{s}$, the *phase function* $P(\vec{s}, \vec{s}')$ describes the probability of this photon to be deviated into direction $\vec{s}'$ upon scattering. The phase function is normalized so that $\int_{4\pi} P(\vec{s}, \vec{s}')d\vec{s}' = 1$. In the case of droplets, which are spherically symmetric, this phase function is axially symmetric and thus depends only on the *phase angle* Θ (also called *scattering angle*) defined by $\cos\Theta = \vec{s} \cdot \vec{s}'$. We thus write it $P(\Theta)$.

phase function

phase angle

scattering angle

2.5.1.2 Mie scattering

Geometrical optics cannot be used to compute the phase function of cloud droplets because the wavelength of visible light ($\lambda \approx 0.5\,\mu m$) is not negligible compared to the radius of the droplets ($1\,\mu m \leq r \leq 15\,\mu m$). . To compute this phase function it is necessary to use Mie scattering [Mie08, BH98] which is a numerical solution of Maxwell's equations for the scattering of electromagnetic radiation by spherical particles. This phase function depends on many parameters, such as the air temperature, water temperature, light wavelength λ and droplet radius r (among others). As a result, each droplet has a different phase function. Figure 2.21 shows an example of this phase function. In practice, the light wavelength and radius are the most significant parameters for clouds droplets. We thus consider the other parameters as constant throughout the clouds.

2.5.1.3 Effective phase function

When the human eye looks at a cloud it does not see each droplet independently but rather a fuzzy, uniform veil like smoke because the droplets are so small and numerous that the eye cannot distinguish them. Thus, what a human sees is not the result of scattering by one single droplet but by a

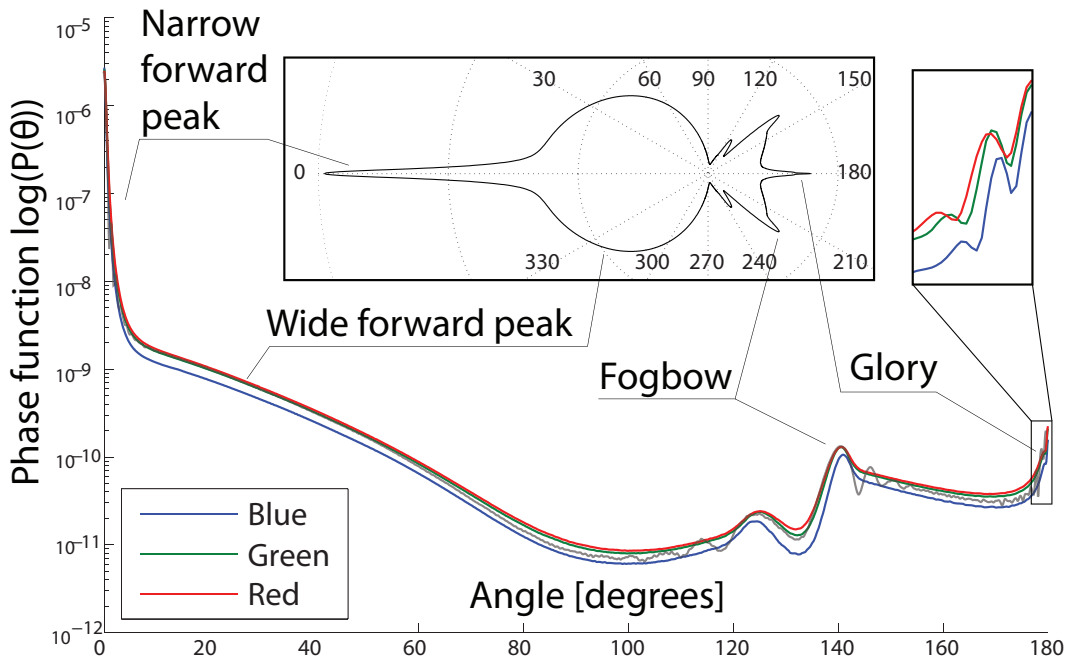


Figure 2.21: *Logarithmic plots of various Mie phase functions (inset: polar logarithmic plot).* In gray: *Mie phase function for the whole visible spectrum and a droplet of 20* µm *radius. Note the small ripples.* In RGB: *Effective Mie phase function for the three red, green and blue channels for the DSD of Figure 2.19. Note how the ripples have disappeared.*

collection of droplets. As a result, what we really are interested in is not the phase function of a single given droplet but rather the phase function of a given volume of droplets. We can compute such an *effective effective phase function* using the DSD of the given volume:

effective phase function

$$P(\Theta,\lambda) = \int_0^\infty P(\Theta,\lambda,r)n(r)dr. \tag{2.3}$$

If the normalized DSD $n(r)$ is assumed constant throughout a whole cloud (which is usually the case), this unique average effective phase function can be used for any droplet in the cloud.

The Mie phase function is highly wavelength-dependent and it is important to take it into account for CG application. Indeed, it is this wavelength-dependency that causes the colors of the glory. As we wrote in §1.5.1, it is common in computer graphics to account for wavelength dependence by considering three different channels (red, green and blue). This gives

$$\begin{aligned} P_R(\Theta) &= \int \nu_R(\lambda)P(\Theta,\lambda)d\lambda, \\ P_G(\Theta) &= \int \nu_G(\lambda)P(\Theta,\lambda)d\lambda, \\ P_B(\Theta) &= \int \nu_B(\lambda)P(\Theta,\lambda)d\lambda, \end{aligned}$$

where P_R, P_G and P_B are the phase functions for each of the three channels. Note that this is an approximation and can cause color shifts in the generated images. However, since a cloud is mainly white (*i.e.,* it scatters all wavelengths almost equally), this should not pose strong issues.

Figure 2.21 shows such an RGB effective phase function for a given DSD. We can see that these integrations have removed the highest-frequency ripples (which are effectively not visible in reality) and smoothed the overall phase function. The end result is very close to the phase functions that are measured empirically from real clouds. .

Note that the Mie phase function can only be computed numerically and this computation is rather time-consuming. As an example, it took us 8 hours to compute the RGB phase function shown on Figure 2.21 using MiePlot [Lav], a tool to easily compute Mie scattering made available by researchers in atmospheric optics[4].

2.5.1.4 Features of the phase function

The effective phase function shows distinctive features inherent to cloud droplets. First of all, it is highly anisotropic (note that the y axis of Figure 2.21 is on a logarithmic scale). The *mean cosine* or *asymmetry parameter* g defined by

mean cosine

asymmetry parameter

$$\begin{aligned} g &= \int_{4\pi} P(\vec{s},\vec{s}')(\vec{s}\cdot\vec{s}')d\vec{s}' \\ &= 2\pi\int_{-\pi}^{\pi} P(\Theta)\cos\Theta\sin\Theta d\Theta \end{aligned} \tag{2.4}$$

gives an idea of the anisotropy of a phase function. A value of $g = 0$ corresponds to an isotropic phase function, while values of $g = -1$ or $g = 1$ correspond to total backward or total forward scattering. In the case of cloud droplets, $g \simeq .9$, which means that they are highly anisotropically forward scattering. Indeed, about 99% of the incoming light is scattered in the forward direction.

This forward anisotropy can be split into two components (see Figure 2.21):

- **Narrow forward peak** There is a strong peak in the forward direction. It corresponds to $\approx 5^\circ$concentrating $\approx 50\%$ of the function.
- **Wide forward lobe** This large lobe represents $\approx 49\%$ of the scattering.

The backward part of the phase function shows complex waves and peaks (see Figure 2.21). Since this part of the phase function represents only $\approx$ 1% of the scattering, these waves have almost no visual effect. Only two of them have visual consequences:

- **140°backward peak** This peak is responsible for the fogbow we described in §2.2.2.2.
- **Backward peak** This 2°-wide peak is responsible for the glory we described in §2.2.2.2.

Depending on the DSD, the glory and fogbow may be more or less visible. The broader the DSD is, the more the DSD will be smoothed and the less strong these peaks will be.

2.5.2 Basic radiative properties

We are considering a cloud of concentration $\rho(\boldsymbol{p})$ and DSD $n(r)$. Let us take the point of view of a photon freshly emitted by the sun. After eight minutes of space travel, it traverses the atmosphere of the earth and finally reaches a cloud. What is the probability of this photon hitting a droplet ?

[4]There may exist more efficient tools in the field of atmospheric physics unknown to the computer graphics community

2.5.2.1 Extinction

extinction cross-section

Case of a single droplet of unique radius Let us call $\sigma_e(r)$ the *extinction cross-section* of a droplet or radius r, *i.e.,* the area normal to the flux that could intercept this flux. It is computed by

$$\sigma_e(r) = 2\pi r^2. \tag{2.5}$$

extinction coefficient

Note that it is not $\sigma_e(r) = \pi r^2$, as one could expect from geometrical optics. A good explanation of this can be found in [BC06]. This cross-section yields the extinction coefficient κ_e defined by

$$\kappa_e(\boldsymbol{p}, r) = \rho(\boldsymbol{p})\sigma_e(r).$$

optical thickness

The extinction coefficient is used to determine the *optical thickness* τ between two points $\boldsymbol{a}$ and $\boldsymbol{b}$ as

$$\tau(\boldsymbol{a}, \boldsymbol{b}, r) = \int_{\boldsymbol{a}}^{\boldsymbol{b}} \kappa_e(\boldsymbol{p}, r) d\boldsymbol{p}.$$

extinction function

For a volume of droplets with concentration $\rho(\boldsymbol{p})$ the probability of the flux traversing this volume along a path of length l from a point $\boldsymbol{a}$ to a point $\boldsymbol{b}$ without hitting a droplet is called the *extinction function* and is defined by

$$\beta(\boldsymbol{a}, \boldsymbol{b}, r) = e^{-\tau(\boldsymbol{a}, \boldsymbol{b}, r)}.$$

extinction cross-section

Case of a DSD A cloud is composed of droplets of various sizes, described by a DSD $n(r)$ (see §2.4.2). For a DSD, the effective extinction cross-section becomes

$$\sigma_e = \int_0^\infty \sigma_e(r) n(r) dr. \tag{2.6}$$

effective radius

It can also be computed through the *effective radius* r_e with

$$\sigma_e = 2\pi r_e^2 \tag{2.7}$$

and

$$r_e = \frac{\int_0^\infty \sigma_e(r) n(r) r dr}{\int_0^\infty \sigma_e(r) n(r) dr}. \tag{2.8}$$

extinction coefficient

Then the effective extinction coefficient is

$$\kappa_e(\boldsymbol{p}) = \rho(\boldsymbol{p})\sigma_e, \tag{2.9}$$

optical thickness

the effective optical thickness is

$$\tau(\boldsymbol{a}, \boldsymbol{b}) = \int_{\boldsymbol{a}}^{\boldsymbol{b}} \kappa_e(\boldsymbol{p}) d\boldsymbol{p} = \int_{\boldsymbol{a}}^{\boldsymbol{b}} \rho(\boldsymbol{p})\sigma_e d\boldsymbol{p}, \tag{2.10}$$

and the effective extinction function is

extinction function

$$\beta(\boldsymbol{a},\boldsymbol{b}) = e^{-\tau(\boldsymbol{a},\boldsymbol{b})} = \exp\left(-\int_{\boldsymbol{a}}^{\boldsymbol{b}} \rho(\boldsymbol{p})\sigma_e d\boldsymbol{p}\right). \tag{2.11}$$

The values $P(\Theta)$ (phase function), r_e (effective radius), σ_e (extinction cross-section), κ_e (extinction coefficient), β (extinction function) and τ (optical thickness), defined by Equations 2.3 – 2.11 are the basic radiative properties of a cloud. All the laws of radiative transfer rely on them, and they indirectly determine the aspect of a cloud. They depend on the contents of a cloud that are $\rho(\boldsymbol{p})$ (droplet concentration) and $n(r)$ (DSD), introduced in §2.4.

Note that since we consider that the DSD does not vary throughout a cloud, the phase function $P(\Theta)$, the extinction cross-section σ_e and the effective radius r_e are constants for a given cloud.

Derived properties The modified Gamma DSD we use has the practical advantage that r_e can be simply computed by $r_e = (\gamma + 2)r_n$. This effective radius r_e has much more meaning for a cloud than the mean radius r_m because it gives a quick idea of its optical properties.

transparency

From these values, we can derive a few other important representative variables. The *transparency* of a path through a volume of cloud from point $\boldsymbol{a}$ to point $\boldsymbol{b}$ is exactly the extinction function β. The *opacity* α of a path through a volume is the complementary, *i.e.,*

opacity

$$\alpha(\boldsymbol{a},\boldsymbol{b}) = 1 - \beta(\boldsymbol{a},\boldsymbol{b}). \tag{2.12}$$

It represents the probability of a photon hitting a droplet on the way from $\boldsymbol{a}$ to $\boldsymbol{b}$.

2.5.2.2 The homogeneous case

For a homogeneous volume of cloud, ρ and κ_e are constant throughout space and the total optical thickness becomes

$$\tau(\boldsymbol{a},\boldsymbol{b}) = l\rho\sigma_e,$$

thus the extinction function becomes

$$\beta(\boldsymbol{a},\boldsymbol{b}) = e^{-l\rho\sigma_e}.$$

A given volume of inhomogeneous cloud may be approximated as an homogeneous volume of constant density by averaging its density, *i.e.,* $\langle\rho\rangle = \iiint \rho(\boldsymbol{p})d\boldsymbol{p}$. Then this whole volume can be represented by its average extinction coefficient $\kappa_e = \langle\rho\rangle\sigma_e$.

2.5.2.3 Mean Free Path

free path

A *free path* is the straight path a photon takes between two scattering events. The *mean free path (MFP)* (MFP)

mean free path

$$l_0 = \frac{1}{\kappa_e} \tag{2.13}$$

of a volume of extinction coefficient κ_e represents the average length of the free paths. A flux of photons will be attenuated 50% after travelling a distance l_0 through the volume.

The effective radius r_e usually measured in real clouds is 5 µm – 15 µm, thus the effective extinction cross-section σ_e is about 150 µm^2 – 1500 µm^2. Given the usual cloud concentrations given in §2.4, the average extinction coefficient κ_e for a whole cloud varies between 10 km^{-1} and 100 km^{-1}, yielding a mean free path of 10 m – 100 m. Continental cumulus clouds, which are the ones we aim to model, are the most dense. For these clouds, a mean free path of 10 m – 30 m is common.

Given all these radiative properties, we are now able to answer the question posed at the beginning of this section. The answer is that most probably, our photon will hit a droplet within the first hundred meters of its way across the cloud. This also means that the droplets of a cloud that are visible to the eye are the ones located in the first few hundreds meters from the cloud boundary. Beyond this distance, everything is hidden by the droplets that are closest to the eye.

2.5.3 Multiple scattering and its consequences

2.5.3.1 Multiple scattering

As we explained in §2.5.2, since the clouds we are considering have a short mean free path (10 m – 30 m) a photon entering the cloud will most probably hit a droplet in the first 10 m – 30 m of its course after entering the cloud. It will then be scattered according to the Mie phase function (§2.5.1), *i.e.,* most probably in the forward direction. It will thus continue its course through the cloud. But 10 m – 30 m after the first scattering event, chances are that this photon will hit another droplet and be scattered again. If the albedo ϖ_0 of cloud droplets were low, this photon would eventually be absorbed. But since it is very high ($\varpi_0 = 1$), the photon will be scattered until it manages to exit the cloud. Given that a cloud spans several km, the photon will most probably be scattered multiple times before exiting the cloud. As an example, it is not rare for a photon to scatter hundreds of times in a cloud.

multiple scattering

order of scattering

backscattering

This phenomenon is called *multiple scattering* and is what governs light transport in a cloud. The *order of scattering* is the number of scattering events a photon underwent (see Figure 2.22). The *backscattering* represents the amount of photons that exited an object through the lit side through which they entered, rather than traversing the object. From our observations (§2.2.2.2) we know that the top of clouds is much brighter than their base, which indicates that backscattering is fairly high in clouds.

2.5.3.2 Multiple forward scattering

Since the phase function of cloud droplets is strongly forward scattering, intuition might lead someone to think that the result of multiple scattering is strongly anisotropic in the forward direction, and that backscattering is low. However, even though we know precisely the scattering behavior at the droplet level, the result of the multiple scattering of light by a whole cloud is very complex and far from similar to what is happening at the microscopic level.

For instance, let us compute the chances of our photon traversing the cloud from top to bottom by multiple strong forward scattering, *i.e.,* by being scattered multiple times by the strong forward peak of the phase function[5]. For a 7 km-thick *cumulus congestus* cloud having an average extinction coefficient

[5] Note that we are differentiating multiple *strong forward* scattering and multiple *forward* scattering. When mentioning multiple strong forward scattering, we mean light that has been scattered multiple times and always in the direction of the strong forward peak, *i.e.,* $\Theta < 5°$. When mentioning multiple forward scattering, we mean light that has been scattered multiple times and always in the direction of either the strong forward peak or the wide forward lobe, *i.e.,* $\Theta < \pi/2$.

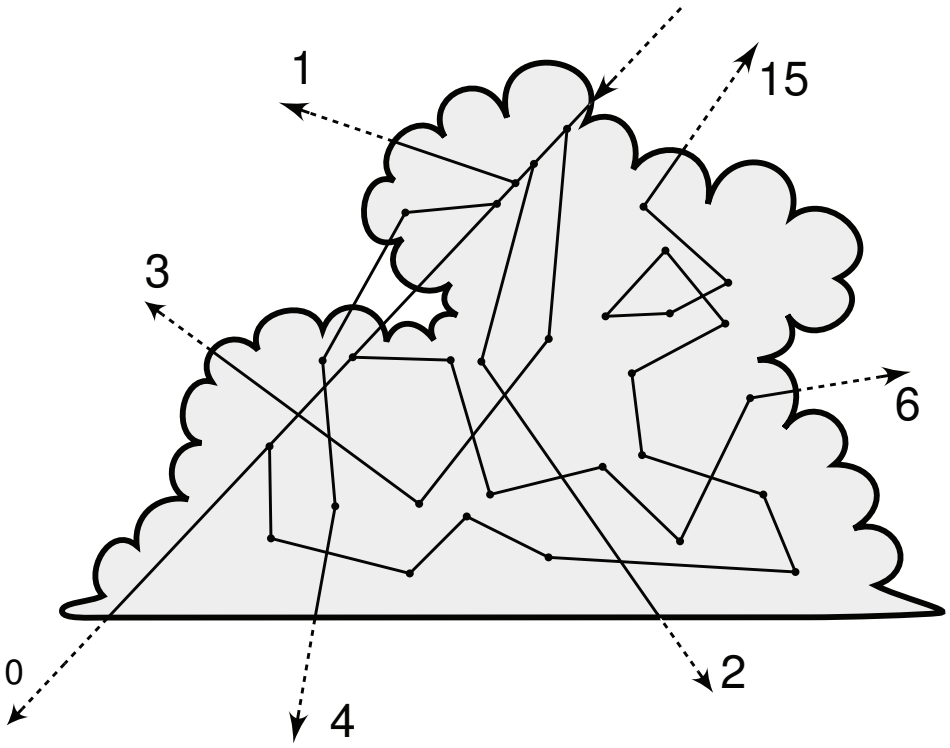


Figure 2.22: *Multiple scattering is what governs light transport in a cloud. A photon entering a cloud may traverse it without hitting any droplet (order 0), or after one scattering event (order 1), or after two scattering events (order 2), or 3, 4, etc. Note that the picture is a schematization. Since a cloud is very dense, in reality free paths are much shorter than what is shown on the figure.*

$\kappa_e = 50\,\mathrm{km}^{-1}$, the photon would have to undergo $7 \times 50 = 350$ strong forward scattering events. If we consider a scattering angle $\Theta < 5^\circ$to be strong forward scattering, this corresponds to a 50% chance of strong forward scattering per scattering event. Thus, the probability of this photon reaching the bottom of the cloud *as the sole result of multiple strong forward scattering* is $0.5^{350} \approx 4 \times 10^{-104}\%$. In comparison, the chance of the photon escaping the cloud by scattering backwards at the first scattering event is 1%.

Of course, things are more complex than that. If the amount of light traversing a cloud was this low, it would be pitch black on any cloudy day. This example only shows that light transport in thick clouds cannot be explained by *the sole action of multiple strong forward scattering*. That is, the anisotropy of the phase function of cloud droplets, although strong, is not what makes light traverse a thick cloud.

2.5.3.3 Light spreading

When a beam interacts with a scattering medium, it is said to undergo *spatial* and *angular spreading* [MFSR77, Sto78, MCH87] (see Figure 2.23). This spreading has been extensively studied in applied optics [Pea86, Gor94, LCH95, MFW98]. The more a light is scattered, the more it will spread angularly and spatially. This knowledge has been in CG used to render [PAS03] or capture [NN03] atmospheric phenomena. Tessendorf and Wasson [TW03] introduced the idea of the multiply-scattered phase function $P_n(\Theta)$ which is defined as the probability of light scattering through an angle Θ after

spatial spreading

angular spreading

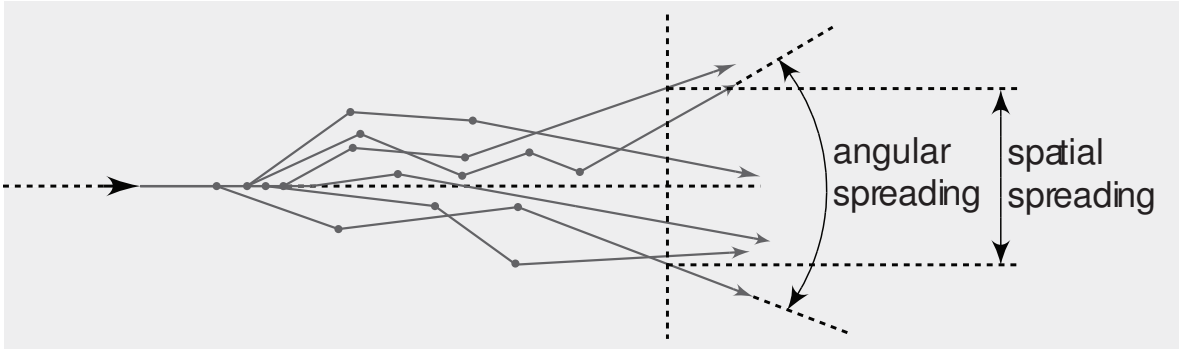


Figure 2.23: *A consequence of multiple scattering is spatial spreading and angular spreading.*

n scattering events. They have shown that this phase function becomes isotropic as n grows large. Thus, if the volume considered is large or dense, light transport becomes isotropic for high orders of scattering.

This explains why the base of the clouds is diffuse and display low contrast as we have seen in §2.2.2. Because of the high thickness and size of clouds, the light that reaches their base undergoes high multiple scattering and thus spread both in space and angle. On the other hand, as we will see in Part II, in thin parts of clouds or in clouds of low density low orders of multiple scattering dominate. As a result, the angular and spatial spreading is low, which explains the view-dependent effects and high contrast observed in §2.2.2.

2.5.3.4 Attenuation

attenuation

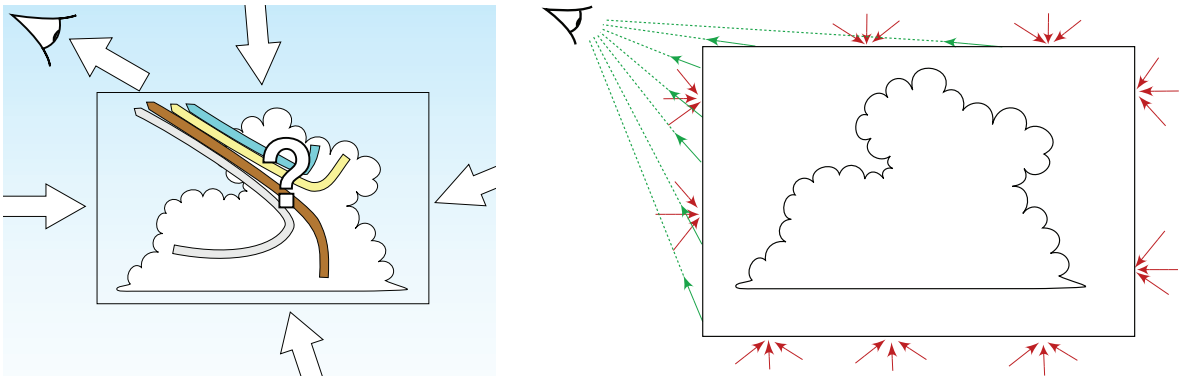
The term *attenuation* is used to describe that part of the light entering an object does not traverse it. As it has been explained by Bohren [Boh01, BC06], attenuation can have various causes. Absorption is the first cause that usually comes to mind, but scattering is also a cause of attenuation. It is scattering that causes attenuation in clouds, since their absorption is negligible.

We present an analysis of light transport in clouds in Chapter 4, in which we give more thorough explanations of multiple scattering in clouds and show how it causes all the visual features of clouds we described in §2.2.2.

2.5.3.5 Diffusion

Diffusion is the name of the process that takes place in a dense media with an isotropic phase function. It is a simplified formulation of the radiative transfer equation (which we will present in the next sections) that takes advantage of this isotropic behavior. This formulation is used, for example, to compute heat transfer in solid materials. Since clouds are quite dense, it has been used in CG to approximate of light transport in clouds, as we will see in §2.6.3.2.

One important aspect of diffusion is that it states that the thicker an object is, the more radiation will be reflected and attenuated. Since thicker clouds have a brighter top and darker base, it seems that light transport may be explained by diffusion. However, like multiple forward scattering the diffusion approximation is an appealing approach but cannot reproduce faithfully all the aspects of light transport in clouds, as we will see in §2.6.3.2.



(a) The classical approach for computing light transport in clouds. A box is defined around the clouds, and what needs to be solved is the radiative transfer equation inside that black box.

(b) Radiative transfer from the box's point of view. Red arrows indicate incoming luminance (*i.e.,* boundary conditions for L). Green arrows indicate outgoing luminance towards the viewer (*i.e.,* values of L to be solved).

Figure 2.24: *Representing the problem to solve with its boundary conditions.*

2.5.4 The Radiative Transfer Equation

To find a physical formulation of light transport in a cloud, let us leave the point of view of the photon and now take the point of view of a human observer. What the human eye sees when looking at a given location $\boldsymbol{p}$ of the visible cloud surface is the luminance $L(\boldsymbol{p}, \vec{s})$ traversing the cloud surface through an infinitesimal area $d\sigma$ at this point $\boldsymbol{p}$ in his direction $\vec{s}$ over an infinitesimal solid angle $d\omega$ (see §1.5.6).

To render the image of a cloud we need to solve $L(\boldsymbol{p}, \vec{s})$ for every point $\boldsymbol{p}$ on the visible cloud surface in the direction $\vec{s}$ of the viewer. Note that the definition of the "surface" is loose. As we explained in §1.5.6, the surface on which luminance is defined is imaginary and need not be a "real" surface. In the case of a cloud, the classical approach is to define a volume in which the cloud is contained, usually a 3D grid for which the boundary surface is thus a box (see Figure 2.24). Boundary conditions are set on that box (*i.e.,* the radiance on the surface of the box for incoming directions is defined), and what one needs to solve is the outgoing radiance on the visible surface of this box.

2.5.4.1 In-scattering and out-scattering

Scattering can also be seen from the point of view of an element of volume. For a given point $\boldsymbol{p}$ and direction $\vec{s}$, the term *in-scattering* refers to the light that, arriving in $\boldsymbol{p}$ from another direction, has been scattered in the direction $\vec{s}$. Conversely, the term *out-scattering* refers to the light that, arriving in $\boldsymbol{p}$ from direction $\vec{s}$, has been scattered in another direction (see Figure 2.25).

in-scattering

out-scattering

2.5.4.2 The equation of radiative transfer

The *equation of radiative transfer*[Cha60] allows us to compute the luminance[6] for any given point $\boldsymbol{p}$ in space toward any given direction .

equation of radiative transfer

[6]Note that what is called *intensity* or *specific intensity* in [Cha60] is actually *radiance* (resp. *luminance*) in radiometry (resp. photometry). In this work we use the *radiance* and *luminance* terms.

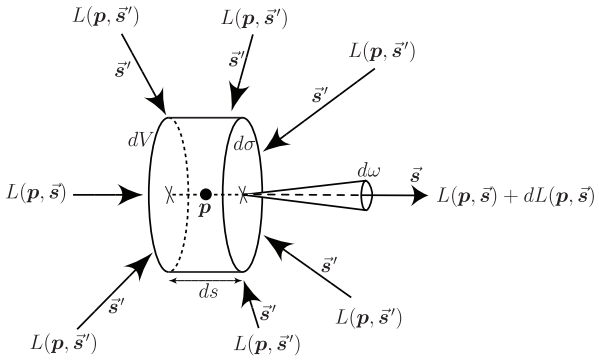


Figure 2.25: *Radiative transfer in an element of volume. The luminance differential along ds is the sum of in-scattered luminance minus out-scattered luminance.*

Let us consider an element of cylindrical volume $dV = d\sigma ds$, where $d\sigma$ is the cross section of the cylinder and ds is its length (see Figure 2.25). If we follow a luminous flux along the length of the cylinder, we find that the difference in luminance between the two ends is what has been in-scattered minus what has been out-scattered, *i.e.,*

$$\begin{aligned} dL(\boldsymbol{p},\vec{s}) &= -\text{ out}-\text{scattering} + \text{in}-\text{scattering} \\ dL(\boldsymbol{p},\vec{s}) &= -\kappa_e(\boldsymbol{p})L(\boldsymbol{p},\vec{s})ds + \kappa_e(\boldsymbol{p})j(\boldsymbol{p},\vec{s})ds \end{aligned} \tag{2.14}$$

emission coefficient

where $j(\boldsymbol{p},\vec{s})$ is the *emission coefficient*, which is due to in-scattering and is computed by

$$j(\boldsymbol{p},\vec{s}) = \frac{1}{4\pi}\int_{\|\vec{s}'\|=1} P(\vec{s},\vec{s}')L(\boldsymbol{p},\vec{s}')d\vec{s}'. \tag{2.15}$$

Equation 2.14 can be rewritten as

$$-\frac{dL(\boldsymbol{p},\vec{s})}{\kappa_e(\boldsymbol{p})ds} = L(\boldsymbol{p},\vec{s}) - j(\boldsymbol{p},\vec{s}),$$

and since the derivative along the cylinder is a directional derivative along $\vec{s}$

$$\frac{dL(\boldsymbol{p},\vec{s})}{ds} = (\vec{s}\cdot\nabla)L(\boldsymbol{p},\vec{s}),$$

we can write

$$-\frac{1}{\kappa_e(\boldsymbol{p})}(\vec{s}\cdot\nabla)L(\boldsymbol{p},\vec{s}) = L(\boldsymbol{p},\vec{s}) - \frac{1}{4\pi}\int_{\|\vec{s}'\|=1} P(\vec{s},\vec{s}')L(\boldsymbol{p},\vec{s}')d\vec{s}'. \tag{2.16}$$

Equation 2.16 is called the *equation of radiative transfer* for a scattering atmosphere. To compute the image of a cloud, the classical approach is solve $L(\boldsymbol{p},\vec{s})$ for every point $\boldsymbol{p}$ on the visible surface of the box in the direction $\vec{s}$ of the viewer, given the boundary conditions (see Figure 2.26). Here, the boundary conditions are the luminance $L(\boldsymbol{p},\vec{s})$ for every point $\boldsymbol{p}$ on the surface of the box in every direction $\vec{s}$ pointing inside the box. They correspond to the luminance of the surrounding environment (*i.e.,* sun, sky and ground luminance, see Figure 2.24). As we can see, $L(\boldsymbol{p},\vec{s})$ depends on its own derivative *and* integral, which makes Equation 2.16 an integro-differential equation, something quite difficult to solve.

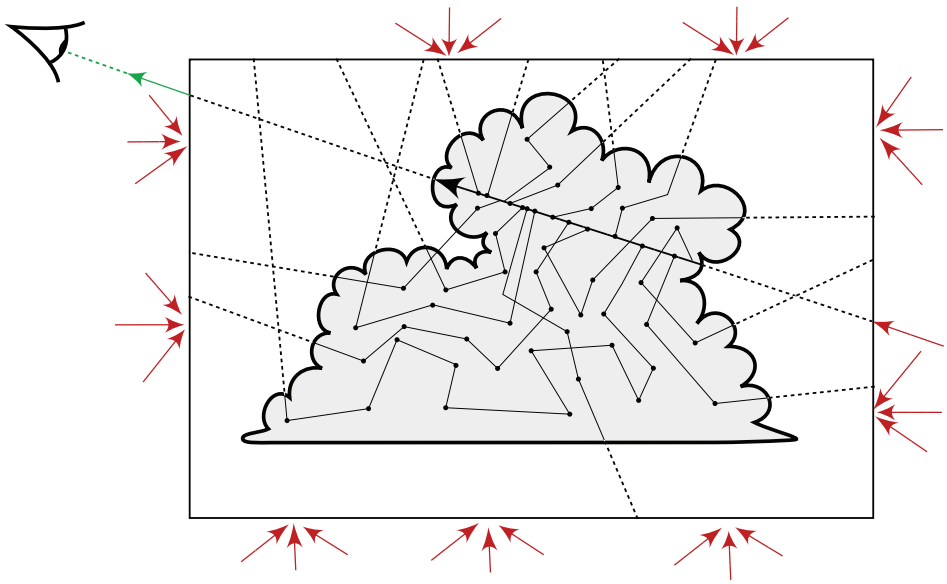


Figure 2.26: *Multiple scattering in the box from the camera's point of view. The luminance reaching the camera is the sum of luminance brought by all paths of multiple scattering.*

2.5.4.3 Successive orders of scattering

The successive order of scattering approach is another classical way of writing the general radiative transfer equation (Equation 2.16). It consists in separating the luminance reaching a point into the sum of the luminances brought by paths of each order of scattering.

We can decompose Equation 2.16 by separating the contribution of each scattering order as

$$L(\boldsymbol{p}, \vec{\boldsymbol{s}}) = \sum_{i=1}^{\infty} L_i(\boldsymbol{p}, \vec{\boldsymbol{s}}), \tag{2.17}$$

where $L_n(\boldsymbol{p}, \vec{\boldsymbol{s}})$ is the contribution of paths that have been scattered n times and is defined by

$$L_n(\boldsymbol{p}, \vec{\boldsymbol{s}}) = \int_0^{\infty} \beta(\boldsymbol{p}, \boldsymbol{p} + l\vec{\boldsymbol{s}}) j_n(\boldsymbol{p} + l\vec{\boldsymbol{s}}, \vec{\boldsymbol{s}}) dl \tag{2.18}$$

with

$$j_n(\boldsymbol{p}, \vec{\boldsymbol{s}}) = \frac{1}{4\pi} \int_{\|\vec{\boldsymbol{s}}'\|=1} P(\vec{\boldsymbol{s}}, \vec{\boldsymbol{s}}') L_{n-1}(\boldsymbol{p}, \vec{\boldsymbol{s}}') d\vec{\boldsymbol{s}}'. \tag{2.19}$$

The boundary conditions for this formulation are given through L_0.

The advantage of this formulation is that it gives a better understanding of the physical process, because a photon is followed at each scattering process. Moreover, if one only wants to compute the contribution of the first n scattering orders (*e.g.,* if the medium is absorbent and the contribution of the subsequent orders are negligible), it is not necessary to compute the contribution of higher orders, which reduces the necessary computations. It is also easier to solve, since the computation of L is reduced to a sum of integrals, for which there exist numerical solving methods (see §2.6.3.1).

2.5.4.4 Complexity of the equation of transfer

As we have seen, the original radiative transfer equation (Equation 2.16) is an integro-differential equation, which is not easy to solve. The successive order of scattering formulation (Equation 2.17) makes it only an infinite sum of infinitely nested integrals, which is only slightly easier to solve. Moreover, since the Mie phase function P can only be computed numerically, it is impossible to find a direct analytical solution for Equations 2.15 or 2.19, on which the others equations depend.

It is also interesting to note that even though we are only interested in the solution of Equation 2.16 for a restricted set of points and directions, solving this equation as it is formulated in Equation 2.16 requires a solution for every point and direction in the cloud. One way of solving this equation more quickly would be to find a different formulation that depends directly (instead of indirectly, which is the case here) on the boundary conditions. This is one of the properties of the models we will present in Chapter 5.

Two options [Len85] are usually taken to solve this equation: either simplify the initial problem to find an analytical solution, or find an efficient computational method. We present in the following subsections a few simplifications of the problem that have been used in previous works (in physics as well as in computer graphics). Computational approaches will be presented along with other CG methods in §2.6.3.1.

To simplify the problem, several options are possible:

- Compute only the first orders of scattering (*e.g.,* single scattering), ignoring multiple scattering. This approximation can be valid when the albedo or the density of the medium is low [Bli82]. Unfortunately, it is not the case for clouds.
- Assume homogeneity. This allows for a constant κ_e throughout the cloud. While this is not helpful by itself, it can simplify the problem when combined with the following options.
- Simplify the phase function. Considering an isotropic or simple procedural (*e.g.,* a Gaussian or Henyey-Greenstein) phase function simplifies the computation of $j(\boldsymbol{p}, \vec{s})$.
- Simplify the shape. Much work has been done on plane-parallel slabs or spheres, which are shapes that are studied in atmospheric optics and astrophysics.

In particular, the problem of multiple scattering in an homogeneous plane-parallel slab with an isotropic phase function admits an analytical solution. We present a couple of approaches and solutions dedicated to slabs in the following section.

2.5.5 Light transport in a slab

In this section, we explain the existing classical approaches for solving light transport in a homogeneous slab.

2.5.5.1 Frame of reference

A slab of cloud is a volume delimited by two parallel planes of infinite lateral extent, at a distance z_1 from each other (see Figure 2.27). Boundary conditions are defined uniformly over both planes, so that $L(\boldsymbol{p}, \vec{s})$ is constant with respect to $x_{\boldsymbol{p}}$ and $y_{\boldsymbol{p}}$. In this configuration, L depends only on $z_{\boldsymbol{p}}$ and $\vec{s}$. As a result, L becomes a function of three real variables $z = z_{\boldsymbol{p}}$, θ and ϕ instead of six originally.

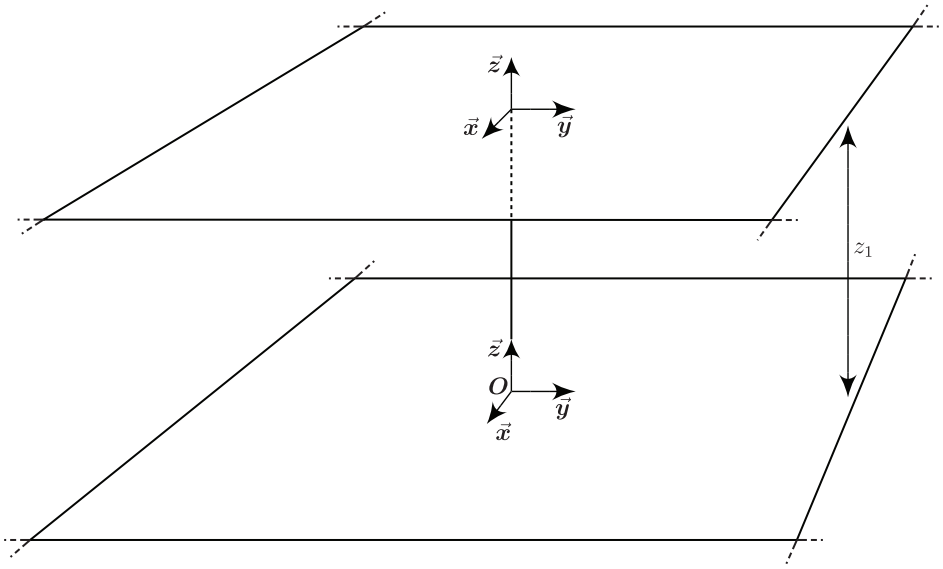


Figure 2.27: *Reference frame for the light transport in a slab of cloud. The top and bottom boundaries of the slab are infinite in the x and y directions. The bottom boundary is the plane $z = 0$ and the top boundary is the plane $z = z_1$.*

It is also commonly expressed in the literature [Cha60] as $L(\tau, \mu, \phi)$ through the optical thickness $\tau = \tau(z, z_1)$ and $\mu = \cos\theta$. This formulation makes it more practical to solve for cases when ρ varies with z. In the case we are considering (the density ρ is homogeneous), τ is simply $\tau = (z_1 - z)\sigma_e\rho$. It is also usual to see it written as $L(\tau, +\mu, \phi)$ and $L(\tau, -\mu, \phi)$, with $\mu > 0$, to distinguish between upward $(+\mu)$ and downward $(-\mu)$ flows.

2.5.5.2 Description of the problem

With these simplifications, the problem reduces to solving

$$\mu \frac{dL(\tau, \mu, \phi)}{d\tau} = L(\tau, \mu, \phi) - \frac{1}{2} \int_0^{2\pi} \int_{-1}^{1} L(\tau, \mu', \phi') d\mu' d\phi' \tag{2.20}$$

with boundary conditions defined through the incoming luminance distributions $L_\uparrow$ and $L_\downarrow$ as

$$\begin{aligned} L(0, +\mu, \phi) &= L_\uparrow(+\mu, \phi) \\ L(\tau_1, -\mu, \phi) &= L_\downarrow(-\mu, \phi). \end{aligned}$$

Many of the problems in radiative transfer are concerned with the transport of an incoming directional beam by a slab. That is, the radiance due to scattering when the slab is lit by a directional source of incident direction μ', ϕ', which translates to the boundary conditions

$$\begin{aligned} L_\uparrow(+\mu, \phi) &= 0 \\ L_\downarrow(-\mu, \phi) &= \delta_{\mu'}(\mu)\delta_{\phi'}(\phi). \end{aligned}$$

This problem is commonly written

$$L(\tau, \mu, \phi, \mu', \phi'). \tag{2.21}$$

Any other boundary conditions on a slab can be translated into this formula. Indeed, for any boundary conditions $L_{inc\uparrow}$ and $L_{inc\downarrow}$, we can write

$$\begin{aligned} L(\tau, \mu, \phi) &= \int\int L(\tau, \mu, \phi, \mu', \phi') L_{\downarrow}(-\mu', \phi') \sin\theta' d\mu' d\phi' \\ &+ \int\int L(\tau_1 - \tau, \mu, \phi, -\mu', \phi') L_{\uparrow}(+\mu', \phi') \sin\theta' d\mu' d\phi'. \end{aligned} \tag{2.22}$$

Thus, a formulation for $L(\tau, \mu, \phi, \mu', \phi')$ allows one to compute radiative transfer in a slab of thickness τ_1 for any given boundary condition[7].

diffuse reflection function

Finally, values of particular interest to physicists are the *diffuse reflection function R* and *diffuse transmission function T* which define the exiting radiances at the boundaries of the slab due to scattering

diffuse transmission function

$$R(\mu, \phi, \mu', \phi') = \mu L(0, \mu, \phi, \mu', \phi') \tag{2.23}$$

$$T(\mu, \phi, \mu', \phi') = \mu L(\tau_1, \mu, \phi, \mu', \phi'). \tag{2.24}$$

Note the similarity between R and a BRDF (§1.5.9). As a matter of fact, *R is exactly* a BRDF. T can be called a *bidirectional transmittance distribution function (BTDF)* (BTDF).

bidirectional transmittance distribution function

2.5.5.3 Properties

In every case, Helmholtz's principle of reciprocity implies that

$$\mu L(0, \mu, \phi, \mu', \phi') = \mu' L(0, \mu', \phi', \mu, \phi), \tag{2.25}$$

$$\mu L(\tau_1, \mu, \phi, \mu', \phi') = \mu' L(\tau_1, \mu', \phi', \mu, \phi). \tag{2.26}$$

This is the equivalent of the reciprocity invariance for BRDFs.

In the case of isotropic particles (*i.e.,* their phase function is rotationally invariant, which is the case for cloud droplets), L is rotationally invariant and bilaterally symmetric so L can be written as

$$L(\tau, \mu, \psi, \mu'), \tag{2.27}$$

with $\psi = |\phi - \phi'|$.

If the phase function of the particles is isotropic, L is axially symmetric and can be written as

$$L(\tau, \mu, \mu'). \tag{2.28}$$

2.5.5.4 Analytical expressions for single scattering and transparency

In a homogeneous slab, there exists an analytical solution for transparency and single scattering. The radiance due to transparency is calculated by the extinction function:

$$L_0(z, \mu, \phi, \mu', \phi') = \delta_{\mu'}(\mu)\delta_{\phi'}(\phi)\beta\left(\frac{z_1 - z}{\mu'}\right). \tag{2.29}$$

[7] Note that by doing this one ignores possible interference effects. It is acceptable in our case.

Knowing the radiance due to paths of order 0 (Equation 2.29) we can use the method of successive orders of scattering (Equation 2.18) to compute the radiance due to paths of order 1, *i.e.,* single scattering:

$$
\begin{aligned}
L_1(z,+\mu,\phi,\mu',\phi') &= \frac{1}{\mu}\int_z^{z_1} \kappa_e e^{-\kappa_e \frac{(z'-z)}{\mu}} P(\Theta) e^{-\kappa_e \frac{(z_1-z')}{\mu'}} dz' && (2.30)\\
&= P(\Theta)\frac{1}{\mu}\int_z^{z_1} \kappa_e e^{-\kappa_e\left(\frac{z'-z}{\mu}+\frac{z_1-z'}{\mu'}\right)} dz' \\
&= P(\Theta) e^{-\kappa_e\left(\frac{z_1}{\mu'}-\frac{z}{\mu}\right)}\frac{1}{\mu}\int_z^{z_1} \kappa_e e^{-\kappa_e\left(\frac{z'}{\mu}-\frac{z'}{\mu'}\right)} dz' \\
&= P(\Theta)\frac{\mu'}{\mu'-\mu}\beta\left(\frac{z_1}{\mu'}-\frac{z}{\mu}\right)\left[\beta\left(\frac{z}{\mu}-\frac{z}{\mu'}\right)-\beta\left(\frac{z_1}{\mu}-\frac{z_1}{\mu'}\right)\right] \\
&= P(\Theta)\frac{\mu'}{\mu'-\mu}\left[\beta\left(\frac{z_1-z}{\mu'}\right)-\beta\left(\frac{z_1-z}{\mu}\right)\right]. && (2.31)
\end{aligned}
$$

$$
\begin{aligned}
L_1(z,-\mu,\phi,\mu',\phi') &= \frac{1}{\mu}\int_0^{z} \kappa_e e^{-\kappa_e \frac{z-z'}{\mu}} P(\Theta) e^{-\kappa_e \frac{z_1-z'}{\mu'}} dz' && (2.32)\\
&= P(\Theta)\frac{1}{\mu}\int_0^{z} \kappa_e e^{-\kappa_e\left(\frac{z-z'}{\mu}+\frac{z_1-z'}{\mu'}\right)} dz' \\
&= P(\Theta)\frac{1}{\mu}\int_0^{z} \kappa_e e^{-\kappa_e\left(\frac{\mu' z-\mu' z'+\mu z_1-\mu z'}{\mu\mu'}\right)} dz' \\
&= P(\Theta) e^{-\kappa_e\left(\frac{z}{\mu}+\frac{z_1}{\mu'}\right)}\frac{1}{\mu}\int_0^{z} \kappa_e e^{\kappa_e\left(\frac{z'}{\mu}+\frac{z'}{\mu'}\right)} dz' \\
&= P(\Theta)\frac{\mu'}{\mu'+\mu}\beta\left(\frac{z}{\mu}+\frac{z_1}{\mu'}\right)\left[\beta\left(-\frac{z}{\mu}-\frac{z}{\mu'}\right)-1\right] \\
&= P(\Theta)\frac{\mu'}{\mu'+\mu}\left[\beta\left(\frac{z_1-z}{\mu'}\right)-\beta\left(\frac{z}{\mu}+\frac{z_1}{\mu'}\right)\right]. && (2.33)
\end{aligned}
$$

Equations 2.29, 2.31 and 2.33 can be computed analytically for any given phase function and boundary conditions. Higher orders cannot be computed analytically in the general case, numerical integration has to be used. The next subsections present a few methods for specific cases.

2.5.5.5 The X and Y functions

By using the properties of invariance of light transport into account, Chandrasekhar [Cha60] has shown that it is possible to express the light transport in a slab in the case of isotropic scattering in

terms of two basic functions: the X and Y functions as

$$L(0,-\mu,\mu') = \frac{1}{2}\left(\frac{1}{\mu+\mu'}\right)[X(\mu)X(\mu') - Y(\mu)Y(\mu')] \tag{2.34}$$

$$L(\tau_1,\mu,\mu') = \delta_{\mu'}(\mu)e^{-\tau_1/\mu} + \frac{1}{2}\left(\frac{1}{\mu-\mu'}\right)[Y(\mu)X(\mu') - X(\mu)Y(\mu')]\,. \tag{2.35}$$

This expression is quite powerful, in that the previous formulation of L with 2 degrees of freedom (DOF) μ and μ' can now be expressed at the boundaries of the slab simply through two 1-DOF functions X and Y. Several works have been devoted to solving these functions [DGS82], and pre-computed tables for the X and Y functions are widely available. However, note that this approach solves L only at the boundaries of the slab. It does not give a solution for $L(\tau,\mu,\mu')$ at $0 < \tau < \tau_1$. It is also restricted to particles of isotropic phase function, which is not the case of cloud droplets.

Note also the similarity between this formulation and the analytical solution for single scattering (Equations 2.31 and 2.33). Actually, these equations can be written in the form of the X and Y functions with

$$X(\mu) = 1,$$
$$Y(\mu) = \beta\left(\frac{z_1}{\mu}\right).$$

This indicates that the X and Y formulation can also be used to model the contribution of a given order of scattering (since it is applicable for order 1) and not only to model the contribution of all orders altogether.

2.5.5.6 The doubling-adding method

The doubling-adding method [van80] follows a simple principle. If we know the transmission and reflection properties of a reference slab of a given optical thickness τ_{ref}, then we should be able to deduce the transmission and reflection of a slab of thickness $2\tau_{\text{ref}}$ by considering it as a stacking of two slabs of thickness τ_{ref}. By extension, we can deduce the behavior of a slab of any given thickness.

For a slab with particles of isotropic phase function lit by a diffuse source, this method admits an analytical solution [Han71, BC06]. Let us consider a reference slab of optical thickness τ_{ref} receiving an illuminance $E_\downarrow$ on the top side and no illuminance on the bottom side. Let us assume we know the reflected exitance M_R and transmitted exitance M_T of such a slab (see Figure 2.28(a)). It can be modeled as

$$M_R = (1-\gamma_{\text{ref}})E_\downarrow$$
$$M_T = \gamma_{\text{ref}}E_\downarrow,$$

transmission coefficient

with γ_{ref} the *transmission coefficient* of this reference slab. Note that we are considering that there is no absorption. All of the incoming illuminance is either reflected or transmitted. We now want to compute the reflected and transmitted exitances M'_R and M'_T of a slab of thickness $2\tau_{\text{ref}}$. We can decompose this slab into two superposed slabs of thickness τ_{ref} (see Figure 2.28(b)) Let us call a the top slab, b the bottom slab, and $E_{a\to b}$ the transmitted exitance leaving a and illuminating b. It is the

sum of what would be transmitted by the reference slab plus the upward exitance of b reflected back to b by a, *i.e.,*

$$\begin{aligned} E_{a\to b} &= M_T + (1-\gamma_{\text{ref}})E_{b\to a} \\ E_{b\to a} &= (1-\gamma_{\text{ref}})E_{a\to b} \\ M'_T &= \gamma_{\text{ref}} E_{a\to b} \\ M'_R &= \gamma_{\text{ref}} E_{b\to a}. \end{aligned}$$

Thus,

$$E_{a\to b} = \gamma_{\text{ref}} E_\downarrow + (1-\gamma_{\text{ref}})^2 E_{a\to b},$$

which solves to

$$E_{a\to b} = \frac{1}{2-\gamma_{\text{ref}}} E_\downarrow.$$

We finally have

$$M'_R = \frac{\gamma_{\text{ref}}}{2-\gamma_{\text{ref}}} E_\downarrow \tag{2.36}$$

$$M'_T = \left(1 - \frac{\gamma_{\text{ref}}}{2-\gamma_{\text{ref}}}\right) E_\downarrow. \tag{2.37}$$

We can generalize Equations 2.36 and 2.37 to

$$M_R(\tau) = (1-\gamma(\tau))\, E_\downarrow \tag{2.38}$$

$$M_T(\tau) = \gamma(\tau) E_\downarrow \tag{2.39}$$

with

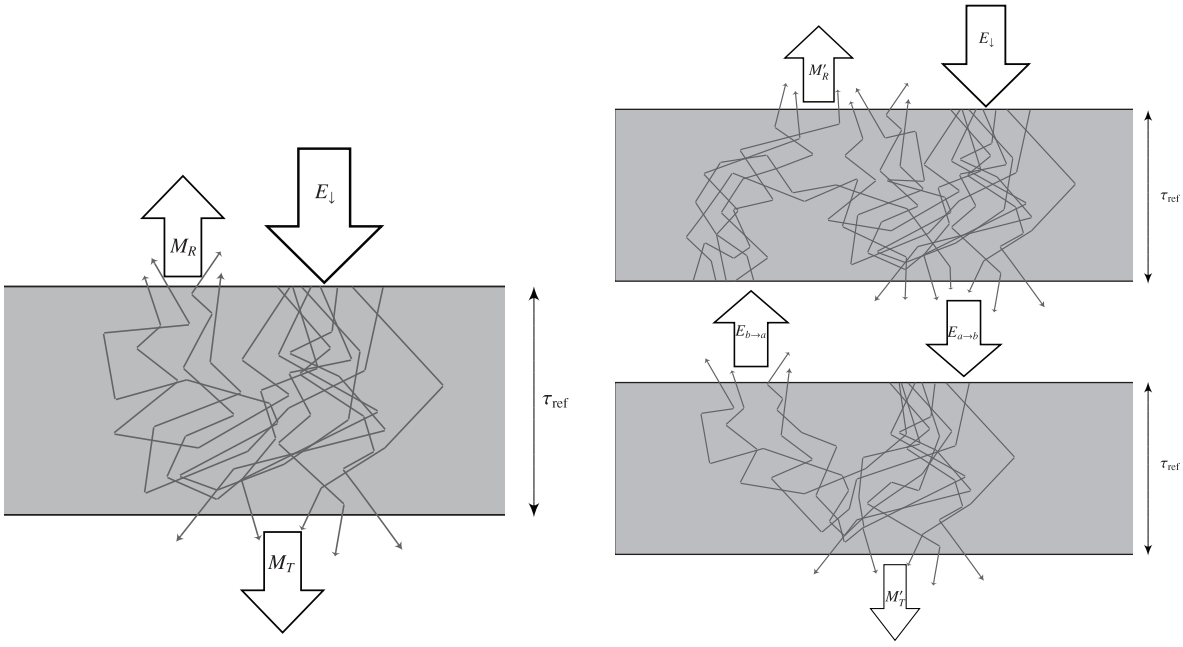
$$\gamma(\tau) = \frac{\gamma_{\text{ref}}}{\tau - (\tau - \tau_{\text{ref}})\gamma_{\text{ref}}}. \tag{2.40}$$

With this method, it is sufficient to know the reference transmission coefficient γ_{ref} for a slab of reference optical thickness τ_{ref} to be able to compute the transmission and reflection for a slab of any given optical thickness. This reference value γ_{ref} can be computed through any other method, *e.g.,* the analytic expression of single scattering with a very small τ_{ref} or any accurate multiple scattering method [Len85].

Note that even though this method allows one to compute the *exitances* reflected and transmitted by a slab of any given optical thickness, it does not give the solution to the *directional distribution* of these exitances, *i.e.,* the reflectance and transmittance. Moreover, note that γ_{ref} depends on the directional distribution of $E_\downarrow$ on the reference slab. That is, two equal illuminances $E_\downarrow = E'_\downarrow$ having different directional distributions (*i.e.,* $dE_\downarrow(\boldsymbol{p}, \vec{s}) \neq dE'_\downarrow(\boldsymbol{p}, \vec{s})$) do not necessarily yield the same reference transmission coefficient γ_{ref}. This implies that $E_\downarrow$ and $E_{a\to b}$ must have the same directional distribution for this approach to be valid.

As a result, it allows one to compute the luminance at the boundaries of a slab only in the case of a slab composed of particles with an isotropic phase function and with isotropic boundary conditions.

Summary of §2.5 : Light transport in a cloud is a very complex process governed by the laws of radiative transfer. In this process, scattering is the central operation and multiple scattering is the main phenomenon. It is dependent on the phase function of droplets which is given by the Mie model, and



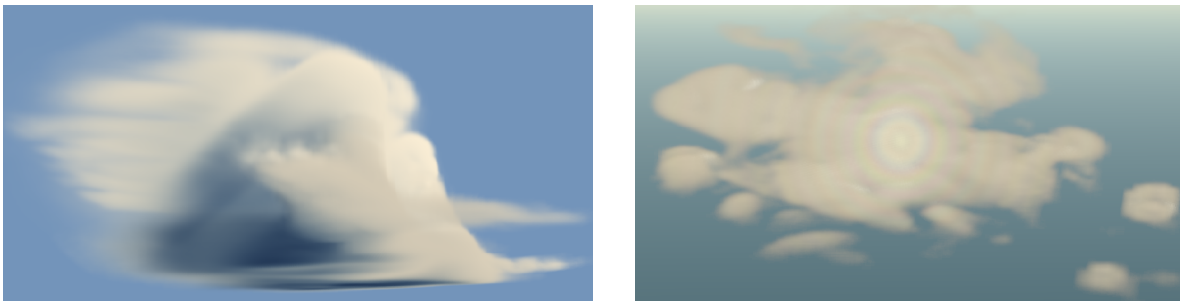
(a) Knowing the reflected and transmitted exitances of a slab...

(b) ...one can deduce the reflected and transmitted exitances of a slab twice as thick.

Figure 2.28: *Principle of the doubling-adding method.*

on the contents of the cloud. The radiative transfer equation applies at the microscopic level and do not give any idea of what is the macroscopic behavior of light for a whole cloud. In order to know the exact luminance flowing out of a cloud, this equation needs to be solved numerically and expensively. For very simplified cases (*i.e.,* simple shapes such as slabs, simple phase functions) there exist simpler, sometimes analytical, models such as the doubling-adding method or the *X* and *Y* functions.

Le transport de la lumière dans un nuage est un processus complexe gouverné par les lois du transfert radiatif. Dans ce processus, l'opération centrale est la diffusion et le phénomène dominant est la diffusion multiple. Elle dépend de la fonction de phase des gouttelettes, qui est donnée par le modèle de Mie, et du contenu du nuage. L'équation du transfert radiatif s'applique à l'échelle microscopique et ne permet pas d'avoir une idée du comportement macroscopique de la lumière pour un nuage entier. Afin de connaître la luminance exacte s'échappant d'un nuage, cette équation doit être résolue numériquement de façon coûteuse. Pour des cas très simplifiés (i.e., des formes simples telles que des dalles, des fonctions de phase simples), il existe des modèles plus simples, parfois analytiques, tels que la méthode doubling-adding *ou les fonctions X et Y.*



(a) [REHL03] (b) [REK+04]

Figure 2.29: *Clouds from Riley et al. represented by a* $128 \times 128 \times 128$ *grid generated from meteorological data. Procedural details are added with Perlin noise. The rendering speed is 1 FPS – 4 FPS.*

2.6 The computer graphics point of view

To render an image of a cloud, several elements are needed:

1. A description of its optical parameters, *i.e.,*
   - Its concentration (or density) $\rho(\boldsymbol{p})$, *i.e.,* a description of its "shape".
   - Its DSD $n(r)$, *i.e.,* a description of its "material".
   - A description of its radiative properties (phase function, extinction coefficient, etc.), normally induced from $\rho(\boldsymbol{p})$ and $n(r)$.
2. A description of the boundary conditions, *i.e.,* of the illuminance if the environment (sun, sky, ground, etc.) on the cloud.
3. A method to solve the light transport given all these parameters.

We present previous shape models in §2.6.1, previous material models in §2.6.2 and previous light transport solving methods in §2.6.3.

2.6.1 Shape models

Since a cloud is a 3-dimensional volume of droplets, what we call a "shape model" is a description of droplet concentration throughout the whole 3D space including the core of the cloud. According to §2.5 a description of the concentration $\rho(\boldsymbol{p})$ in every point $\boldsymbol{p}$ of the cloud is needed to compute light transport in it.

2.6.1.1 Meteorological data

To render a realistic cloud, it would be ideal to have values of $\rho(\boldsymbol{p})$ from real cloud measurements. It is nowadays possible to retrieve a complete 3D density field thanks to radar observation [Mas71]. Such a field is classically encoded into a volumetric 3D grid storing a concentration value in each of its cells. The resolution of this grid depends on the precision of the instruments used.

Using this kind of data, Riley *et al.* [REHL03, REK+04] have developed an interactive rendering method to render atmospheric phenomena including clouds from meteorological data (see Figure 2.29). Unfortunately, the resolution of such data is currently still too coarse for realistic applications. Indeed, to produce the image of a cloud at a 1024^2 resolution (the resolution of current screens),

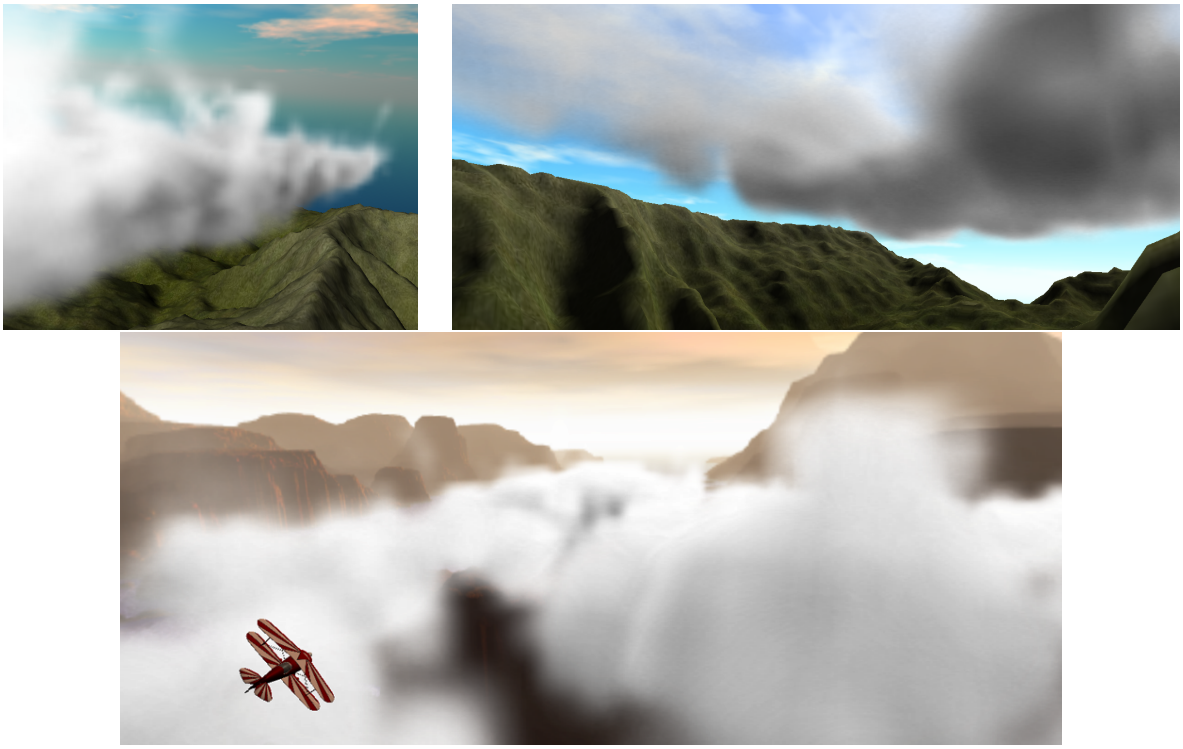


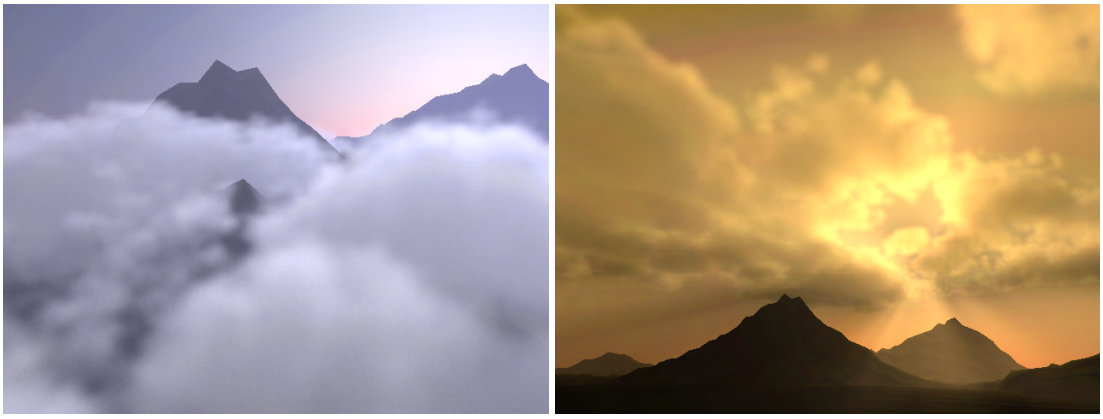
Figure 2.30: *Clouds from [HBSL03] generated from a simulation on graphics hardware on a* $64 \times 32 \times 64$ *grid. The resulting shape is more blurry than real cumulus clouds (see Figures 2.4 – 2.12). The distant clouds are not part of the simulation.*

the resolution of this 3D grid would need to be roughly 1024^3. This resolution needs to be even higher if the point of view is close to the cloud or if one wants to render a whole cloudy sky. The method of Riley *et al.* relies on a 128^3 grid, thus the resulting images are too blurry when compared to real clouds (we address discretization issues in more details in §2.6.3.4).

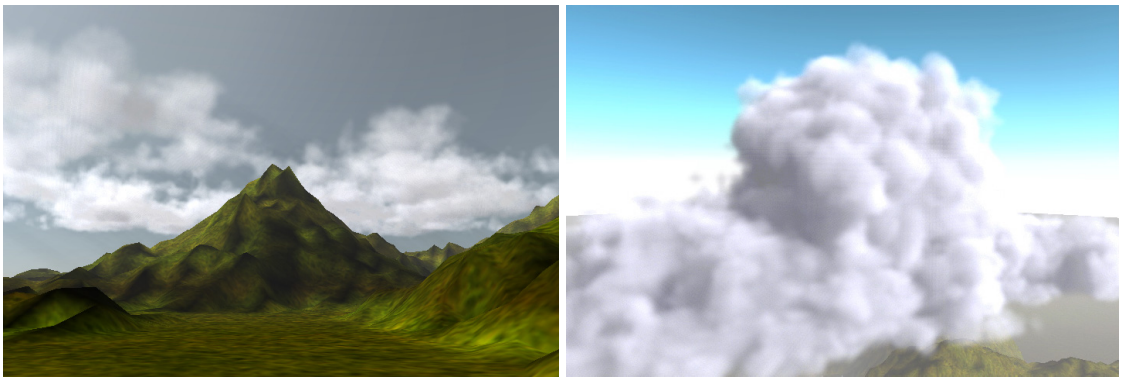
As a result, it is not possible to reproduce the fine details of clouds from meteorological data only.

2.6.1.2 Numerical simulations

An alternative is to use data computed numerically by a weather simulation. Although not all the physics of clouds is understood, enough is known to reproduce the fluid dynamics and thermodynamics of clouds. Such an approach has been used in computer graphics [Kv84, HBSL03]. Unfortunately, this approach also suffers from resolution problems. These simulations are run on a 3D grid stored in computer memory. They are limited by both the available memory and the expensive computation resources required to solve fluid dynamics (which increases in polynomial time with the resolution of the grid). As a result, such simulation data are also too coarse for realistic rendering (see §2.6.3.4). As an example, none of the existing methods provides a resolution fine enough to capture the high-detailed wisps of clouds or the sharp interface of cumulus clouds (see Figure 2.30). Another drawback of these methods is that it is very difficult to control their outcome so as to obtain a specific cloud shape because cloud dynamics (and fluid dynamics in general) have a very chaotic behavior [LF02]. This is a serious issue for methods aimed at being used by artists.



(a) Clouds from [DKY+00]



(b) Clouds from [MYND01]

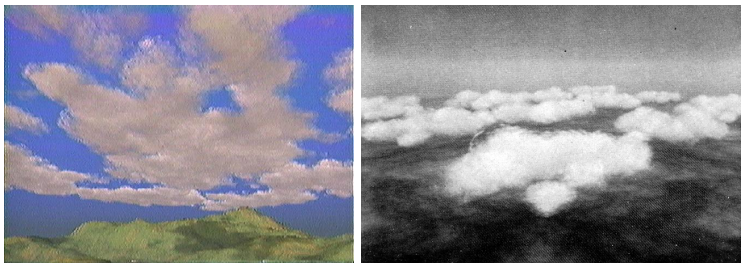
Figure 2.31: *Clouds simulated with cellular automata (top) or coupled map lattice (bottom) on* $256\times256\times20$ *grids. More details are present than for coarser grids (Figure 2.30), but the shapes are still blurry compared to real cumulus clouds (see Figures 2.4 – 2.12). The rendering technique is that of [DKY+00], based on multiple forward scattering. Note the low contrast and lack of backscattering as compared to real clouds. Aerial perspective and sky illumination is taken into account (note the shafts of light in the sunset picture).*

Since numerical weather simulations are computationally expensive, some other simulations approaches use cheaper procedural methods mimicking cloud formation on 3D grids using simpler rules, such as cellular automata [DKY+00] or "coupled map lattice" [MYND01]. While these approaches are computationally less expensive, they still suffer from the same resolution issue. With a resolution of $256 \times 256 \times 10$, some wisps begin to appear but cumulus clouds are still not sharp enough (see Figure 2.31).

2.6.1.3 Phenomenological and artist-driven approaches

Other procedural methods consist in modeling clouds using phenomenological approaches rather than direct simulations. Since clouds are either in the form of layers (stratiform) or heaps (cumuliform), these approaches usually model them using 2D planes [Gar85, Max86], height fields [Max94a], sets of ellipsoids [Gar85, Ney97, ES00, Ney00], or even sets of boxes [Wan04] (see Figures 2.32 and 2.33). These ellipsoids (or "metaballs" [NND96]) can serve as drawing tools [Ebe97, SSEH03] for an artist do design a cloud shape and reproduce the fluffy aspect of cumulus-type clouds. Before this thesis,

(a) Clouds represented by a simple stacking of 2D layers.



(b) Altocumulus clouds represented by ellipsoids and Fourier noise.

(c) Cumulus clouds represented by ellipsoids and Fourier noise.

Figure 2.32: *Clouds from [Gar85] modeled with planes and ellipsoids. Note the high level of detail brought by procedural noise. Rendering is offline, mainly due to the low performance of hardware at that time.*

we proposed such an approach to model clouds [BN04]. The shape resulting from these approaches is usually an envelop of the cloud, *e.g.,* in the form of a mesh. The concentration of droplets inside the cloud is defined with respect to this envelop (*e.g.,* constant inside the envelop and null outside the envelop) and is thus not depending on any grid resolution, avoiding the limitations of other methods.

Since these 2D planes or ellipsoids are only coarse descriptions, these approaches [Ebe97, SSEH03, KPHE02] add finer details via automatic procedural noise generators such as Perlin noise [Per85] or "Fourier noise" [Gar85] which they advect [Ney03] in case of animation (see Figures 2.32 and 2.33). Curiously, although this automatic procedural idea could be used to add low-scale details to numerical simulations to compensate for their coarse resolution, we do not know of any work doing this. On the other hand, it has been used to add details to meteorological data [REHL03]. A few works have used the ellipsoids approach to add details to very coarse meteorological data [DNYO98, TB02].

It is to note that there also exist such procedural methods to generate 3D cloud fields in the domain of atmospheric physics [I3R08]. As an example [EW04] use a coarse 2D shape retrieved from radar measurements to generate a detailed 3D shape, based on known statistics of water distribution in clouds. Seldom known in our domain, such methods could strongly benefit to the CG community.

2.6.2 Material models

Because the Mie phase function is costly to compute and following the uses of atmospheric physicists, most CG works use procedural approximations which are faster to compute, such as the

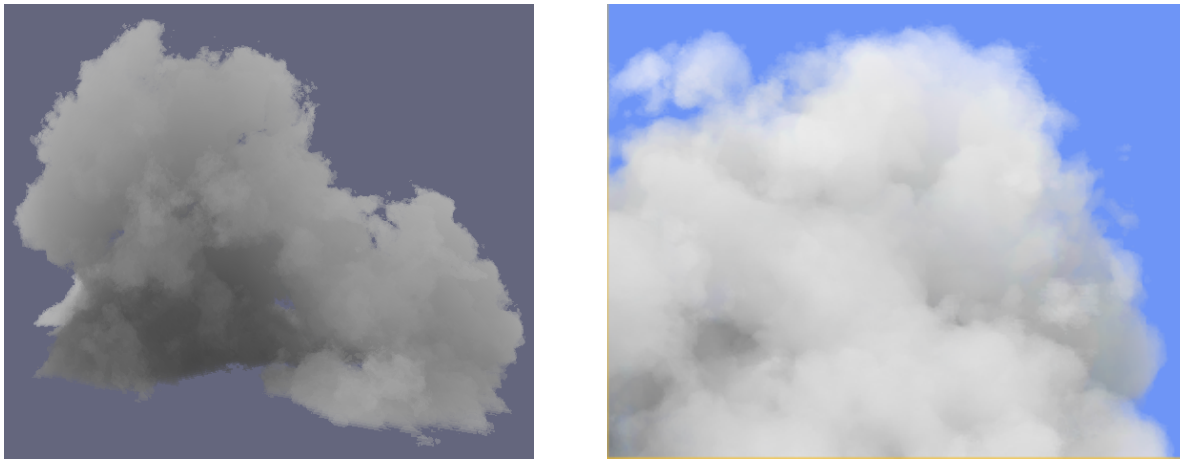


Figure 2.33: *Clouds from [SSEH03], modeled by an implicit surface created from ellipsoids and with Perlin noise. Note the fine detail added by the procedural noise. Interactive rendering framerates are obtained by approximating multiple forward scattering. Note the low contrast as opposed to real clouds (see Figures 2.4 – 2.12).*

Henyey-Greenstein function [Max94b, NND96, HAP05], the Gaussian function [PAT+04] or even the Rayleigh function [HL01]. Figure 2.34 shows some examples of these phase functions. These works do not rely on any DSD at all, and it is unclear how they determine the other optical properties such as the extinction coefficient.

The Henyey-Greenstein model [HG40] is a procedural phase function model designed to approximate various anisotropic phase functions. It is defined by

$$P(\Theta) = \frac{1 - g^2}{(1 + g^2 - 2g\cos\Theta)^{\frac{3}{2}}} \tag{2.41}$$

where g is the asymmetry (anisotropy) parameter given by Equation 2.4.

The Rayleigh phase function is actually the phase function of atmospheric particles (which cause the sky to be blue, see §2.2.2.3 and §2.6.4), and is thus not desired for the rendering of clouds.

Although Henyey-Greenstein or Gaussian functions are anisotropic and can thus present view-dependent effects, these effects are very different from what a Mie phase function would give. Most often, the resulting contrast of the clouds is lower than in reality[8] (see Figures 2.30 – 2.33). Moreover, these phase functions do not reproduce the fogbow and glory.

Only recently has computer graphics research started using the effective phase function of clouds, taking the DSD into account [REK+04] (see Figure 2.29). The DSD can be modeled by procedural methods (see §2.4.2) or retrieved from remote sensing of real clouds [MSPS04]. The complex Mie computations have been made easier to handle thanks to freely available code and tools [BH98, Lav].

Although not used in computer graphics, there exist specific approximations of the Mie phase function used in atmospheric optics. One of them is to describe it in a base of spherical harmonics, which can make the solving of the radiative transfer equation easier.

[8] Note that this lack of contrast may also come from a coarse shape description as shown in §2.6.1, or from an approximative light transport model as we will see in §2.6.3.

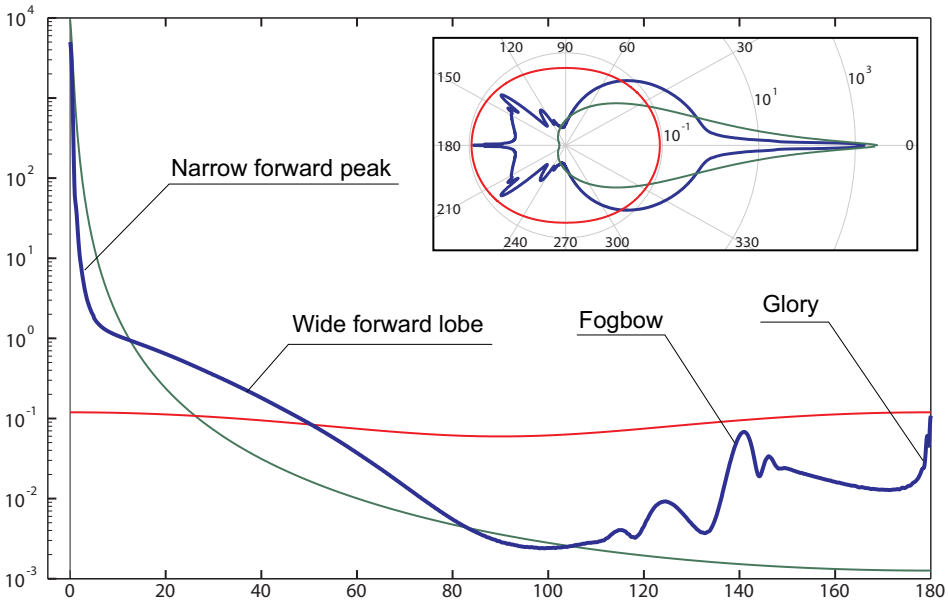


Figure 2.34: *Logarithmic plots of various phase functions used in previous works (inset: logarithmic polar plots).* Red: *Rayleigh.* Green: *Henyey-Greenstein with* $g = 0.99$. Blue: *Mie.*

Another interesting approximation is to approximate the narrow forward peak. This can be done by assuming that the flux scattered in the peak is simply transmitted in the direction of the incident beam [Len85, pp. 83 – 86]. This is equivalent to removing the peak from the phase function on one hand and decreasing the extinction coefficient on the other hand. We refer to this model as the "chopped-peak Mie" model. As we will see in our study of light transport (§4.4.2), this model is particularly handy and gives practically the same results as the original Mie phase function for multiple scattering.

2.6.3 Light transport models

Since the equation of radiative transfer in clouds is known (§2.5), the most straightforward approach is to solve it numerically. This approach is presented in §2.6.3.1. As we will see, it has many drawbacks. To overcome them, other methods take advantage of specific knowledge about clouds. We present these methods in §2.6.3.2.

2.6.3.1 General radiative transfer solvers

FEM methods The radiative transfer equation, especially in the form of successive orders of scattering (§2.5.4.3), can be directly solved via numerical methods. By discretizing space and directions into cells and bins, the problem becomes a finite element method (FEM), *i.e.,* roughly speaking a gigantic matrix to inverse. The *zonal method* [RT87] presents such an approach to solve multiple isotropic scattering for CG applications. The *discrete ordinates* method [Cha60] or the spherical harmonics method [Tan89, Eva98] handle anisotropic scattering, but their computational cost is extremely high. To worsen things, the strong anisotropy of the Mie phase function and the very high albedo of cloud droplets make this problem quite badly conditioned. Moreover, since all these FEM methods rely

FEM

discrete ordinates

on discretization, they suffer from resolution issues (see §2.6.3.4). Even if cleverer discretization methods can be used [QXF+07, SKSU05] instead of regular 3D grids, their results are still limited.

Monte-Carlo methods An alternative to FEM are Monte-Carlo methods. Monte-Carlo integration is a stochastic method for numerical integration. Given an integral of the form

$$I = \int f(x)dx,$$

this integral can be approximated by

$$I \approx \langle g(X) \rangle^{(N)} = \frac{1}{N} \sum_{i=1}^{N} g(x_i),$$

where $g(x) = f(x)/p(x)$, $p(x)$ is an arbitrary probability density function (PDF) and x_i are N realizations of a random variable X of PDF $p(x)$. The error ϵ between I and $\langle f \rangle^{(N)}$ is proportional to $1/\sqrt{N}$. Thus, the result of Monte-Carlo integration will be more precise as N increases. Depending on the choice for $p(x)$, the method can be more or less fast, accurate or biased.

PDF

In the case of clouds, the naive approach would be to use Monte-Carlo integration to compute the integrals involved in Equations 2.18 and 2.19 for every order n, every point $\boldsymbol{p}$ in the cloud and every direction $\vec{s}$, enabling one to compute Equation 2.17.

Many optimizations, variants and flavors of Monte-Carlo methods exist depending on the task. In the case of rendering, there exist two major methods to solve the radiative transfer that are cleverer than this naive approach. They are *photon mapping* and *Monte-Carlo path tracing*.

photon mapping

Monte-Carlo path tracing

Photon mapping is a very intuitive way of computing light transport. The method consists in "shooting" a large number of virtual photons from the light sources and having them stochastically interact with the media in the scene. Each time a photon is scattered (or reflected, in the case of a surface), its position and incident direction are stored in a *photon map*. When enough photons have been shot, the scene is rendered with classical path tracing [SM03], summing the contributions of the photons in the photon map. First introduced by Jensen [Jen96] for the computation of light transport between surfaces, it has been extended to scattering media [JC98] and is now thoroughly described in [Jen01].

Monte-Carlo path tracing [Jen01, JAD+03], which is a technique anterior to photon mapping, can be seen as the backward version of photon mapping. Instead of shooting photons from the light sources, this approach shoots "viewing rays" from the camera. This rays can be seen photons traveling backwards in time. They interact with matter in the exact same way as photons. At each interaction between a path and matter, instead of storing a photon in a photon map, "shadow rays" are sent in the direction of the light sources to compute the irradiance at the interaction point.

All of these approaches (FEM as well as Monte-Carlo methods) and their variants have advantages and disadvantages and can be combined but their common point is that they are too slow for realtime or even interactive rendering.

Precomputed radiance transfer One way to make any slow method faster is to precompute light transport in a first step so that the result can be displayed immediately at rendering time [GGSC96, LH96, SKS02]. However, this does not reduce the overall computation time needed and requires a lot of memory. Moreover, it prevents the animation of the shape of the cloud. A solution would be to store the solution for all possible cloud shapes (provided one knows them in advance), but this would have prohibitive memory requirements.

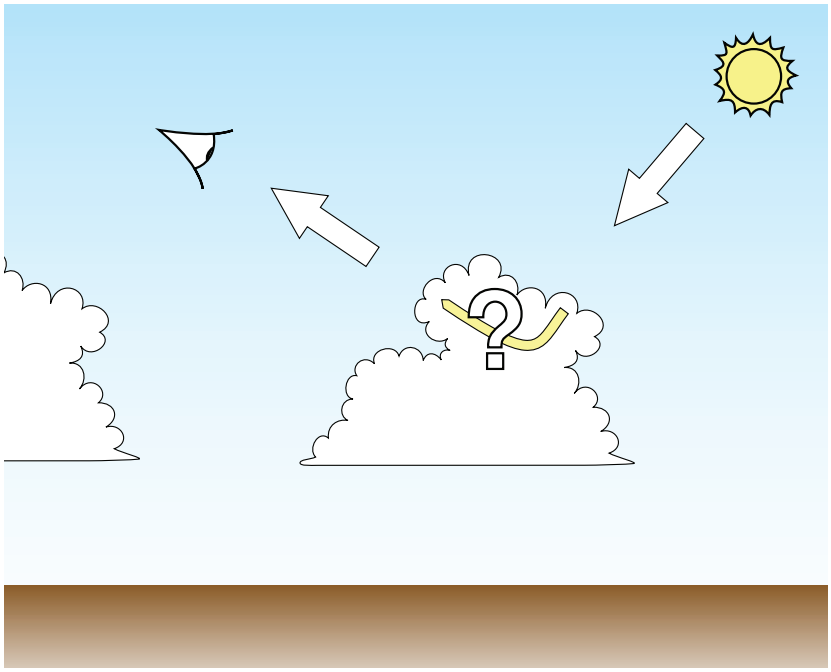


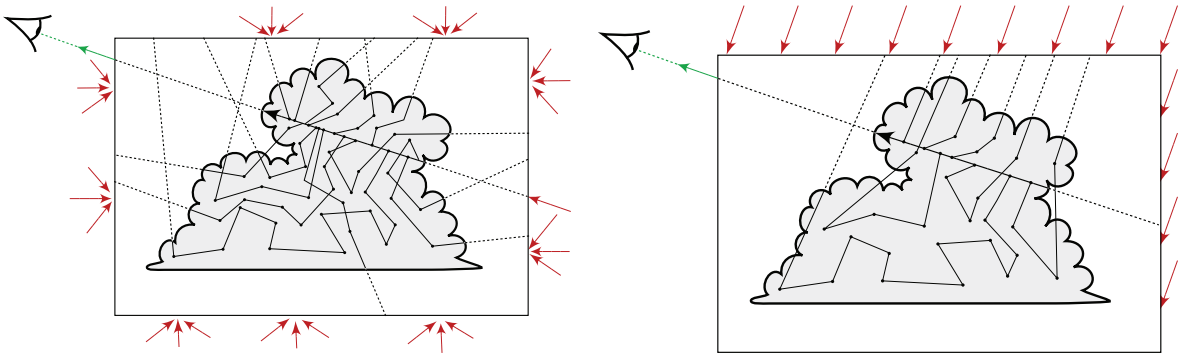
Figure 2.35: *A simplified version of the problem to solve. The problem becomes slightly simpler than in Figure 2.20. Of course, by doing this one misses all the visual features coming from sky and ground illumination.*

2.6.3.2 Cloud-specific methods

Introduction To find faster approaches than those presented in §2.6.3.1, one way is to take advantage of a priori knowledge concerning the objects to be rendered. The key idea is that if one knows first hand that a given term is going to be negligible, one can skip its computation and still obtain a faithful result.

As an example, Blinn [Bli82] used a low albedo hypothesis, which allowed him to compute only single scattering to render the rings of Saturn. In the case of clouds Kajiya *et al.* presented two methods for rendering participating media, depending on whether the albedo of the media was high or low [Kv84]. They show results for clouds. Max [Max94b] presented a clever offline method for speeding up and increasing the quality of the discrete ordinates method in the case of an anisotropic phase function, which applies to clouds.

Ignoring sky and ground contributions A very usual approximation is to consider that the sun is the only relevant light source for a cloud (*i.e.,* sky and ground illumination are negligible). This simplifies the boundary conditions and the computations (see Figure 2.35). As a matter of fact, not using this approximation is the exception in computer graphics. Only a few works on clouds take into account sky and ground illumination [DKY+00, NND96]. As we noted in §2.2.2.1, these sources are actually far from negligible since their contribution to a cloud's irradiance is *at the minimum* 25%. However, since they are very diffuse, their absence is less visible than if the sun contribution was to be missing. To make up for the absence of the subtle tints they bring, some works add an "ambient" light term [DKY+00, TB02].



(a) Taking all the environment into account. (b) Taking only the sun into account.

Figure 2.36: *Considering only the sun as shown in Figure 2.35, the number of paths to compute is greatly reduced. Of course, by doing this one misses all the visual features coming from sky and ground illumination.*

The multiple forward scattering approach Another common cloud-specific method is to take advantage of the strong anisotropy of the phase function. The idea is simple: since a droplet scatters almost of all the light in the forward direction, one can consider that

- The backward scattering is negligible.
- The result of multiple scattering by cloud droplets will still be anisotropic in the forward direction.

multiple forward scattering

This is the *multiple forward scattering* approach (see Figure 2.37). It has been proposed in many ways. The general idea is as follows.

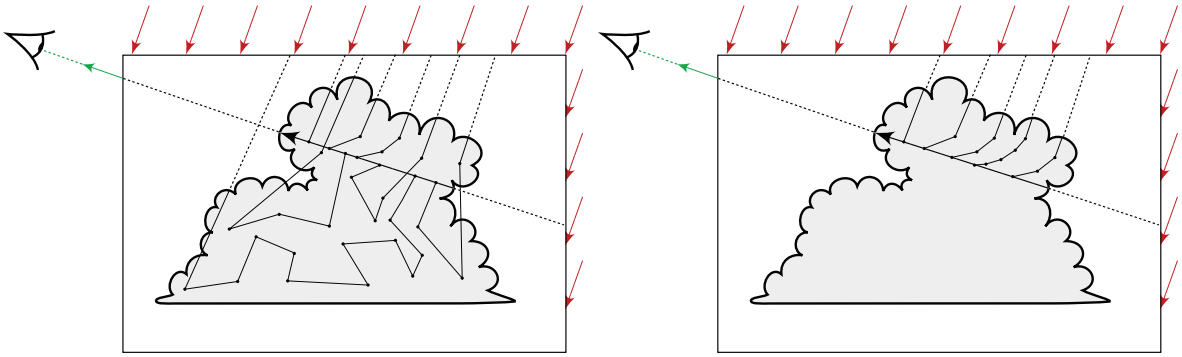
- In a first pass, the incident forward scattered light is accumulated along the incident direction into the cloud like in a photon map.
- In a second pass, the light reaching the viewer is computed by integrating in-scattering along the view direction.

This process is actually very similar to numerically computing single scattering only. The difference is that during the first step, light is spread (*i.e.,* it is accumulated in a forward solid angle rather than just in a unique direction) to mimic the spatial spreading of light due to scattering. This approach has been proposed in many flavors particularly because it is possible to implement it efficiently on graphics hardware [MDN04] and thus reach realtime performance [DKY+00, HL01, KPHE02, SSEH03, REK+04] (see Figures 2.29 – 2.31, 2.33, 2.39). Note that although [SSEH03] add procedural noise to their shape model, they do not take it into account in their light transport model, which removes some of the details and contrast.

most probable paths

MPP

Although this multiple forward scattering idea is rather intuition-driven, Premože *et al.* proposed a sound mathematically-based version of this approach in the form of the *most probable paths* (*MPP*) [PAS03, PAT+04] in the scope of Monte-Carlo path tracing (see Figure 2.38). The core idea is to trace only the paths that carry most of the luminance, assuming the others are negligible. Given the strong anisotropy of the phase function, the most probable paths this method finds are mostly forward or slightly curved (see Figure 2.37(b)). Their work is a practical approach for solving the angular and spatial spreading of light due to multiple scattering. They also brought it to realtime performance [HAP05]. Although it is more mathematically sound, it is quite close to other multiple forward scattering approaches.



(a) Taking all paths into account.

(b) Taking only the multiple forward scattering paths into account.

Figure 2.37: *The multiple forward scattering approach.*

However, although appealing, this multiple forward scattering approximation has its limits. As we explained in §2.5.3, multiple forward scattering is only one of the possible way light is transported in clouds and does not explain all the observed visual features. It may well explain the view-dependent features such as the silver lining but it is not the process involved in dense or thick parts of the clouds. As a result, these methods fail to reproduce the diffuse aspect of the bottom of the clouds and strongly underestimate the amount of backscattering, resulting in clouds having a dim lit side and low contrast as compared to real clouds.

The diffusion approximation Conversely, Stam has modeled light transport in clouds by a diffusion process [Sta95b, Sta95a]. This approach is valid if the medium is very dense, since the multiply-scattered phase function becomes isotropic. As one might expect, as opposed to multiple forward scattering approaches this methods reproduces well the strong backscattering at the top of clouds and the soft diffusive base of clouds but fails at reproducing the view-dependent effects which are due to the anisotropy of the phase function, such as the silver lining.

Other various methods Another way to speed up the rendering time is to approximate the initial problem regardless of a priori knowledge, which usually leads to missing visual features. Computing only the first orders of scattering [NND96] means ignoring the contribution of higher orders of multiple scattering which is important in the diffuse transmission of clouds, as we will see in Chapter 4. Considering the phase function as isotropic [DKY+00, HL01] eliminates all view-dependent effects.

Since light transport in clouds is very complex it is difficult to find a suitable phenomenological method. A few methods have been proposed [Gar85, Ney00, TB02]. Their advantage is that they are extremely efficient. Their drawback is that they reproduce the visual features of clouds too simplistically, failing to account for all the details and complexities of real clouds.

In addition to their extensive work on clouds modeling, animating and rendering [DNYO98, NND96, ND99, DKY+00, MYND01] Dobashi and Nishita have also proposed methods to render specifically the effects of lightning on clouds [DYN01, DEYN07]. We do not address these specific effects in this work.

Figure 2.38: *Clouds from [PAS03]. The model is the same as in [SSEH03] (note the fine details due to procedural noise). Here Monte-Carlo path tracing is sped up by the most probable paths approach, a mathematical formulation of multiple forward scattering. Diffuse sources were added at the boundary of the clouds to compensate for the underestimation of diffusive scattering.*

2.6.3.3 Other relevant methods

Subsurface scattering Curiously, no research work has considered using subsurface scattering-based techniques [HK93, JMLH01] to render clouds. This may be due to the fact that a cloud does not have a surface *per se*. However, as we have seen in §2.5.5, it is not necessarily an issue. Thus, BSSRDF models could thus be of some interest to us. This approach seems to have already been used in the industry [Dun04].

Practical subsurface scattering approaches use approximations that decrease the difficulty of the modeling of the BSSRDF. Hanrahan *et al.* [HK93] assume single scattering is the main phenomenon. Jensen *et al.* [JMLH01] use the diffusion approximation [Sta95b]. In addition, these works assume that light transport is a local process, *i.e.,* the light exiting the surface at a point $\boldsymbol{p}$ entered the surface in a neighborhood of $\boldsymbol{p}$. In the case of [JMLH01], this translates to two important simplifications: the surface around $\boldsymbol{p}$ is considered planar, and the area over which to integrate S is reduced to a limited area around $\boldsymbol{p}$. By using the dipole approximation of light diffusion [EVNT78, FPW92], they obtain a very efficient approximation of subsurface scattering for a photon tracing approach. Although their rendering is offline, this approach has been brought to realtime rendering speeds by Dachsbacher and Stamminger [DS03] via the use of graphics hardware. During this thesis, we have proposed an improvement of this method [BBNM07].

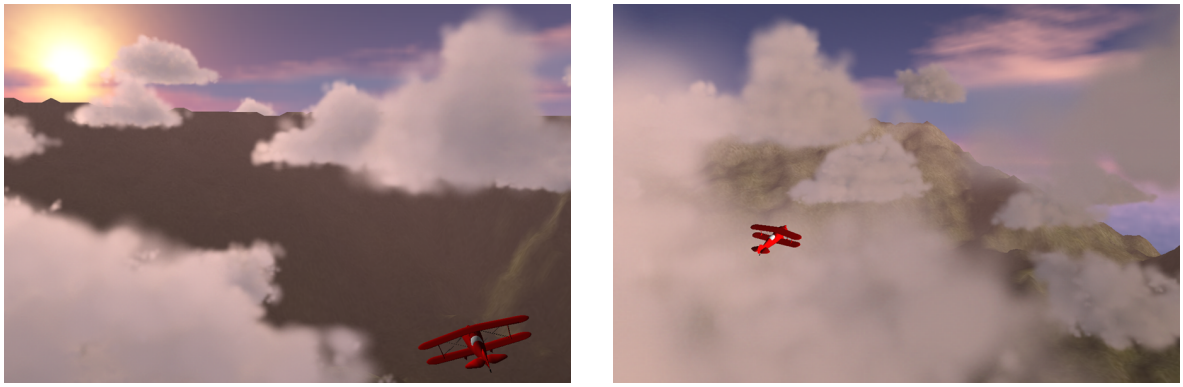


Figure 2.39: *Clouds from [HL01]. This realtime rendering method simulates low orders of multiple forward scattering. Note the low contrast on the edges compared to real cumulus clouds (see Figures 2.4 – 2.12).*

As we can see, the works on subsurface scattering use approximations that have already been applied to clouds (*e.g.,* single scattering, diffusion) or that are too restrictive in the case of clouds (*e.g.,* assuming light transport is local). As a result, they do not meet our requirements. Nevertheless, there is one important aspect of this approach that we wish to keep in mind: all these methods assume that *light transport is taking place in a restricted volume delimited by a surface*. This is a very sound idea in the case of the objects they are rendering, which are well-bounded solid (or liquid) objects. In the case of clouds, this idea has never been applied. Since cumulus clouds have a sharp interface, it might be interesting to use this paradigm, even if none of the existing subsurface scattering approaches are fit for clouds.

Radiosity The radiosity method [GTGB84] is a FEM designed to solving the problem of light transport between purely diffuse surfaces (the zonal method cited in §2.6.3.1 is the volumetric equivalent of the radiosity method). It cannot be used to solve light transport inside a cloud. However, as we will show in our first cloud rendering model (Chapter 6), it may be used to compute inter-reflections between the ground and the clouds. Since this method is a standard approach in computer graphics, we do not explain it in this thesis. We refer the reader to the literature [FvDFH90] for more information.

2.6.3.4 Hazards and issues of discretization

Because of the very discrete nature of computers and the absence of analytical solutions for most physical problems, discretization is at the core of a huge range of computational solving techniques. For any given problem, there usually exist one (if not many) computational solution in the form of solving a discretization of the problem.

In the case of light transport in a cloud for example, the discrete ordinates method consists in discretizing the space into cells and the directions into bins that are small enough so that the difference of radiance between them is below a precision threshold. However, because of the limitations in memory and computing power of current hardware, the resolution of these grids is quite coarse and this threshold cannot be as low as we would require.

The consequences of using too coarse grids can range from bad to disastrous. In the case of clouds, the first visible consequence is that the resulting shape is too blurry and lacks the contrast and fine details of real clouds such as wisps and sharp edges. However, this is only the tip of the iceberg.

The outcome of a simulation is different depending on the resolution of the grid. Moreover, some phenomena only happen at very fine scales and are simply not captured by a coarse grid, although they may influence large scale phenomena. For example, the convective movements creating cumuliform clouds gradually disappear on fluid simulation using too coarse grids. Stevens *et al.* [SBA+00] have shown that on a stratocumulus fluid simulation the resulting structure of the flow depended more strongly on the grid resolution than on the model or the solver used. Generally speaking, the use of coarse grids smoothes out vorticity [FSJ01], which is quite an issue for the simulation of convective clouds.

Finally, these methods scale poorly. As the resolution of the grid increases, the computational complexity and memory requirements increase by at least $O(n^3)$ [LF02]. It is thus useless to hope for the increase of computational power to make up for all these problems, especially in the scope of the simulation of a complete sky composed of detailed, fluffy clouds.

In addition to these errors, using such a grid may even be a waste of computation time. Indeed, let us consider a big, dense cloud spanning several kilometers. As we have seen in §2.5.2.3, its mean free path is about 10 m – 30 m. This means that everything that is beyond 200 m is practically invisible (there is a 99% opacity at 200 m for a 30 m mean free path). Thus, even if the grid was fine enough to capture all the details of the inhomogeneities in the core of the cloud, we would not see them directly. These details would of course affect light transport, and thus the appearance the cloud, but since one of the consequences of multiple scattering is the blurring of light (see §2.5.3.3) it is probable that accounting for these details would not make a visible difference. On the other hand, we know that the inhomogeneities at the boundaries of the clouds (*i.e.,* wisps) are directly visible and strongly participate to the appearance of a cloud.

Because of all these issues, one can seriously wonder about the quality of the result of computing very complex physical equations on a very coarse grid, as opposed to computing a more approximate formulation on a finer representation (or at least on a representation that tries to account for visual quality).

As a result, we try as much as possible not to use approaches based on blind and coarse discretizations in this thesis. When relying on volumetric representations (*i.e.,* grids), we try to use them wisely, *i.e.,* by avoiding wasting memory and computational time on invisible details (see §7.2.1).

2.6.4 Sky and aerial perspective models

Clouds are not the only scattering medium in the scenes we are considering. The atmosphere (*i.e.,* air molecules and dust particles) scatters light as well (it is this scattering that makes the sky look blue). Since clouds and atmosphere are vastly different media it makes sense to model them separately, which is the common approach in CG. Moreover, the atmosphere is far less dense that clouds. Single scattering predominates, which makes it easy to solve radiative transfer in the atmosphere.

Besides the blue sky, the most visible effect of atmospheric scattering is aerial perspective (see §2.2.2.3). It is the effect of scattering of light by the atmospheric particles that lie between the viewed objects and the viewer. At small distance scales, this aerial perspective is invisible. However, it becomes

clearly visible at large scales (see Figure 2.13). As an object is more distant from the viewer, it has more blueish and white tints and its contrast decreases.

The phase function for atmospheric particles is called the Rayleigh phase function [BH98]. It is only slightly anisotropic and it is not wavelength-dependent. The albedo of atmospheric particles however is wavelength-dependent, with a higher albedo for blue wavelengths. As a result it makes the sun look yellow or reddish and the rest of the atmosphere blue. Actually, the blue color of the sky can be seen as the aerial perspective for an object infinitively far away.

The other effect of atmospheric scattering is the shafts of lights as show on Figure 2.8(c) and described in §2.2.2.3. They are caused by the shadow that the clouds (or other objects) cast onto the atmosphere.

There exist several realistic models to compute the sky color and the aerial perspective for computer graphics [NMN87, PSS99, DKY+00, HP03], some of which are realtime. Some of them account for the shafts of light [DKY+00]. They can be easily integrated with the rendering of any other objects, including clouds. This allows one to benefit from a realistic outdoor environment in which one can render clouds.

Summary of §2.6 : To compute light transport at decent speed, computer graphics models use various approximations of the radiative transfer equation or phenomenological models induced from observations. Naive physically-based approaches do not meet our speed requirements. Cloud-specific methods (*e.g.,* multiple forward scattering approximation or diffusion approximation) fail to reproduce all the visual features of clouds. Pure phenomenological models do care about reproducing important visual features but do not reproduce them faithfully enough in all their complexity. Furthermore, rendering methods relying on volumetric grids suffer from discretization issues.

Pour calculer le transport lumineux à des vitesses décentes, les modèles en synthèse d'image utilisent différentes approximations de l'équation du transfert radiatif ou des modèles phénoménologiques déduits d'observations. Les approches naïves basées sur la physique n'atteignent pas les vitesses que nous requérons. Les méthodes spécifiques aux nuages (e.g., les approximations par multiple forward scattering *ou par* diffusion*) ne reproduisent pas toutes les caractéristiques visuelles des nuages. Les méthodes phénoménologiques pures mettent l'accent sur la reproduction des caractéristiques visuelles importants, mais ne les reproduisent pas suffisamment fidèlement dans toute leur complexité. De plus, les méthodes de rendu se basant sur des grilles volumiques souffrent de problèmes de discrétisation.*

2.7 Summary of Chapter 2

In this chapter, we have described the important visual features of clouds (§2.2), which are the ones we want to reproduce in our work: detailed shape, puffy or wispy edges, glory, fogbow, view-dependant lighting effects (silver lining, dark edges, pseudo-specular effect), bright creases, high contrast, sky and ground illumination, etc. We have documented this case study with examples and studies of photographs.

In §2.4 and §2.5 we have described how light is transported in clouds. A cloud is a collection of millions of microscopic water droplets (or ice crystals) scattering light in complex ways. The formulation

of the radiative transfer equation gives a microscopic description of light transport in clouds, but not a macroscopic one. As an example, a cloud droplet scatters 99% of the light it receives in the forward direction, yet a cloud is more reflective than transmissive.

Table 2.1 summarizes all the main properties of clouds.

In §2.6 we have presented the existing approaches for solving the radiative transfer equation. Dedicated computer graphics approaches for the fast rendering of clouds rely either on approximations of the equation of light transport based on simple heuristics (*e.g.,* "the light transport at the microscopic level is mostly forward, thus it should also be mostly forward at the macroscopic level" – which is too simplistic) or on simple phenomenological models which fail to capture all the complex behavior of light. Other classical approaches accurately solve the radiative transfer equation but on coarse grids, which gives a coarse and inaccurate result and fails to reproduce some features (*e.g.,* the sharp contrast between the lit and unlit parts of a cloud).

As a result, the goal of this thesis is to revisit the physics of light transport to propose new methods for the rendering of clouds. Our methods should be based on the physics of light transport, carefully approximated so as to obtain the same result and features as observed in reality while achieving real-time performance. To do so, we first try to understand the behavior of light transport in a cloud at a mesoscopic and macroscopic level and to understand the origin of the observed visual features. This is done in a study of light transport described in Chapter 4. This study enables us to find phenomenological models of light transport in a slab of cloud at a mesoscopic level (Chapter 5). Using these models we then propose two new approaches for the realtime rendering of stratiform (Chapter 6) and cumuliform (Chapter 7) clouds.

Dans ce chapitre, nous avons décrit les caractéristiques visuelles importantes des nuages (§2.2), qui sont ceux que nous voulons reproduire dans nos travaux: une forme détaillée, des silhouettes gonflées ou torturées, la gloire, l'arc-en-nuage, des effets dépendant du point de vue (silhouette lumineuse lorsque le soleil est derrière le nuage, sombre lorsqu'il est derrière l'observateur, effet pseudo-spéculaire), des creux entre les lobes lumineux, un contraste fort, l'illumination par le ciel et le sol, etc. Nous avons documenté cette étude de cas par des photographies.

En §2.4 et §2.5 nous avons décrit comment la lumière est transportée dans les nuages. Un nuage est une collection de millions de gouttelettes d'eau (ou de cristaux de glace) microscopiques diffusant la lumière de façon complexe. L'équation du transfert radiatif constitue une formulation décrivant ce phénomène à une échelle microscopique et non macroscopique. A titre d'exemple, une gouttelette de nuage seule diffuse 99% de la lumière qu'elle reçoit vers l'avant, et pourtant un nuage dans son ensemble est plus réflectif que transmissif.

Le tableau 2.1 résume les principales propriétés des nuages.

En §2.6 nous avons présenté les approches existantes résolvant l'équation du transfert radiatif. En synthèse d'images, les approches dédiées aux nuages reposent soit sur des approximations de l'équation basées sur des heuristiques simples (e.g., "le transport de la lumière à l'échelle microscopique est principalement vers l'avant, donc il devrait être aussi principalement vers l'avant à l'échelle macroscopique" – ce qui est trop simpliste), soit sur des modèles phénoménologiques simples qui ne permettent pas de reproduire totalement le comportement complexe de la lumière. D'autres approches classiques résolvent plus précisément l'équation du transfert radiatif, mais sur des grilles grossières, ce qui donne des résultats imprécis ou faux et ne reproduit pas non plus toutes les caractéristiques visuelles (e.g., le contraste fort entre les parties illuminées et non illuminées d'un nuage).

En conséquence, le but de cette thèse est de ré-étudier la physique du transport de la lumière pour proposer de nouvelles méthodes de rendu de nuages. Nos méthodes devront être basées sur la physique du transport de la lumière, approximée prudemment de façon à obtenir en temps réel les mêmes résultats et les mêmes caractéristiques visuelles que ce qu'on peut observer dans la réalité. Pour y arriver, nous allons d'abord essayer de comprendre l'origine des caractéristiques visuelles observées. Ceci est fait dans une étude du transport de la lumière décrite au chapitre 4. Cette étude nous permettra de trouver des modèles phénoménologiques du transport de la lumière dans une dalle de nuage à une échelle mésoscopique (chapitre 5. En utilisant ces modèles, nous proposerons ensuite deux nouvelles approches pour le rendu temps réel de nuages stratiformes (chapitre 6) et cumuliformes (chapitre 7).

Table 2.1 Typical values related of various cloud properties.

Property	**Typical values**
Illuminance (§1.5.7) on a cloud due to the sun during the day	$40{,}000\,\text{lx} \leq E_{\text{sun}} \leq 80{,}000\,\text{lx}$
Illuminance (§1.5.7) on a cloud due to the sky during the day	$10{,}000\,\text{lx} \leq E_{\text{sky}} \leq 20{,}000\,\text{lx}$
Illuminance (§1.5.7) on a cloud due to the ground during the day	$2{,}000\,\text{lx} \leq E_{\text{ground}} \leq 80{,}000\,\text{lx}$
Characteristic visual features due to the shape (§2.2.1)	Puffy, wispy or soft edges, with fractal detail
Characteristic visual features on the top of the clouds (§2.2.2)	Bright, crisp and detailed. Bright creases, dark edges. Overall white and blue colors
Characteristic visual features on the bottom of the clouds (§2.2.2)	Dark, soft and diffuse. Bright edges ("silver lining"), when the sun is behind the cloud . Overall grey and brownish colors
Mean droplet radius (§2.4)	$2\,\mu\text{m} \leq r_m \leq 8\,\mu\text{m}$
Droplet concentration (§2.4)	$50\,\text{cm}^{-3} \leq N_0 \leq 1500\,\text{cm}^{-3}$
Size (§2.4)	0 km – 15 km.
Asymmetry of the phase function (§2.5.1)	$g \approx 0.9$
Effective radius (§2.5.2)	$5\,\mu\text{m} \leq r_e \leq 15\,\mu\text{m}$
Extinction cross-section (§2.5.2)	$150\,\mu\text{m}^2 \leq \sigma_e \leq 1500\,\mu\text{m}^2$
Extinction coefficient (§2.5.2)	$10\,\text{km}^{-1} \leq \kappa_e \leq 100\,\text{km}^{-1}$
Mean free path (§2.5.2.3)	$10\,\text{m} \leq l_0 \leq 100\,\text{m}$

Chapter 3

Mathematical and computational tools

Contents

3.1 Introduction

We introduce in this chapter a few tools that we will use in the rest of this thesis. Interval estimation (§3.2) will be used to control the accuracy of the results of our simulations in our study of light transport (Chapter 4). High-performance computing (§3.3) will be used to run these simulations. GPU programming (§3.4) and HDR rendering (§3.5) will be used in our rendering models in Part III.

3.2 Interval estimation

In Monte-Carlo integration (§2.6.3.1), the integral $\mu = \int f(x)dx$ is approximated by the expected value of a random variable as

$$\mu \approx \langle g(Y) \rangle_N = \frac{1}{N} \sum_{i=1}^{N} g(y_i),$$

with y_i the N realizations of a random variable Y whose probability density function is $p(x)$ and with $g(x) = f(x)/p(x)$. If we call X the random variable such as $X = g(Y)$, we have

$$\mu \approx \langle X \rangle_N. \tag{3.1}$$

In Chapter 4 we study light transport by analyzing the result of various Monte-Carlo path tracing simulations. For this analysis to be robust, the error ε_N between μ and $\langle X \rangle_N$ has to be small enough. We present in this section a way to evaluate this error.

Let us call X a random variable as described by Equation 3.1. Let us call μ the (unknown) expected value of X. Let us write $\langle X \rangle_N$ the sample mean of X, *i.e.,*

$$\langle X \rangle_N = \frac{1}{N} \sum_{i=1}^{N} x_i$$

where x_i are realizations of X. Let us write $S_N^2(X)$ the sample variance of X, *i.e.,*

$$S_N^2 = \langle X^2 \rangle_N - \langle X \rangle_N^2$$

3.2.1 Interval estimation of the sample mean

central limit theorem

First, The *central limit theorem* [Ken51] states that as N increases, the distribution of $\langle X \rangle_N$ approaches the gaussian distribution with a mean μ and variance $\sigma_N^2(X)/\sqrt{N}$ irrespective of the PDF of X. Thus, with N sufficiently large, we know that $\langle X \rangle_N$ follows a gaussian (or normal) probability distribution around μ.

PDF

interval estimation

Second, *interval estimation* [Ken51] allows us to estimate the error between $\langle X \rangle_N$ and μ when $\langle X \rangle_N$ follows a gaussian distribution. The probability that $\langle X \rangle_N$ lies in the interval $[\mu - c, \mu + c]$ is $1 - \alpha$, with

$$c = t_{\alpha/2} \frac{S_N}{\sqrt{N}},$$

where

$$T_{N-1}(t_{\alpha/2}) = 1 - \frac{\alpha}{2}, \tag{3.2}$$

with $T_{N-1}(x)$ the cumulative distribution function (CDF) of Student's t-distribution with $N-1$ degrees of freedom. $[\mu - c, \mu + c]$ (or simply c) is called the *confidence interval* and $1-\alpha$ is called the *confidence level*.

CDF

confidence interval

confidence level

This means that if one wants $\langle X \rangle_N$ to lie in $[\mu - c, \mu + c]$ with a probability of $1-\alpha$, one has to choose N so that

$$t_{\alpha/2} \frac{S_N}{\sqrt{N}} \leq c,$$

which translates to

$$N \geq \left(t_{\alpha/2} \frac{S_N}{c} \right)^2 . \tag{3.3}$$

Equation 3.3 reads as follows: if we want to have $1-\alpha$ chances that $\langle X \rangle_N = \mu \pm c$, we have to take at least $\left(t_{\alpha/2} \frac{S_N}{c} \right)^2$ samples.

Because Student's t-distribution $T_{N-1}(x)$ converges to a gaussian distribution when N grows large, Equation 3.2 for defining $t_{\frac{\alpha}{2}}$ can be replaced by

$$\mathcal{N}(t_{\alpha/2}) = 1 - \frac{\alpha}{2},$$

where $\mathcal{N}$ is the CDF of the normal distribution.

It would be convenient to have c depending on μ (*i.e.,* as some fraction of μ). Since μ is unknown, c can be defined through $\langle X \rangle_N$ or S_N, depending on the needs. If we choose $c = k\langle X \rangle_N$, Equation 3.3 becomes

$$N \geq \left(\frac{t_{\alpha/2} S_N}{k \langle X \rangle_N} \right)^2 . \tag{3.4}$$

Note that in this case, the smaller μ is, the more samples have to be used. If we choose $c = kS_N$, Equation 3.3 becomes

$$N \geq \left(\frac{t_{\alpha/2}}{k} \right)^2 . \tag{3.5}$$

It is interesting to note that in this latter case, N does not depend at all on μ.

3.2.2 Interval estimation of the sample variance

We can also define a confidence interval for the estimation S_N^2 of the variance σ^2. Here, the probability that S_N lies in the interval $[b_1, b_2]$ is $1-\alpha$, with

$$\begin{aligned} b_1 &= \sqrt{\frac{N}{l_2}} S_N, \\ b_2 &= \sqrt{\frac{N}{l_1}} S_N, \end{aligned}$$

with

$$\begin{aligned} \chi^2_{N-1}(l_1) &= \frac{\alpha}{2}, \\ \chi^2_{N-1}(l_2) &= 1 - \frac{\alpha}{2}, \end{aligned}$$

with χ^2_{N-1} the CDF of the chi-square distribution with $N-1$ degrees of freedom.

Here again, b_1 and b_2 can be defined through $\langle X \rangle_N$ or S_N. If we choose $b_1 = S_N - kS_N$ and $b_2 = S_N + kS_N$, we obtain

$$\begin{aligned} N &= l_1(1+k)^2 \\ \Leftrightarrow \chi^2_{N-1}\left(\frac{N}{(1+k)^2}\right) &= \frac{\alpha}{2}. \end{aligned} \tag{3.6}$$

That is, to have $1-\alpha$ chances that $S_N = \sigma \pm k\sigma$, we have to take at least N samples where N is the solution of $\chi^2_{N-1}\left(\frac{N}{(1+k)^2}\right) = \frac{\alpha}{2}$.

We will use this knowledge in Chapter 4.

Summary of §3.2 : The error of the solution given by Monte-Carlo path tracing can be bounded by an error interval depending on the number of samples taken. This knowledge is important to ensure the simulations we run in our light transport study (Chapter 4) are accurate enough. This way, we can safely analyze them and derive models from them.

L'erreur dans la solution donnée par le tracé de chemin de Monte-Carlo peut être bornée par un intervalle d'erreur dépendant du nombre d'échantillons pris. Cette connaissance est importante pour s'assurer que les simulations que nous effectuons dans notre étude du transport lumineux (chapitre 4) sont suffisamment précises. De cette façon, nous pouvons sereinement les analyser et en dériver des modèles.

3.3 High-performance computing

high-performance computing

node

High-performance computing refers to the use of computer clusters (or supercomputers), *i.e.,* systems made of multiple computers (called *nodes*) linked together. Such systems are also called parallel computers. The use of many computers at once enables one to profit from the cumulated power of all the computers, provided it is used correctly.

In practice, developing software and algorithms to take efficient advantage of parallel computers is much more difficult than for standard architectures. Let us take a simple example. We want to compute $x = a + b + c + d$ on a 2-nodes cluster in an efficient way. What we need to do is divide the task into subtasks that will be either executed on node C_1 or node C_2.

This operation can be computed in different forms. One of them is

$$x = a + (b + (c + d)).$$

That is, $x = a + t_1$ with $t_1 = b + t_2$ and $t_2 = c + d$. Having divided the task into 3 subtasks, we can distribute them. Let us say C_1 will compute x and t_2 and C_2 will compute t_1. The problem of this repartition is obvious: x depends on t_1 which depends on t_2. Thus, C_2 needs to wait for C_1 to compute t_2 in order to compute t_1, then C_1 needs to wait for C_2 to compute t_1 in order to compute x.

As a result, this task division is not faster than if it was executed on a single computer, since only of the computer is effectively working at a time. Moreover, when taking the data transfers between C_1 and C_2 into account, this parallel implementation is actually *slower* than a standard one.

Thus, one of the keys of efficient supercomputing is to be able to divide a task into *independent* subtasks. In our example, this can be done by computing

$$x = (a + b) + (c + d),$$

that is, $x = t_1 + t_2$ with $t_1 = a + b$ and $t_2 = c + d$. This way, t_1 and t_2 can be computed concurrently on separate computers. Any non-independent tasks require *synchronization* and *communication* that decrease the computing efficiency.

However, this is not all. The other important thing to know is that the time complexity of a parallel algorithm depends both on the computing complexity (*i.e.,* the number of instructions to compute) and on the *network complexity* (*i.e.,* the amount of data transfer to do). If on a cluster of size K a problem of size N has a computing complexity $O(N/K)$ thanks to a wise subtask division but a network complexity $O(N)$, its total time complexity will be $O(N)$, which might not be better than the equivalent sequential algorithm.

network complexity

Some communication operations are very classical in supercomputing. A *scatter* operation is when a node sends the same data to all the others nodes. A *gather* operation is when all nodes send data to one node.

scatter

gather

If the time complexity of a serial algorithm is $O(N)$, an efficient parallel algorithm should have a computing complexity and a network complexity no greater than $O(N/K)$ for K nodes. We will make use of supercomputing in Chapter 4, in which we will apply these notions in order to implement an efficient Monte-Carlo path tracer.

Summary of §3.3 : Parallel programming requires the algorithms and data structures to be adapted (sometimes drastically) to the architecture, so as to get the maximum out of the computing power. It is why we alter the traditional raytracing algorithms in our simulations in Chapter 4 to make them run on a cluster.

La programmation parallèle requiert que les algorithmes et les structures de données soient adaptés (parfois drastiquement) à l'architecture afin de tirer profit au maximum de la puissance de calcul. C'est pourquoi nous modifions les algorithmes traditionnels de lancé de rayon dans nos simulations au chapitre 4 afin de les exécuter sur un cluster d'ordinateurs.

3.4 Graphics programming and GPU

Roughly speaking, a graphics processing unit (GPU) is a supercomputer with rasterization capabilities. It is composed of many small processors working in a single instruction, multiple data (SIMD) fashion. Each processor executes the same program (called *shader*) on different input data and outputs a different result. The input of these programs are 1D, 2D or 3D arrays, commonly called *textures*. These textures may represent color values (RGB textures), opacity values (alpha textures), distance values ("*depth maps*" or "*shadow maps*"), binary values (masks), or any other value one may imagine as long as it can be encoded by a limited set of floating-point variables.

GPU

SIMD

shader

texture

depth map

shadow map

In particular, current graphics hardware is optimized to compute distances from a given point to surfaces and to find the minimum of these distances in the view directions (depth maps). It is designed to quickly interpolate linearly values of neighboring *texels* (texture elements). It can provide a quick reasonable approximation of the average value of a set of neighboring texels. This last feature is called *MIP-mapping* and allows one to perform a computational integral very quickly, especially over a given square zone of the texture.

texel

MIP-mapping

The output of a shader is also an array, which can represent an image to be displayed on screen, or a texture to be used by another shader, or simply anything else.

The high computing speed provided by GPUs comes at a price. The possible computations that one can do in shader is limited and, in particular, random access to the memory is only possible in read mode. That is, a shader can read at any location in a texture but can write to only one restricted location in the framebuffer.

In Part III we will use GPU programming through shaders to achieve the realtime rendering of clouds. In particular we will take advantage of the MIP-mapping capabilities of GPUs to perform quick integrations.

3.5 HDR rendering and tone mapping

The human eye can detect luminance values ranging from 10^{-4} nits to 10^{8} nits in a logarithmic way. A typical computer screen encodes the colors it displays on 3 bytes (red, green, blue), yielding approximately 16 billion colors but only 768 possible luminance values. As a result, even with the best light transport method used, computer generated images can only approximate real-life images. The term high dynamic range (HDR) designates all the techniques and hardware relating to addressing these issues.

HDR

An HDR image is an image that encodes the full range of luminance values. An low dynamic range (LDR) image is an image that encodes only a partial range of values. A photograph from a normal digital camera is an typical example of LDR image. Because of the limited capabilities of the photodetectors, it saturates if regions of too high luminance and it is underexposed in regions of too low luminance, depending on the exposure. Similarly, a computer screen is a LDR display because it cannot display in the full range of visible luminance.

LDR

One of the first HDR method to overcome this issue is to have the screen map the input color values to the output luminance in a non-linear way. This mapping is called the *gamma correction*. It exploits the eye's logarithmic sensitivity by mapping input RGB values to the output luminance exponentially. However, it does not transform a computer screen into a HDR display.

gamma correction

Methods that convert an HDR image to a LDR image or display are called *tone mapping* methods. There exist a wide range of tone mapping methods, aimed at various goals, such as reproducing the behavior of a camera of the human eye. As shown in §1.5.12, the range of luminance of clouds is very wide. Since we are going to compute HDR luminance values of clouds to be displayed on an LDR computer screen, we need such a tone mapping method. We use a simplified version of Goodnight *et al.*'s technique [GWWH03]. We refer the reader to the existing literature for more details on available tone mapping techniques [DCWP02].

tone mapping

3.6 Summary of Chapter 3

We introduced a few methods we will use in the rest of this document, with which the reader should be familiar in addition to the units and notations presented in §1.4 and §1.5.

Interval analysis (§3.2) and parallel programming (§3.3) will be used to develop the simulation programs we will use in our study of light transport in slabs presented in Chapter 4.

GPU programming (§3.4) and HDR rendering (§3.5) will be used in our realtime clouds rendering models presented in Chapters 6 and 7.

Nous avons brièvement présenté quelques méthodes que nous utiliserons dans la suite de ce document. La connaissance de ces méthodes, ainsi que celle des unités et notations présentées en §1.4 et §1.5, est nécessaire pour la bonne compréhension de ce qui va suivre.

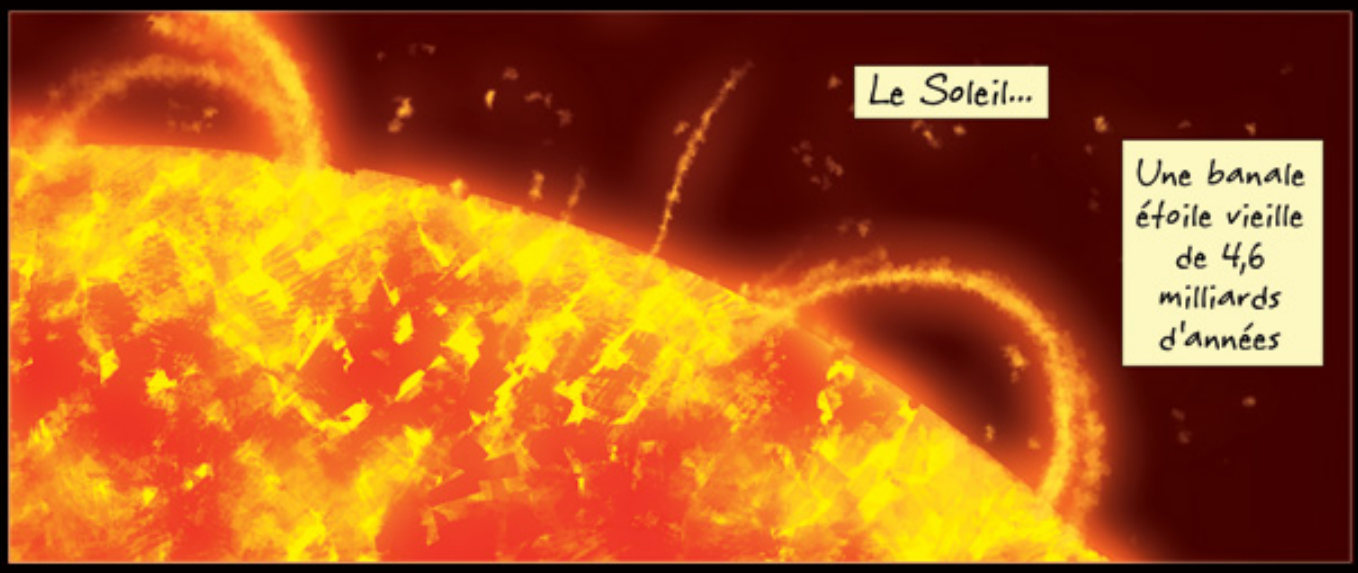
Les intervalles de confiance (§3.2) et la programmation parallèle (§3.3) seront utilisés pour développer les programmes de simulation que nous utiliserons dans notre étude du transport de la lumière au chapitre 4.

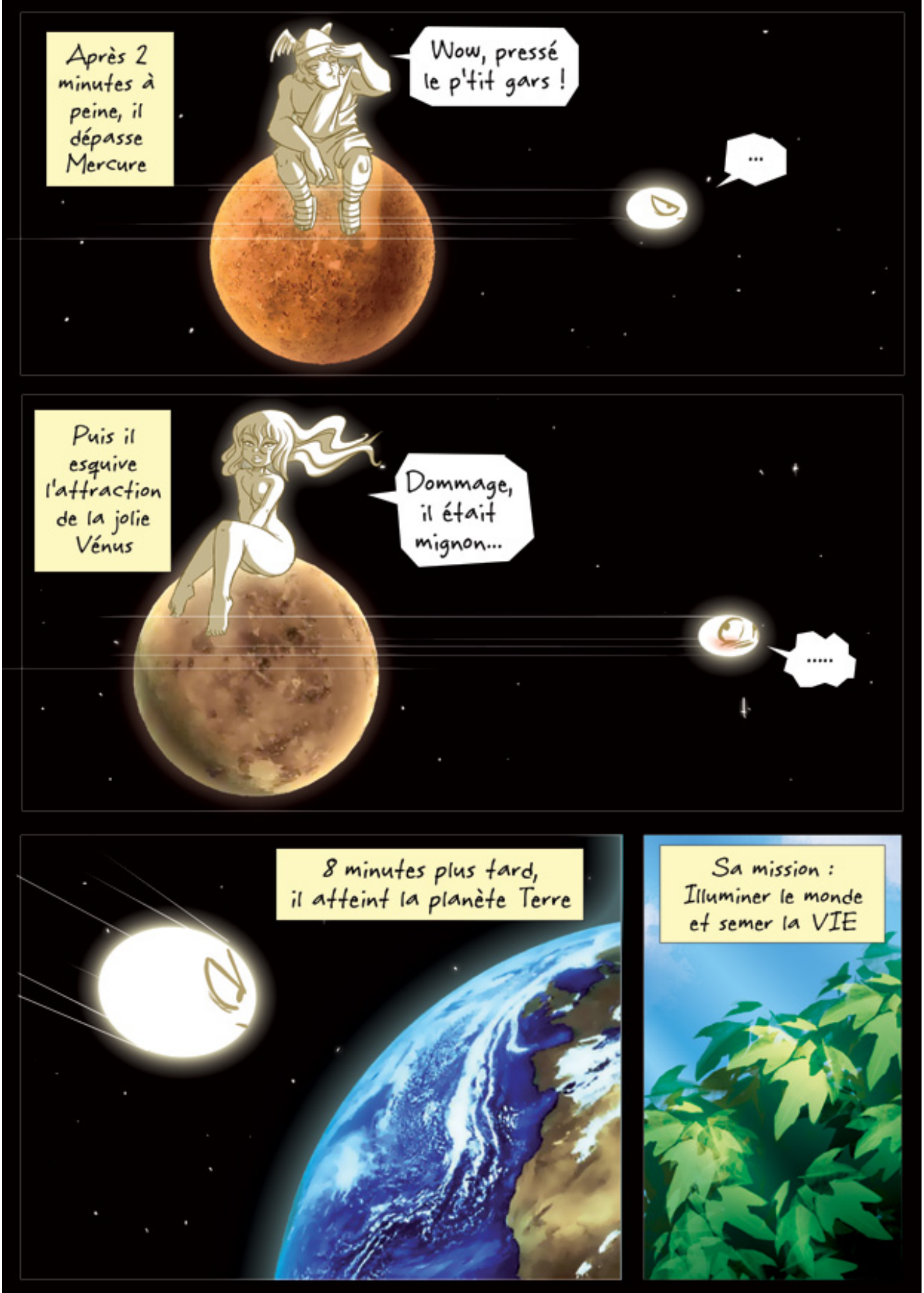
La programmation sur GPU (§3.4) et le rendu en HDR (§3.5) seront utilisés dans nos modèles de rendu de nuages aux chapitres 6 et 7.

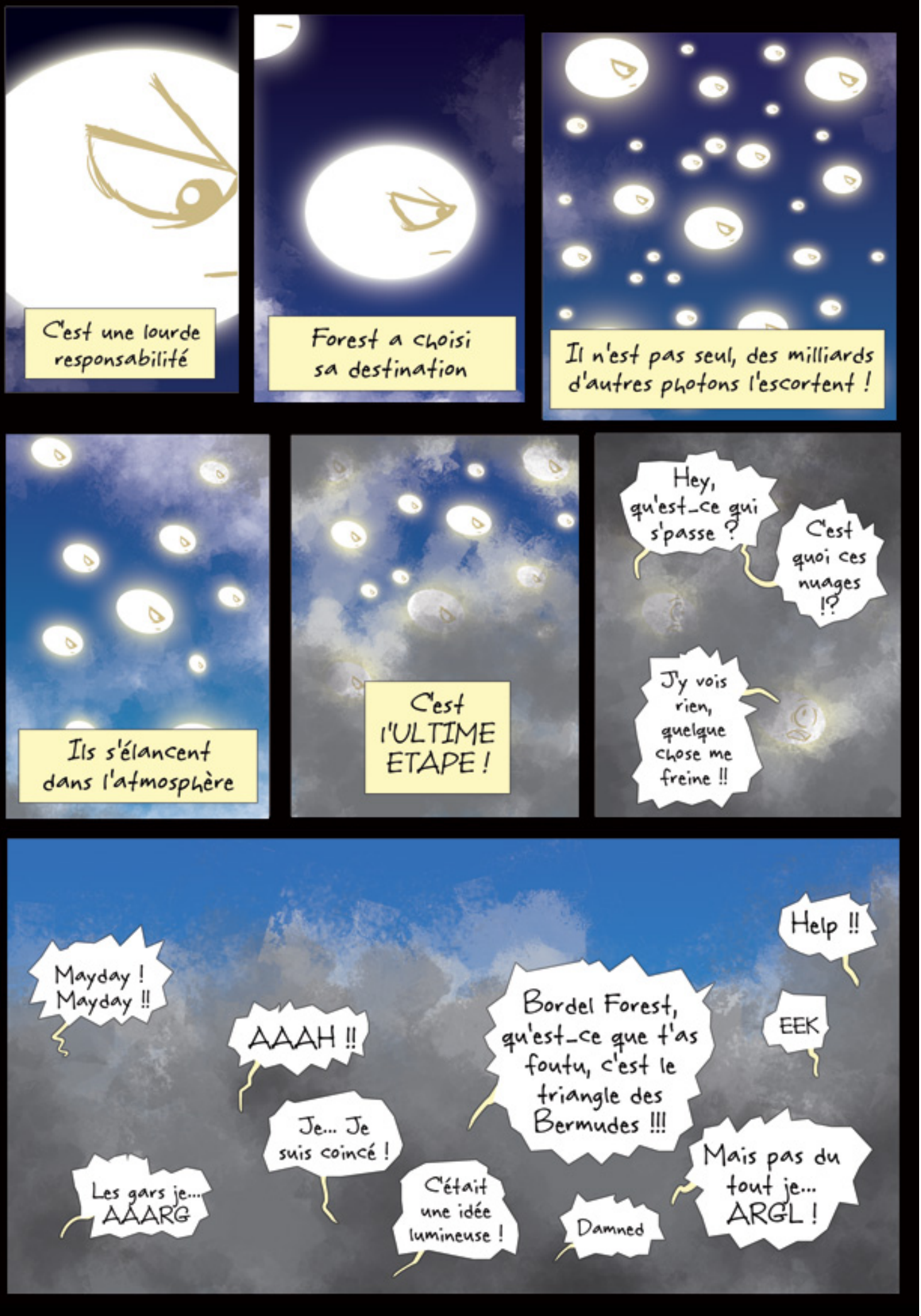
Part II

A study of light transport

COURS, FOREST !




Après 2 minutes à peine, il dépasse Mercure
Wow, pressé le p'tit gars !
...
Puis il esquive l'attraction de la jolie Vénus
Dommage, il était mignon...
.....
8 minutes plus tard, il atteint la planète Terre
Sa mission : Illuminer le monde et semer la VIE


C'est une lourde responsabilité
Forest a choisi sa destination
Il n'est pas seul, des milliards d'autres photons l'escortent !
Ils s'élancent dans l'atmosphère
C'est l'ULTIME ETAPE !
Hey, qu'est-ce qui s'passe ?
C'est quoi ces nuages !?
J'y vois rien, quelque chose me freine !!
Mayday ! Mayday !!
AAAH !!
Je... Je suis coincé !
Les gars je... AAARG
Bordel Forest, qu'est-ce que t'as foutu, c'est le triangle des Bermudes !!!
C'était une idée lumineuse !
Damned
Help !!
EEK
Mais pas du tout je... ARGL !

Quelques kilomètres plus bas, à ROUBAIX...

Chapter 4

A study of light transport in a slab of cloud

Contents

4.1 Introduction

4.1.1 Motivations

As we have seen in Chapter 2, light transport in a cloud is a very complex phenomenon. Our goal is to reproduce this phenomenon as faithfully as possible (*i.e.,* reproducing *all* the visual features listed in §2.2) in realtime. To achieve this goal, it is unavoidable to approximate the radiative transfer problem in order to do fewer computations. The issue is how to approximate it in a manner so as to keep the important visual features and ignore only the visually negligible aspects.

Various proposals for such approximations have been made in the past. The most common one for realtime rendering is the multiple forward scattering model (§2.6.3.2). We have seen that, although appealing, this model is limited. Intuitively, if multiple forward scattering were the dominant phenomenon in clouds, they should be brighter on their unlit side than on their lit side. This is true for thin clouds and cloud edges (thus the ability of this model to reproduce the silver lining feature), but not for the rest of the cloud. Most of the time, real clouds as seen from above are blinding white while their base can be quite dark. With the multiple forward scattering model, cloud tops are too dark and cloud bases are too bright.

Various questions arise from this observation: What are the precise limits of validity of the multiple forward scattering model? Why and where does it break down? What is the dominant phenomenon then? How can it be modeled? And more generally, what is the cause of each of the visual features wee see in clouds? By answering these questions we can start trying to find new models for the light transport in clouds that will faithfully reproduce these visual features.

4.1.2 Overview

This chapter is devoted to addressing the issues above. Through a solid and exhaustive experimental study of light transport in clouds we will answer these questions. This study will allow us to better understand the physics of clouds and give us a base from which we will develop new models for the light transport in clouds in Chapter 5 and Part III.

To explain how this study is driven, let us take a simple example. Let us imagine we want to model the reflectance of an alien surface of unknown material. That is, we have no physical description of this material. We are only provided with an instrument to measure luminance. First, we can describe this problem as finding the BRDF $R(\theta, \phi, \theta', \phi')$ of this surface (see §1.5.9). We may have some a priori knowledge of this material, such as R is rotationally invariant and can thus expressed as $R(\theta, \psi, \theta')$. This removes one degree of freedom from the problem. What we can do is illuminate this surface from various incident angles θ' and measure its luminance L in various directions (θ, ψ) using the luminancemeter. Then we can plot the resulting luminance measurements against ψ for various values of θ and realize they are constant with respect to ψ, *i.e.,* R is axially symmetric. As a result, the problem has now two degrees of freedom. Then we can plot L against θ' and see that the diagram resembles a $\cos\theta'$ function. Finally, we can plot L against θ and see that the diagram resembles a (say) $1 - \sin^x\theta$ function. As a result, we can probably approximate well this BRDF as

$$R(\theta, \phi, \theta', \phi') = a - b\sin^c\theta$$

where a, b, c are the parameters of the model found by fitting this model against the measurements. The quality of this model can be controlled by measuring the error between the measurements and the values predicted by this model.

This approach is a phenomenological approach. That is, it relies on observations of a phenomenon to describe it. Note that it used in various branches of physics. What we have done is this example is

1. Characterize the problem in terms of finding the BRDF of the surface
2. Choose the degrees of freedom of the problem. We decided to pick the incident and exiting angles as DOF and not, say, the location of measurement (*i.e.,* we assume the BRDF is uniform over the surface), the time (*i.e.,* we assume the BRDF does not change with time) or the wavelength of light (*i.e.,* we do not care about the color of the reflection).
3. Reduce the number of DOF as much as possible. Here we used the a priori knowledge that the BRDF was rotationally invariant.
4. Measure the resulting function for various sample values.
5. Analyze the resulting measurements and find empirical models fitting the measurements.

Since there exist no analytical model for the radiative transfer in clouds, we will use this approach to find a phenomenological model of radiative transfer in clouds. The issue here is that radiative transfer in any given cloud has an infinite number of DOF. As a result, in this study we will

1. Find a simpler case (*i.e.,* one that has a finite number of DOF) to study than an arbitrary cloud shape (*e.g.,* a slab)
2. Characterize the problem (*e.g.,* finding the BRDF or BSSRDF of a cloud)
3. Choose the degrees of freedom that seem significant to us.
4. Measure the resulting function for various sample values.
5. Analyze the resulting measurements.
6. Find empirical models fitting the measurements.

The measurements can be done on a virtual model of cloud with an accurate radiative transfer solver, instead of doing measurements on real clouds.

§4.2 describes steps 1 to 3, *i.e.,* the preparation of the study. §4.3 describes how we implemented our radiative transfer solver to do the measurements (step 4). §4.4 presents our observations and analysis of the results (step 5). The quick description of multiple scattering in clouds given in §2.5.3 is actually a very brief summary of the findings of this study. We develop it here in full detail.

Chapter 5 is devoted to present procedural models (step 6) matching the observed behaviors. These models were derived from the study presented in this chapter, and in particular from the analysis described in §4.4. We then present applications of these models to rendering of clouds in Part III.

4.2 Preparation of our study

For our study to be understandable and efficient, it needs to result in a *mesoscopic* description of light transport in clouds. Indeed, we are already provided with a *microscopic* description through the equation of radiative transfer (see §2.5.4). This theory allows us to understand the microscopic behavior of light transport in clouds, *i.e.,* light is scattered in a forward anisotropic way by individual clouds droplets, but not to understand the emergent phenomena (*e.g.,* why a cloud generally reflects more light than it transmits). We do not necessarily want a *macroscopic* description of light transport in clouds, *e.g.,* we do not need to know how much light a cloud *as a whole* reflects or transmits. What we are interested in is the behavior of light transport between these two scales. For instance, how

is light transported in thin parts of a cloud? How is light transported in cloud edges? How is light transported in thick parts of a cloud? How is light transported inside the core of a cloud? How is light transported in a crease between two lobes of cloud? And so on.

Finally, this description should be understandable, in order to grasp its meaning and find appropriate models to describe it efficiently. It should link light transport with the visual features we see in clouds. For instance, we know the glory and fogbow come from single scattering because they are clearly visible in the Mie phase function. It is plausible that the edge effects (silver lining and dark edges) come from multiple forward scattering. But it is not clear where the other features come from (*e.g.,* bright creases, bright lit side, dark and diffuse unlit side, pseudo-specular effect). In addition to these findings, this study should give us quantitative information about these behaviors, *e.g.,* help us know exactly how much light will traverse a given cloud edge.

> *"The success of any physical investigation depends on the judicious selection of what is to be observed as of primary importance, combined with a voluntary abstraction of the mind from those features which, however attractive they appear, we are not yet sufficiently advanced in science to investigate with profit."*
> – James Maxwell.

As we will see in the following sections, the number of DOF for this study can be rather high. In practice we are limited by two factors: storage space and computation time. The amount of space we need depends on the number of DOF and on the size of the output data. The amount of time we need depends on the number of DOF, on the efficiency of our simulator and on the precision of the results.

There are many things we wish to observe, but we have to constrain ourselves in the limits described above. Thus, we have to choose carefully the inputs and outputs of our study so that it has few DOF in order to stay within the limits. Then we have to write an efficient simulator. In the rest of this section, we list the possible inputs and outputs available for studying. We explain which ones we choose and why.

4.2.1 Available shapes for study

Studying light transport in the shapes we use for rendering would allow us to have a case as close as possible to our application. However, a study on a specific cloud shape would relate only to that shape and would not be applicable to other shapes. A study on any given cloud shape would yield an infinite number of DOF. Moreover, it would be hard to find an efficient and understandable model of light transport from arbitrary cloud shapes.

Some basic shapes have been studied in atmospheric optics such as slabs and spheres. Their disadvantage is that they are very different from real cloud shapes. They might, however, be close to parts of clouds. For example a sphere or ellipsoid or cylinder might be a good approximation of a cloud lobe. A slab might be a good approximation of a stratiform cloud. Moreover, since they are simple, it should be easier to understand how light transport is behaving. Finally, these shapes have a few DOF, which makes them accessible to study.

Since we want an understandable study that allows us to find a mesoscopic description of light transport, we chose to study light transport in simple shapes. We chose slabs for their maximum simplicity[1].

[1] Since spheres seem good approximations of parts of cumuliform clouds, our initial goal was to first study slabs for improving our initial comprehension of light transport, then study other shapes such as spheres, prism, cylinders, pairs

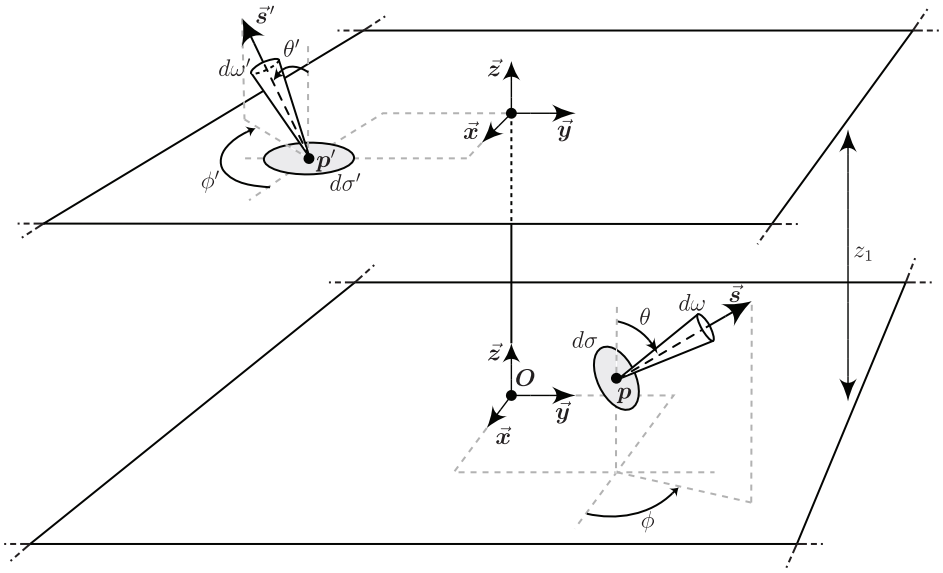


Figure 4.1: *How our BSSRDF is defined in a slab. The slab receives incident illumination from $\vec{s}'$ in every point $\mathbf{p}'$ on the top boundary. The BSSRDF $S(\mathbf{p}, \vec{s}, \mathbf{p}', \vec{s}')$ encodes light transport from any points $\mathbf{p}'$ on the top boundary and any upwards directions $\vec{s}'$ to any points $\mathbf{p}$ (*including points inside the slab*) and any direction $\vec{s}$.*

In the following, we will use the notions and notations for slabs introduced in §2.5.5. We use the reference frame as shown on Figure 2.27. The top of the slab is at $z = z_1$ while the bottom of the slab is at $z = 0$, with $z_1 > 0$ the thickness of the slab.

4.2.2 Available outputs for study

As we noted in §2.6.3.3, modeling light transport in clouds by a BSSRDF is an interesting idea. The mesoscopic representation we aim for could take the form of a BSSRDF linking the irradiance on the lit part of the slab and the radiance of a point within the slab boundaries. However, a BSSRDF is used to model sub-surface scattering, and a cloud does not have a surface per se. Indeed, a cloud is not a solid or liquid object where molecules are linked together, but a collection of liquid water droplets in air.

However, by definition, there is no need for an actual surface to exist in order to define radiance and irradiance. Even though these units are defined through an element of surface, this element of surface can be purely virtual. Since a BSSRDF is a ratio between a radiance and an irradiance, it can be defined between any two elements of surface, including ones that do not exist in reality. Indeed, a BSSRDF essentially describes a radiance transfer between two points and two directions. The reason it uses surface-based units such as radiance and irradiance instead of, say, intensities, is only for practical reasons [NRH+77]. Thus, we can use the BSSRDF paradigm to define light transport in clouds. This is one of the contributions of this thesis.

The question is now: where should we define these virtual elements of surface for clouds? In the case of a slab, the answer is somewhat obvious. The element of surface where irradiance is measured

of cylinders (to understand the behavior of creases), etc. Unfortunately, we lacked time for this second study (see §8.4). Moreover, each new model adds some DOF, which has dramatic consequences on the simulation costs.

should correspond to the lit boundary of the slab, *i.e.,* the top boundary of the slab in our case. Regarding the element of surface where radiance is measured, we chose to define it as the infinitesimal element of surface located at the measurement point $\boldsymbol{p}$ and normal to the direction of measurement $\vec{\boldsymbol{s}}$ (see Figure 4.1).

Let us write this BSSRDF as $S(\boldsymbol{p}, \vec{\boldsymbol{s}}, \boldsymbol{p}', \vec{\boldsymbol{s}}')$. We call $\boldsymbol{p}$ the *viewpoint* (as it represents the point where radiance is measured) and $\vec{\boldsymbol{s}}$ the *view direction*. The top boundary represents the *lit surface*, and $\vec{\boldsymbol{s}}'$ is called the *incident direction* of light. Let us list the degrees of freedom (*i.e.,* input parameters) of this BSSRDF for a slab. First, we know that $\boldsymbol{p}'$ is on the top surface, *i.e.,* $\boldsymbol{p}' = (x_{\boldsymbol{p}'}, y_{\boldsymbol{p}'}, z_1)$. Thus we can write S as $S(\boldsymbol{p}, \vec{\boldsymbol{s}}, x_{\boldsymbol{p}'}, y_{\boldsymbol{p}'}, \vec{\boldsymbol{s}}')$. Second, since the slab is infinite to its lateral extent and its boundary conditions are constant with respect to x and y, S is invariant with respect to translation, *i.e.,*

$$S(x_{\boldsymbol{p}}, y_{\boldsymbol{p}}, z_{\boldsymbol{p}}, \vec{\boldsymbol{s}}, x_{\boldsymbol{p}'}, y_{\boldsymbol{p}'}, \vec{\boldsymbol{s}}') = S(x_{\boldsymbol{p}} + x, \quad y_{\boldsymbol{p}} + y, \quad z_{\boldsymbol{p}}, \quad \vec{\boldsymbol{s}}, \quad x_{\boldsymbol{p}'} + x, \quad y_{\boldsymbol{p}'} + y, \quad \vec{\boldsymbol{s}}')$$

for any given x and y. As a result, S can be written as $S(z, \vec{\boldsymbol{s}}, x', y', \vec{\boldsymbol{s}}')$ with $x' = x_{\boldsymbol{p}'} - x_{\boldsymbol{p}}$, $y' = y_{\boldsymbol{p}'} - y_{\boldsymbol{p}}$, and $z = z_{\boldsymbol{p}}$. For convenience, we choose to align the reference frame with $\boldsymbol{p}$ so that $\boldsymbol{p} = (0, 0, z)$. Thus, $x' = x_{\boldsymbol{p}'}$ and $y' = y_{\boldsymbol{p}'}$. Finally, since the material of the slab (cloud droplets) is isotropic, we know that S is rotationally invariant and bilaterally symmetric. It can thus be written

$$S(z, \theta, x', y', \psi, \theta'), \tag{4.1}$$

with $\psi = |\phi' - \phi|$. For convenience, we choose to align the reference frame so that $\phi = 0$. Thus, $\psi = |\phi'|$.

As a result, this BSSRDF has six degrees of freedom:

- $0 \leq z \leq z_1$,
- $0 \leq \theta \leq \pi$,
- $-\infty < x' < \infty$,
- $-\infty < y' < \infty$,
- $0 \leq \psi \leq \pi$,
- $0 \leq \theta' \leq \pi$.

Note that for the particular case of $z = 0$ (*i.e.,* looking at the slab from underneath) we have $0 \leq \theta \leq \pi/2$, and for the particular case of $z = z_1$ (*i.e.,* looking at the slab from above) we have $\pi/2 \leq \theta \leq \pi$. We call z the *height* of the viewpoint in the slab. We also define the value $z' = z_1 - z$ as the *viewpoint depth*, *i.e.,* the distance from the lit surface to the viewpoint. As we will see in this study, the viewpoint depth has often more significance than the viewpoint height. In the rest of this section, we list the remaining available inputs to the study.

4.2.3 Available parameters of the study

In addition to the traditional input of a BSSRDF, there are other parameters that we can choose as additional degrees of freedom. In this section we list what are the available parameters and we explain which ones we pick and which ones we choose to ignore.

4.2.3.1 Wavelength of light

As we have seen in §2.5, a cloud droplet does not absorb any light in the visible spectrum, and its phase function display colors only in the peak corresponding to the glory. When looking at real clouds it seems that this peak only influences the glory, *i.e.,* its coloring probably does not affect the rest of the aspect of the cloud. Clouds seem to re-emit the colors they receive without changing them.

As a result, it does not seem of primary importance to us to study the coloring of light due to multiple scattering. In consequence, we do *not* take into account the wavelength of light in our study. In the following, we use optical parameters (*i.e.,* phase function, etc.) that correspond to the average visible spectrum and not a particular wavelength or channel.

4.2.3.2 Optical parameters

As we have seen in §2.5, there are two actual degrees of freedom in terms of optical properties for a slab: the optical thickness along the vertical axis τ and the phase function $P(\Theta)$. The optical thickness $\tau = z_1 \rho \sigma_e$ depends on the thickness z_1, the concentration ρ and the extinction cross-section σ_e. The extinction cross-section σ_e and the phase function $P(\Theta)$ depend on the DSD $n(r)$. We describe the DSD $n(r)$ through the modified Gamma distribution (Equation 2.2 in §2.4.2), as it has been shown to be a good model of the DSD observed in clouds. Thus, for the DSD the parameters we can act on are the two parameters γ (broadness) and r_n (characteristic radius) of the distribution. To summarize, the two DOF τ and $P(\Theta)$ are defined through the 4 parameters z_1, ρ, r_n and γ.

As it has been explained in previous works, one of the consequences of multiple scattering is that it blurs the multiply-scattered phase function (see §2.5.3.3). As a result, even if we would make the phase function vary (*e.g.,* by making the DSD vary), the consequences would probably be hardly visible except maybe for low orders of scattering. Nevertheless, we can also compute simulations with phase function models other than the reference Mie model to compare them against the original Mie phase function. In consequence, we chose two different models for study: the reference Mie model (referred to as "ref") and the "chopped-peak" model (referred to as "chopped", see §2.6.2).

chopped-peak model

In the reference model, we use the Mie phase function $P_{\text{ref}}(\Theta)$ and the extinction cross-section $\kappa_{\text{ref}} = \rho \sigma_e$ as described in the literature (see §2.5). In the "chopped-peak" model, we use a different phase function $P_{\text{chopped}}(\Theta)$ and a different extinction cross-section κ_{chopped}. As explained in §2.6.2, the chopped model roughly consists in removing the narrow peak of the Mie phase function and dividing the extinction coefficient of the cloud by 2. We describe more precisely how we compute $P_{\text{chopped}}(\Theta)$ and κ_{chopped} in Appendix A. It is essentially the same method as described in [Len85].

The only DOF left is the optical thickness τ. We choose to define it through the geometrical thickness z_1, leaving all the other parameters (γ, r_n, ρ) constant. We chose this parameter because it seemed the most meaningful to us. Note that it is not necessary to make the density ρ vary to study the influence of density on light transport. Indeed, since $\tau = z_1 \rho \sigma_e$, changing the density ρ is equivalent to changing the thickness z_1. As a result, studying light transport in slabs of various thicknesses will also allow us to study the light transport in slabs of various densities.

For our study to be meaningful, all these parameters should be in the range of observed values as described in Table 2.1. In order to be as close as possible to cumulus clouds, we used the following values, taken from literature [Dei64].

- $\gamma = 2$.

- $r_n = 0.75\,\mu\text{m}$.
- $\rho = 400\,\text{cm}^{-3}$.
- $0 \leq z_1 \leq 15\,\text{km}$.

This yields a mean free path l_0 of 23 m with the reference model and of 46 m with the chopped model.

Our BSSRDF described by Equation 4.1 becomes

$$S_{\text{model}}(z_1, z, \theta, x', y', \psi, \theta'), \tag{4.2}$$

where the subscript model indicates that we use either the reference model (model = ref) or the chopped-peak model (model = chopped).

4.2.3.3 Order of scattering

Another possible parameter for study is the order of scattering of the multiple scattering paths. Intuition tells us that the view-dependent effects come from low-orders paths while the view-independent effects come from high-order paths (from the light spreading paradigm explained in §2.5.3.3). As a result, we chose to study the contribution of each scattering order in our BSSRDF. Equation 4.2 becomes

$$S_{\text{model}}(n, z_1, z, \theta, x', y', \psi, \theta'), \tag{4.3}$$

with $1 \leq n < \infty$ the order of scattering. Note that here we *do not* consider 0-order scattering, *i.e.,* transparency. It can easily be computed by $e^{-\tau}$. We are only considering photons that have been scattered at least once in the cloud.

4.2.4 Summary

In §4.2.3 we have added three new DOF (slab thickness, scattering model and scattering order) to the BSSRDF described by Equation 4.1. As a result, what we are going to study in this chapter is the 9-DOF, order-dependent, bi-directional slab scattering distribution function represented as

$$S_{\text{model}}(n, z_1, z, \theta, x', y', \psi, \theta'), \tag{4.4}$$

in the range

- $1 \leq n < \infty$,
- model = {ref, chopped},
- $0 \leq z_1 \leq 15\,\text{km}$,
- $0 \leq z \leq z_1$,
- $0 \leq \theta \leq \pi$,
- $-\infty < x' < \infty$,
- $-\infty < y' < \infty$,
- $0 \leq \psi \leq \pi$,
- $0 \leq \theta' \leq \pi/2$.

Note that one degree of freedom is binary (model) and one is discrete (n), while the others are continuous. Note also that three DOF are unbounded (n, x' and y').

This function is not a BSSRDF in the original sense anymore. It is defined for a slab only, $\boldsymbol{p}$ is not restricted to the surface of the slab, and it depends on the order of scattering and the thickness of the slab. To avoid confusion, we call S a *bidirectional sub-surface scattering distribution function (BSSSDF)* (BSSSDF).

bidirectional sub-surface scattering distribution function

Since S is a high-dimensional function, difficult to understand, we derive some more meaningful lower-dimensional subsets of it in the following. We will give a meaning to all of these functions in the next section. The colors in the formulas are there to help the reader notice more easily the important part of each equation.

We write the BSSSDF S for a group of consecutive orders n to m as

$$S_{\text{model}}([n,m],z_1,z,\theta,x',y',\psi,\theta') = \sum_{i=n}^{m} S_{\text{model}}(i,z_1,z,\theta,x',y',\psi,\theta'). \tag{4.5}$$

We call

$$C_{\text{model}}(n,z_1,z,\theta,\psi,\theta') = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} S_{\text{model}}(n,z_1,z,\theta,x',y',\psi,\theta')dx'dy' \tag{4.6}$$

the *bidirectional scattering distribution function (BSDF)* (BSDF).

bidirectional scattering distribution function

We define the function D as

$$D_{\text{model}}(n,z_1,z,x',y',\theta,\psi,\theta') = \frac{S_{\text{model}}(n,z_1,z,\theta,x',y',\psi,\theta')}{C_{\text{model}}(n,z_1,z,\theta,\psi,\theta')}. \tag{4.7}$$

We call D the normalized *surface density distribution* of S. In other words, the BSDF C corresponds to the bidirectional distribution of S while D is the surface distribution of S.

surface density distribution

We write $\boldsymbol{c} = (x_{\boldsymbol{c}}, y_{\boldsymbol{c}}, 0)$ the first moment (*i.e.,* the mean) of D, and $\sigma = (\sigma_x, \sigma_y)$ the second moment (*i.e.,* the standard deviation) of D. That is,

$$\boldsymbol{c}_{\text{model}}(n,z_1,z,\theta,\psi,\theta') = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} D_{\text{model}}(n,z_1,z,x',y',\theta,\psi,\theta')\boldsymbol{p}'dx'dy', \tag{4.8}$$

$$\sigma^2_{x,\text{model}}(n,z_1,z,\theta,\psi,\theta') = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} D_{\text{model}}(n,z_1,z,x',y',\theta,\psi,\theta')(x'-x_{\boldsymbol{c}})^2dx'dy', \tag{4.9}$$

$$\sigma^2_{y,\text{model}}(n,z_1,z,\theta,\psi,\theta') = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} D_{\text{model}}(n,z_1,z,x',y',\theta,\psi,\theta')(y'-y_{\boldsymbol{c}})^2dx'dy'. \tag{4.10}$$

We call

$$R_{\text{model}}(n,z_1,\theta,\psi,\theta') = C_{\text{model}}(n,z_1,z_1,\theta,\psi,\theta'), \tag{4.11}$$

$$T_{\text{model}}(n,z_1,\theta,\psi,\theta') = C_{\text{model}}(n,z_1,0,\theta,\psi,\theta'), \tag{4.12}$$

the *reflectance* and *transmittance* of a slab, respectively. Note that R is a BRDF and T is a BTDF

reflectance

(see §1.5.9, §2.5.5.2).

transmittance

Finally, we call

$$M_{R,\text{model}}(n, z_1, \theta') = \frac{1}{2\pi}\int_0^{2\pi}\int_0^{\pi/2} R_{\text{model}}(n, z_1, \theta, \psi, \theta')\sin\theta\cos\theta d\theta d\psi \tag{4.13}$$

$$M_{T,\text{model}}(n, z_1, \theta') = \frac{1}{2\pi}\int_0^{2\pi}\int_{\pi/2}^{\pi} T_{\text{model}}(n, z_1, \theta, \psi, \theta')\sin\theta\cos\theta d\theta d\psi \tag{4.14}$$

reflected exitance

the *reflected exitance* and *transmitted exitance* of a slab, respectively (see §1.5.8, §2.5.5.6).

transmitted exitance

All these values are order-dependent. We also define the order-independent equivalents as

$$S_{\text{model}}(z_1, z, \theta, x', y', \psi, \theta') = S_{\text{model}}([1,\infty], z_1, z, \theta, x', y', \psi, \theta'), \tag{4.15}$$

$$C_{\text{model}}(z_1, z, \theta, \psi, \theta') = C_{\text{model}}([1,\infty], z_1, z, \theta, \psi, \theta'), \tag{4.16}$$

$$D_{\text{model}}(z_1, z, \theta, x', y', \psi, \theta') = D_{\text{model}}([1,\infty], z_1, z, \theta, x', y', \psi, \theta'), \tag{4.17}$$

$$\boldsymbol{c}_{\text{model}}(z_1, z, \theta, \psi, \theta') = \boldsymbol{c}_{\text{model}}([1,\infty], z_1, z, \theta, \psi, \theta'), \tag{4.18}$$

$$\sigma_{x,\text{model}}(z_1, z, \theta, \psi, \theta') = \sigma_{x,\text{model}}([1,\infty], z_1, z, \theta, \psi, \theta'), \tag{4.19}$$

$$\sigma_{y,\text{model}}(z_1, z, \theta, \psi, \theta') = \sigma_{y,\text{model}}([1,\infty], z_1, z, \theta, \psi, \theta'), \tag{4.20}$$

$$R_{\text{model}}(z_1, \theta, \psi, \theta') = R_{\text{model}}([1,\infty], z_1, \theta, \psi, \theta'), \tag{4.21}$$

$$T_{\text{model}}(z_1, \theta, \psi, \theta') = T_{\text{model}}([1,\infty], z_1, \theta, \psi, \theta'), \tag{4.22}$$

$$M_{R,\text{model}}(z_1, \theta') = M_{R,\text{model}}([1,\infty], z_1, \theta'), \tag{4.23}$$

$$M_{T,\text{model}}(z_1, \theta') = M_{T,\text{model}}([1,\infty], z_1, \theta'). \tag{4.24}$$

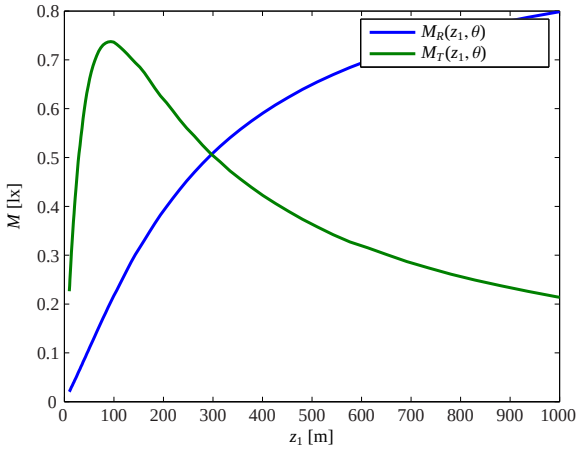
We give a meaning to all these functions in the next section, with an explanation on how they can help us.

4.2.5 Tools for analysis

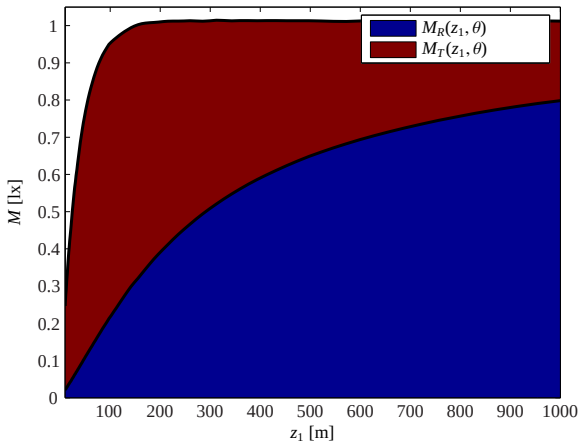
We have defined the function to analyze (Equation 4.4) with its 9 input parameters. The experimental part of the study consists in sampling massively the 9-dimensional input space V and computing S for these input values. We can then analyze it by visualizing the results. Since S is 9-dimensional, it cannot be directly visualized. We can, however, plot it in some subset of V. We list here the plots we have found useful. We begin with the lowest dimensional, order-independent expressions. We also explain the meaning of these expressions and what their analysis can tell us. In the following examples the model used is the chopped model, unless it is explicitly specified.

4.2.5.1 Reflected and transmitted exitances

The reflected exitance M_R (Equation 4.23) and the transmitted exitance M_T (Equation 4.24) represent the flux exiting the slab from top and bottom in all directions due to scattering when illuminated with



(a) Example plot for the reflected and transmitted exitances M_R and M_T against slab thickness z_1. $\theta' = 0°$.



(b) Example area plot for the reflected and transmitted exitances M_R and M_T against slab thickness z_1. $\theta' = 0°$. Each of the areas represents one exitance. The sum of these areas (black outline) represents $M_R + M_T$.

Figure 4.2: *Example exitance plots.*

a unit illuminance of incident angle θ'. Since the cloud is not absorbing visible light, we know that all the light incident to it is exiting. That is,

$$M_R(z_1, \theta') + M_T(z_1, \theta') + e^{-\frac{\tau_1}{\cos\theta'}} = 1. \tag{4.25}$$

The third term in Equation 4.25 is the exitance traversing the slab without being scattered (*i.e.,* due to transparency), since we only account for scattered light in our representations.

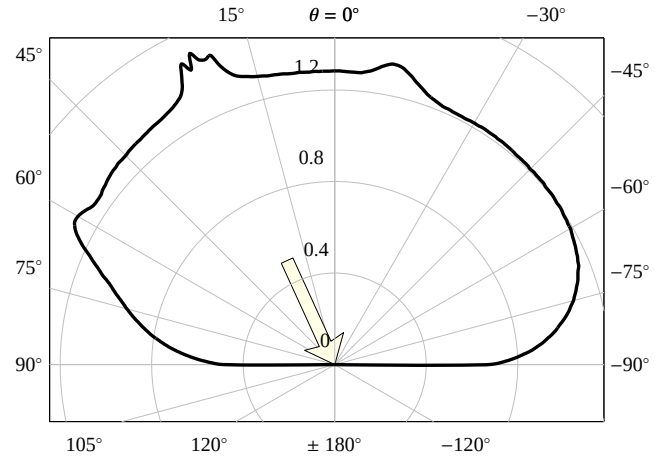
From the literature, we know that $M_R(0, \theta') = 0$ (*i.e.,* when there is no matter to scatter light, it cannot be reflected) and that $\lim_{z_1 \to \infty} M_R(z_1, \theta') = 1$. According to our observations, clouds are generally brighter on their lit side (top) than on their unlit side (base). This should translate to $M_R > M_T$. To verify this, we can plot $M_R(z_1, \theta')$ and $M_T(z_1, \theta')$ against z_1 (keeping θ' constant) and see in which conditions we have $M_R > M_T$.

Figure 4.2(a) shows an example of such a diagram. Both exitances can also be represented on an area plot, *i.e.,* plotting M_R and M_T against z_1 and stacked on top of each other, as shown on Figure 4.2(b). This type of plot allows us to visualize the contribution of each exitance to the total exiting exitance and to validate our simulations by verifying that this total exitance is equal to one.

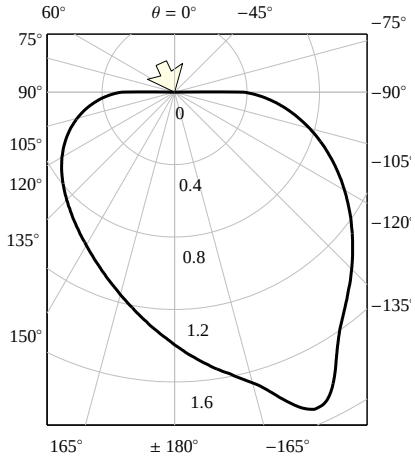
4.2.5.2 Reflectance and transmittance - BRDF and BTDF

The exitances M_R and M_T described in §4.2.5.1 give us the total flux exiting through the boundaries of the cloud but they do not give us its angular distribution. The reflectance R (Equation 4.21) and transmittance T (Equation 4.22) give us a directional distribution.

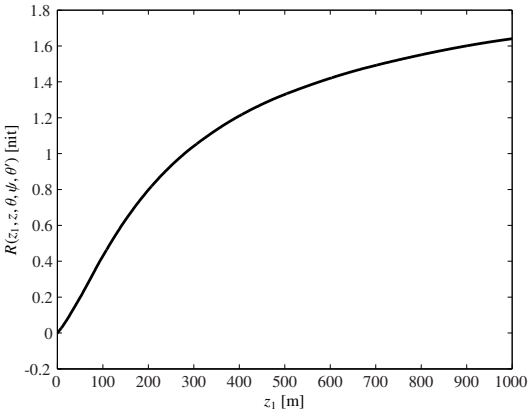
As mentioned above, R is a BRDF and can thus be plotted in the classical way as explained in §1.5.9, *i.e.,* plotting $R(z_1, \theta, \psi, \theta')$ against θ in polar coordinates with z_1, ψ and θ' constant (see Figure 4.3(a)). T is a BTDF and can essentially be plotted like R, except that $\pi/2 \leq \theta \leq 3\pi/2$ for T (see Figure 4.3(b)).



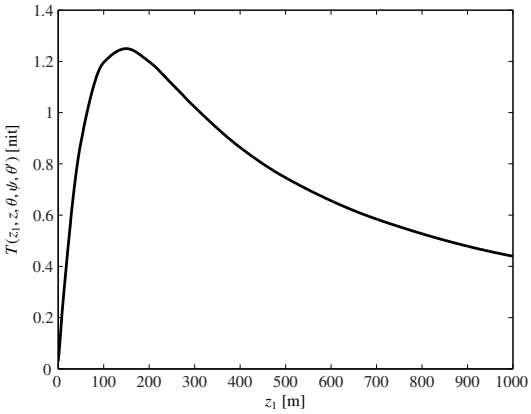
(a) Example plot for the BRDF R against view angle θ. $z_1 = 500\,\text{m}, \psi = 0°, \theta' = 25°$.



(b) Example plot for the BTDF T against view angle θ. $z_1 = 200\,\text{m}, \psi = 0°, \theta' = 25°$.



(c) Example plot for the BRDF R against z_1. $\theta = 45°, \psi = 0°, \theta' = 0°$.



(d) Example plot for the BTDF T against z_1. $\theta = 135°, \psi = 0°, \theta' = 0°$.

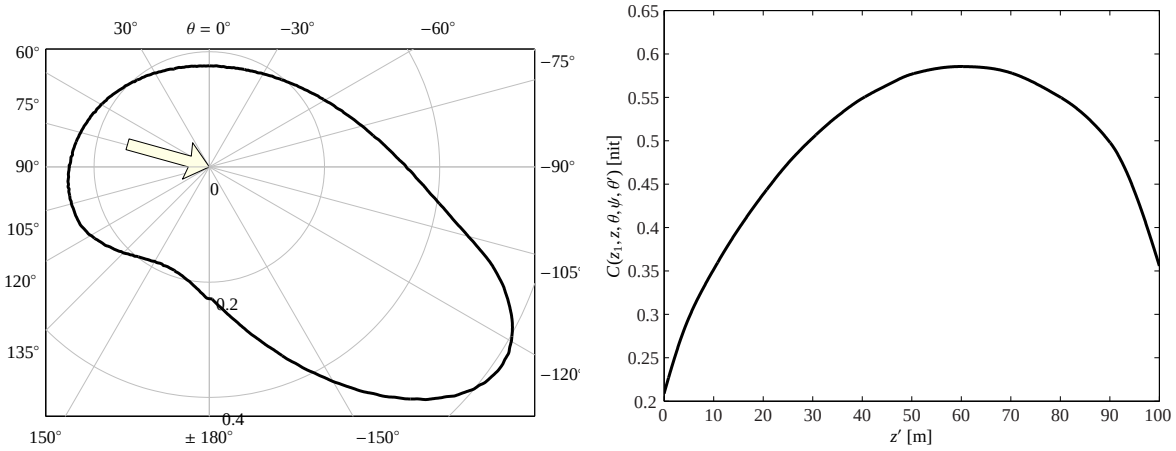
Figure 4.3: Top: *Example BRDF and BTDF plots. The arrow indicates the incident light direction on the top boundary of the slab.* Bottom: *Example plots of the reflectance R and the transmittance T against the slab thickness* z_1.

This will allow us to see the amount of anisotropy in R and T. Strong anisotropy means that there are view-dependent effects, whereas low anisotropy means no view-dependent effects. According to our observations, there are both view-dependent and view-independent effects in clouds. Analyzing R and T should help us find the conditions for these effects to be present.

We can also plot R and T against the slab thickness z_1 (see Figure 4.3). It should essentially look like the plots of M_R and M_T against z_1, except in this case it is direction-dependent, *i.e.,* it represents the radiance exiting in a specific direction instead of exiting in all directions.

4.2.5.3 BSDF

The BRDF R and BTDF T described in §4.2.5.2 are special cases of the BSDF C (Equation 4.16) which is defined not only for the boundaries but also for any point inside the slab.



(a) Example plot for the BSDF C against view angle θ. $z_1 = 500\,\text{m}, z = 200\,\text{m}, \psi = 0°, \theta' = 75°$. The arrow indicates the incident light direction on the top boundary of the slab

(b) Example plot for the BSDF C against viewpoint depth z' $z_1 = 100\,\text{m}, \theta = 90°, \psi = 0°, \theta' = 0°$.

Figure 4.4: *Example BSDF plots.*

As a result, it can be plotted just like R or T and for $0 \leq \theta \leq 2\pi$), and used in the same way to analyze the anisotropy of radiance distribution including at viewpoints inside the slab (see Figure 4.4(a)).

Second, we can plot C against z, while keeping the other parameters constant (see Figure 4.4(b)). This kind of plot shows the evolution of radiance as the point of view is moving up or down in the slab while the view and light directions stay constant. This kind of plot is similar to that of the temperature profile in a heat conduction system. It is also possible to plot C against z_1, although it is not exactly certain how these plots might help us.

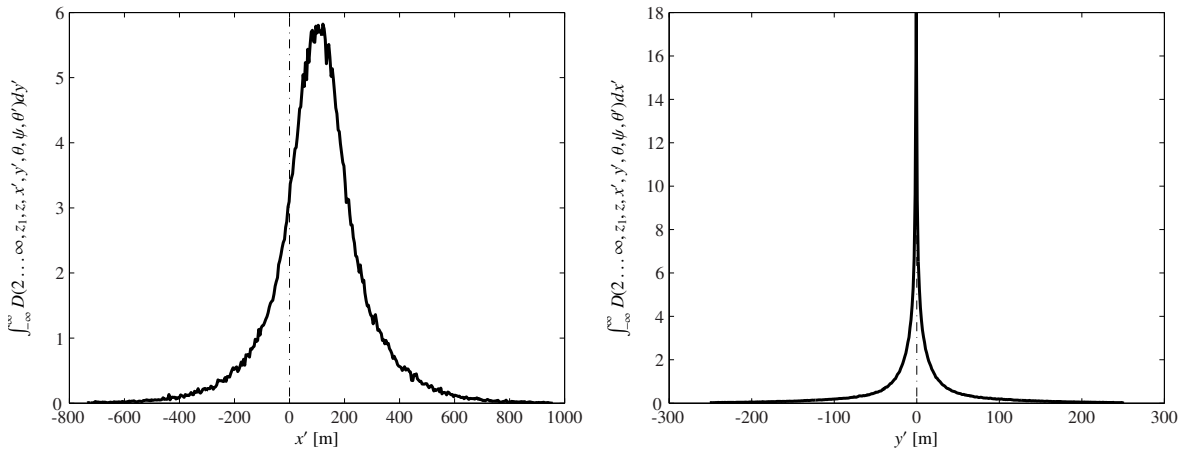
This function has several useful purposes. First, it might help us understand how light is behaving inside the slab instead of only seeing the end result of light exiting the slab. Second, it should enable us to account for point of views that are inside a cloud. Last, but not least, it may help us find formulation in clouds of arbitrary shape. Indeed, our light transport model of Chapter 7 proposes a formulation in which light transport in a given cloud shape can be well approximated by light transport in a slab with a point of view generally inside the slab.

4.2.5.4 Spatial distribution

The spatial distribution D (Equation 4.17) represents the amount of light a path will contribute to at $\boldsymbol{p}$ in direction $\vec{s}$ when entering on the top slab boundary with a direction $\vec{s}'$ in $\boldsymbol{p}' = (x', y', z_1)$. In other words, it represents the probability for a light path to enter at coordinates (x', y') on the slab surface, other parameters being equal.

For better readability, we plot D against one variable only, either x' or y'. That is, we plot either $\int_{-\infty}^{\infty} D(z_1, z, x', y', \theta, \psi, \theta')dy'$ against x' (Figure 4.5(a)) or $\int_{-\infty}^{\infty} D(z_1, z, x', y', \theta, \psi, \theta')dx'$ against y' (Figure 4.5(b)). Figure 4.5 shows examples of these plots.

The shape of this distribution will allow us to examine the "locality" of light transport. If D is wide or uniform, it means that light arriving at a point comes from anywhere on the surface of the slab. In



(a) Plot of D against x'. $z_1 = 200\,\text{m}, z = 0\,\text{m}, \theta = 45°, \psi = 90°, \theta' = 67°$. The high frequencies are due to noise from the Monte-Carlo computations.

(b) Plot of D against y'. $z_1 = 100\,\text{m}, z = z_1, \theta = 0°, \psi = 0°, \theta' = 0°$.

Figure 4.5: *Example plot for the spatial distribution D against entry point position $\boldsymbol{p}' = (x', y', z_1)$. The dash-dot line is at $x' = y' = 0\,\text{m}$, i.e., at the vertical of the viewpoint $\boldsymbol{p}$.*

that case, it means the most probable path approach [PAS03] is useless, since all paths are equally probable. On the other hand, if that distribution is peaked or narrow, it implies that light transport in a slab is a local process (*i.e.,* only a small part of the slab boundary near $\boldsymbol{p}$ is involved in light transport) and that the most probable path approach makes sense.

4.2.5.5 Moments of the spatial distribution

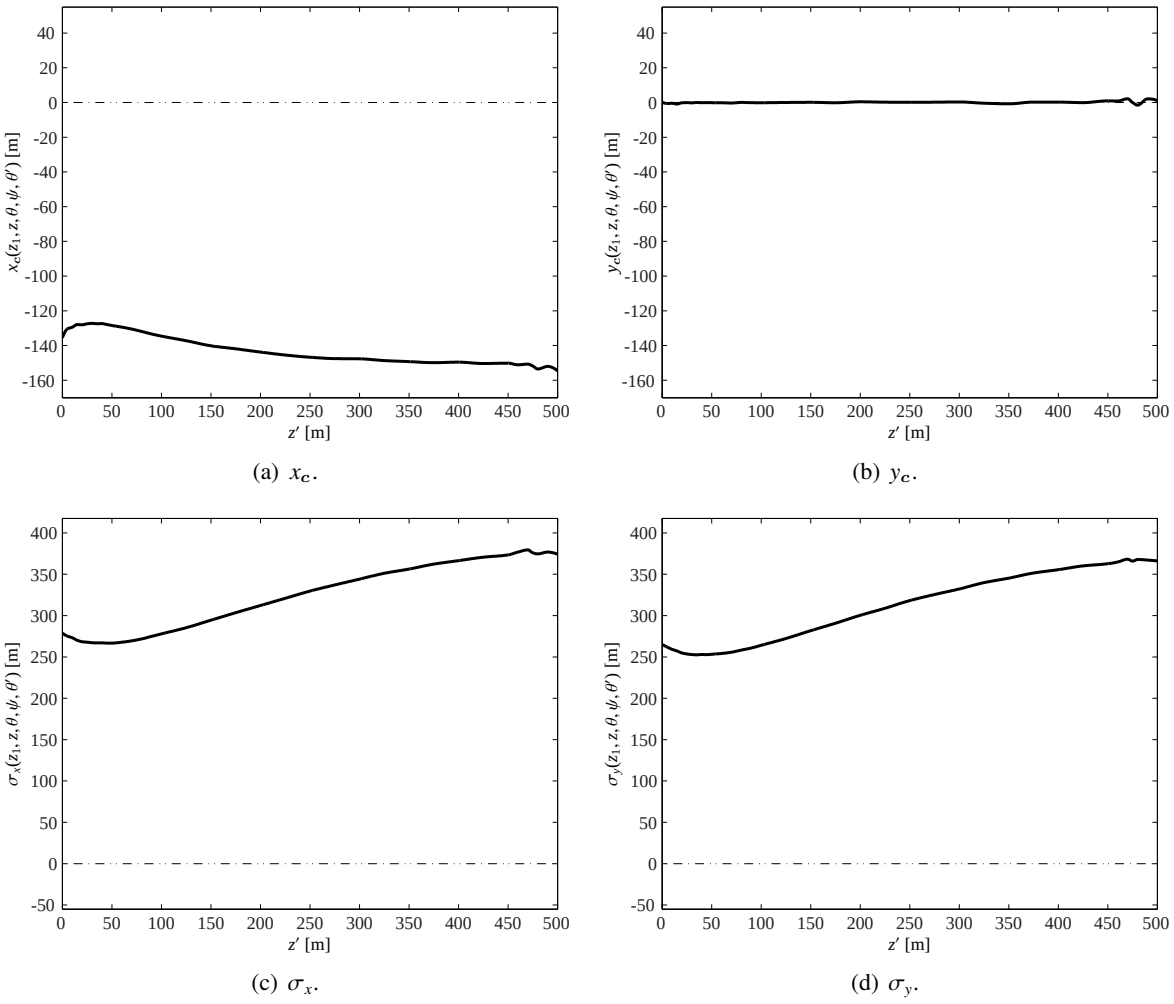
most probable entry point

The spatial distribution D can be characterized by its first two moments $\boldsymbol{c}$ (Equation 4.18) and (σ_x, σ_y) (Equations 4.19, 4.20). The first moment $\boldsymbol{c}$ is the mean of D. It represents the *most probable entry point* of light paths on the surface of the slab. That is, if we consider a point of view in a slab of thickness z_1, at a height z inside that slab, looking in direction $\vec{s}$ while the slab is illuminated from direction $\vec{s}'$, then light paths arriving at the point of view will most likely enter in $\boldsymbol{c}$ or in a neighborhood around it. It is exactly the entry point of the most probable path. Thus, if the most probable path approach is valid, the analysis of this function will give us valuable information on the entry point of the most probable path.

The second moment (σ_x, σ_y) of D gives an idea of the size of this neighborhood[2]. Since the second moment is the standard deviation of D, it gives an idea of the breadth of the distribution D. The larger (σ_x, σ_y), the more D will be spread and, as we wrote in §4.2.5.4, the less local light transport will be.

These moments (which have the same parameters as the BSDF C) can be plotted in various ways. Figure 4.6 shows plots of $\boldsymbol{c}$ and (σ_x, σ_y) against the viewpoint depth z'.

[2] Note that the second moment of a 2D function is actually a 2D tensor. Using only two scalar values as we do is already a slight approximation.



(a) x_c. (b) y_c.

(c) σ_x. (d) σ_y.

Figure 4.6: *Example plots for the first two moments of the spatial distribution against z'. The dashed line is at $x' = y' = 0$ m. For $\mathbf{c}$, it represents the location of $\mathbf{p}$ (while $\mathbf{c}$ represents the location of the most probable entry point $\mathbf{p}'$). $z_1 = 500, \theta = 90°, \psi = 0°, \theta' = 0°$. The high frequencies are due to noise from the Monte-Carlo computations.*

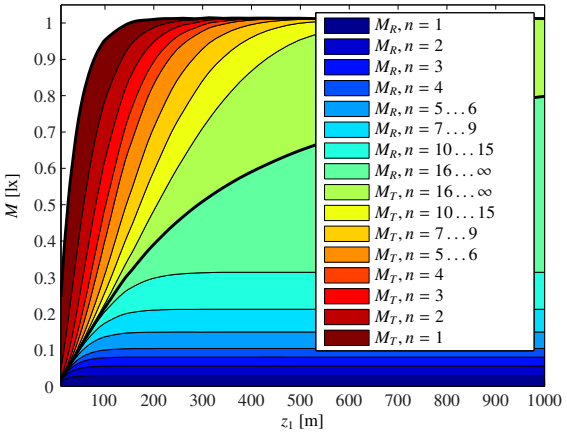
4.2.5.6 Order-dependence

All of the plots we presented above were *order-independent*, *i.e.,* showing the effect of multiple scattering taking into account paths of all orders. We can also analyze the *order-dependent* corresponding functions, *i.e.,* consider separately the contribution of each order of scattering to the final multiple scattering result.

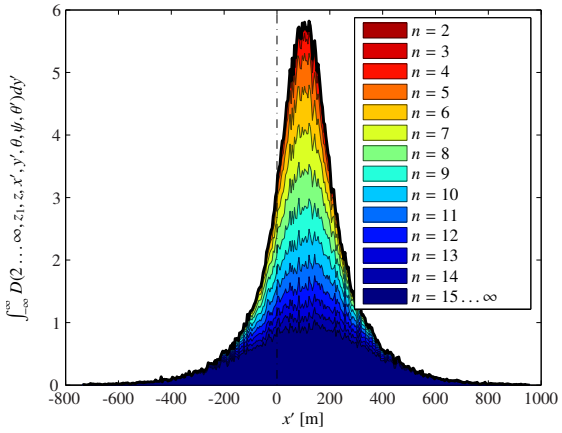
order-independent

order-dependent

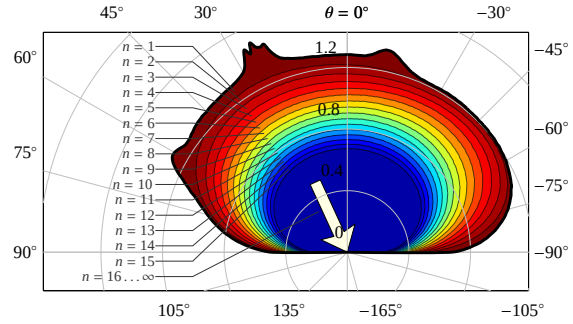
One way of viewing these order-dependent plots is to trace area plots, as shown on Figure 4.7 page 124. In these plots, each area represents the contribution of an order (or group of orders) of scattering. When stacking these areas on top of each other, the outline of the result is the order-independent plot. This kind of plot allows us to see the relative contribution of various orders of scattering to the total result. Note that it make no physical sense to trace area plots for the moments of the spatial distribution (contrary to the other values, the order-independent most probable entry point is not the sum of the order-dependent most probable entry points, but a weighted average of those).



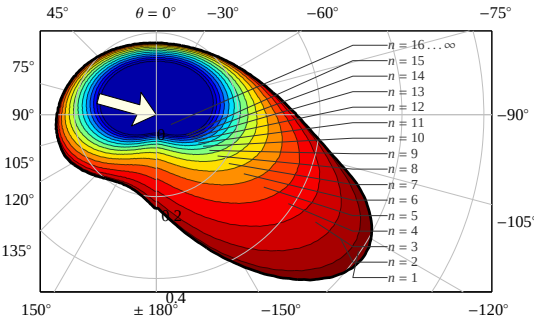
(a) Example area plot of the exitances M_R and M_T. This plot is the order-dependent version of Figure 4.2(b).



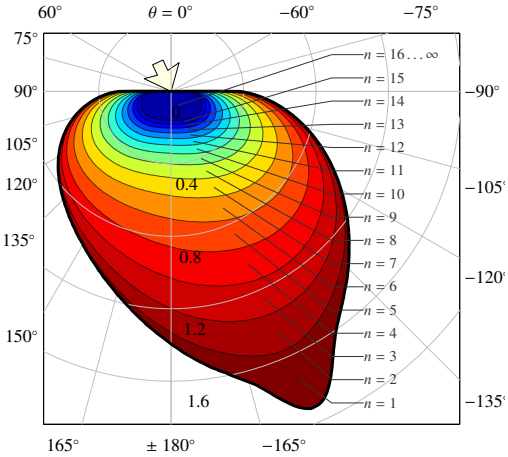
(b) Example area plot of the spatial distribution D. This plot is the order-dependent version of Figure 4.5(b). The high frequencies are due to noise from the Monte-Carlo computations.



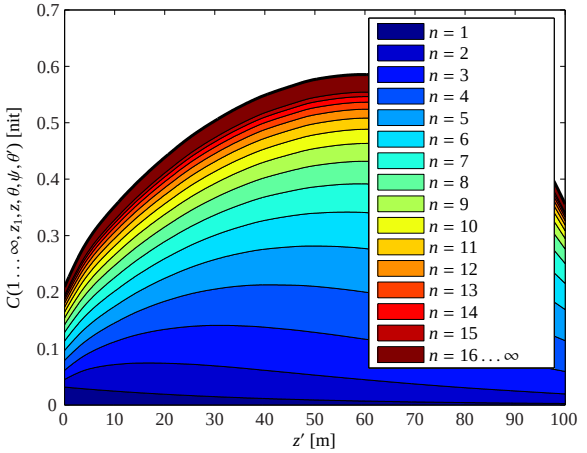
(c) Example area plot of the BRDF R. This plot is the order-dependent version of Figure 4.3(a).



(d) Example area plot of the BSDF C against θ. This plot is the order-dependent version of Figure 4.4(a).

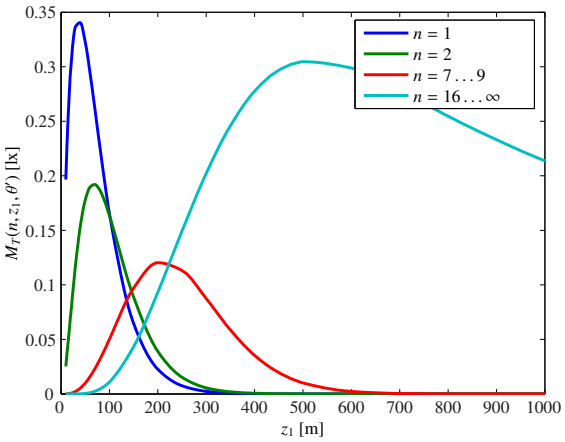


(e) Example area plot of the BTDF T. This plot is the order-dependent version of Figure 4.3(b).

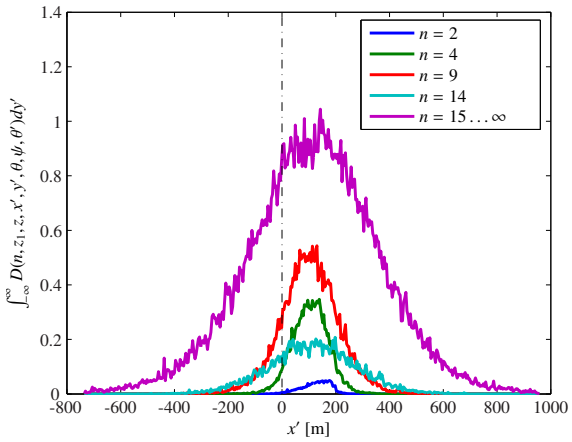


(f) Example area plot of the BSDF C against z'. This plot is the order-dependent version of Figure 4.4(b).

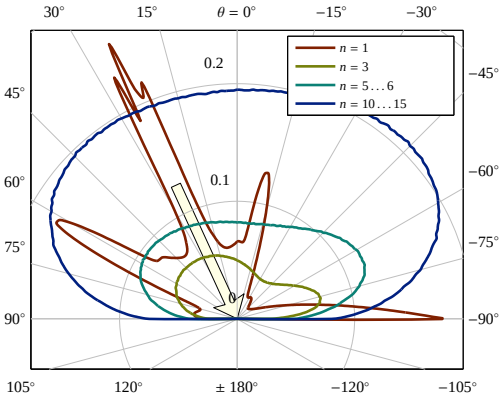
Figure 4.7: *Example area plots of order-dependent functions. Figure 4.8 page 125 display the counterpart plots for given orders of scattering.*



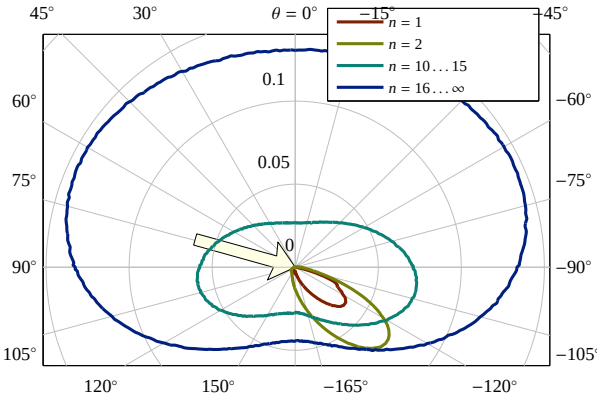
(a) Example order-dependent plot of the reflected exitance M_T against z_1. This plot is the order-dependent version of Figure 4.7(a).



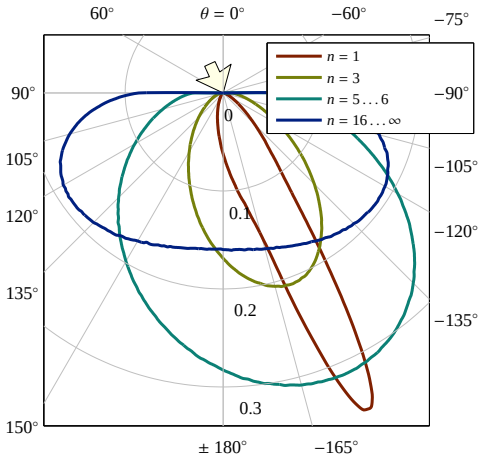
(b) Example order-dependent plot of the spatial distribution D. This plot is the order-dependent version of Figure 4.7(b). The high frequencies are due to noise from the Monte-Carlo computations.



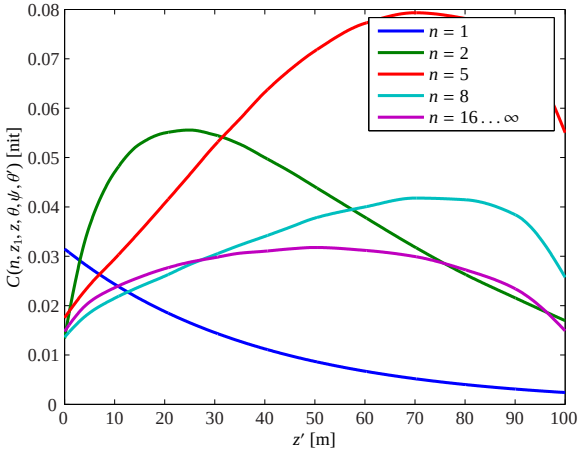
(c) Example order-dependent plot of the BRDF R. This plot is the order-dependent version of Figure 4.7(c).



(d) Example order-dependent plot of the BSDF C against θ. This plot is the order-dependent version of Figure 4.7(d).



(e) Example order-dependent plot of the BTDF T. This plot is the order-dependent version of Figure 4.7(e).



(f) Example order-dependent of the BSDF C against z'. This plot is the order-dependent version of Figure 4.7(f).

Figure 4.8: *Example plots of order-dependent functions. Figure 4.7 page 124 display the counterpart area plots.*

Another way of viewing the order-independent functions is to simply plot them separately for given orders, as shown on Figure 4.8 page 125. This kind of plot allows us to see the individual behavior of the functions for each orders, and to see the relative importance of each of them.

In the study of exitance, analyzing the relative importance of each order will allow us to understand which orders are significant in light transport and which orders are negligible. This can be used as an important information to approximate accurately and efficiently light transport. Indeed, various approximations relying on low-order scattering have been used in the past [Max86, NND96, DKY+00, HL01]. These approximations can only be valid if it can be shown that higher orders are negligible. This study will enable us to tell whether this is the case.

In the study of reflectance, transmittance and BSDF, an order-dependent analysis will allow us to understand which orders are more or less anisotropic. This should help us understand what causes the view-dependent features we observe in clouds.

In the study of spatial distribution and its moments, an order-dependent analysis will allow us to understand which orders are more or less local.

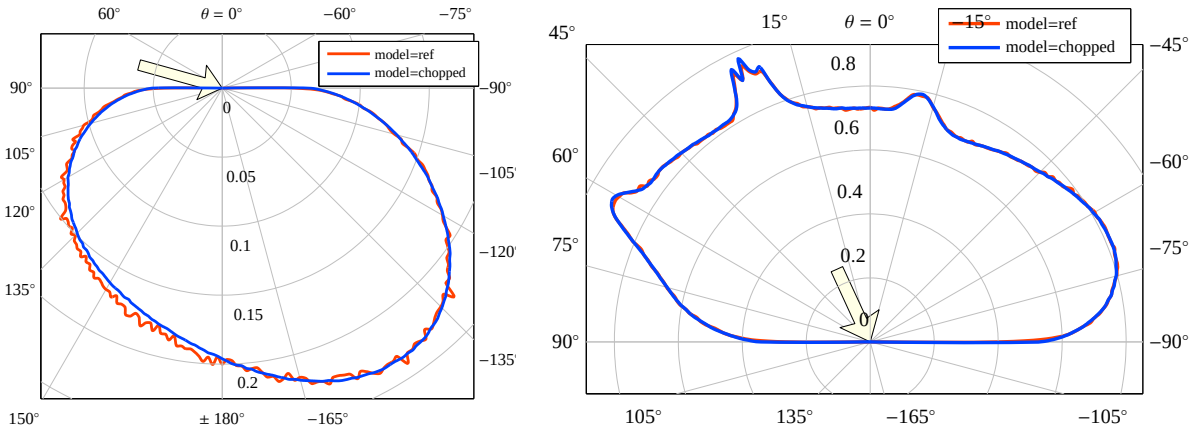
As a result, all these analyses should allow us to understand which kinds of paths cause the visual features we see in clouds. It is also interesting to analyze whether the behaviors of different orders are similar or not. If they are, we might be able to find a formulation of order-dependent light transport that suits any given order of scattering.

4.2.5.7 Model comparison

Comparing all the previously described functions with the two chosen models (see Figure 4.9 will allow us to see whether the chopped model is a good approximation of the reference model or not. If it is, it may help us gain in efficiency.

Summary of §4.2 : A slab shape is the simplest configuration on which we can study the behavior of light transport. While it is very different from a real cloud, it is simple enough to let us to understand the mesoscopic behavior of light transport. In this study, we will analyze light transport in a slab of cloud by observing the behavior of functions such as the exitance, the reflectance or the spatial distribution of entry point of light paths with respect to their input values such as the slab thickness and the order of scattering. These analyses will help us understand the causes of the visual effects we see in clouds. Moreover, they will allow us to find a phenomenological formulation of light transport in a slab in Chapter 5.

Une forme de dalle est la configuration la plus simple sur laquelle nous puissions étudier le comportement du transport lumineux. Bien que cette forme soit très différente de celle d'un vrai nuage, sa simplicité nous permet de comprendre le comportement mésoscopique du transport de la lumière. Dans cette étude, nous allons analyser le transport lumineux dans une dalle en observant le comportement de fonctions telles que l'exitance, la réflectance ou la distribution spatiale des points d'entrée des chemins lumineux en fonction des paramètres d'entrées tels que l'épaisseur de la dalle ou l'ordre de diffusion. Ces analyses nous permettront de comprendre les origines des effets visuels que nous pouvons voir dans les nuages. De plus, elles nous permettront de trouver une formulation phénoménologique du transport de la lumière dans une dalle au chapitre 5.



(a) Comparison of the BTDF T against θ between the chopped and the reference model. $z_1 = 200\,\mathrm{m}, \psi = 0°, \theta' = 75°$. The high frequencies are due to noise from the Monte-Carlo computations.

(b) Comparison of the BRDF R against θ between the chopped and the reference model. The results are almost undistinguishable. $z_1 = 200\,\mathrm{m}, \psi = 0°, \theta' = 25°$.

Figure 4.9: *Example comparison of results between the reference model and the chopped model.*

4.3 Implementation of our simulations

To compute the simulations needed by our study, we use the basic Monte-Carlo path-tracing algorithm (§2.6.3.1) that we adapt to our needs. It is a parallel implementation optimized to compute all the functions (R, T, C, $\boldsymbol{c}$, etc.) needed by our study for numerous cases.

4.3.1 Classical Monte-Carlo path tracing

Monte-Carlo path tracing consists in tracing numerous paths from the eye to the light source inside the volume [Jen01]. Each path i represents a possible paths that light can take to reach the viewpoint and corresponds to a luminance value $L[i]$. These paths are computed by a stochastic process. If a sufficient number N of paths have been traced (see §3.2), the average of their radiance is a good approximation of the radiance, *i.e.,*

$$L(\boldsymbol{p}, \vec{\boldsymbol{s}}) \approx \frac{1}{N} \sum_{i=1}^{N} L[i].$$

The classical Monte-Carlo path-tracing algorithm to compute the mean radiance L at $\boldsymbol{p}$ in direction $\vec{\boldsymbol{s}}$ in a slab illuminated from direction $\vec{\boldsymbol{s}}'$ is described in Algorithm 4.1. The output L of this algorithm is an approximation of the radiance, *i.e.,* $L(\boldsymbol{p}, \vec{\boldsymbol{s}}) \approx L$.

4.3.2 Parallel Monte-Carlo path tracer

We modify the classical algorithm to also compute the entry points of light paths and their distribution and to compute all order-dependent functions. We also do these computations for multiple incident directions at once. This allows us to calculate all function described in §4.2.5. It is described in Algorithm 4.2. The following explains how it works (see Figure 4.10).

Algorithm 4.1 Monte Carlo ray-tracing algorithm to compute the mean radiance L arriving at the viewpoint $\boldsymbol{p}$ in the view direction $\vec{s}$ in a slab of thickness z_1 with incident light direction $\vec{s}'$. $\mathcal{L}$ is a random variable whose PDF is the extinction function β. $\vec{\boldsymbol{V}}$ is a random variable defined so that the PDF of $(\vec{v} \cdot \vec{\boldsymbol{V}})$ is the phase function $P(\cos\Theta)$.

$L \leftarrow 0$
for $i = 1 \ldots N$ **do**
  $\boldsymbol{x} \leftarrow \boldsymbol{p}$
  $\vec{v} \leftarrow \vec{s}$
  $L[i] \leftarrow 0$
  while $0 \leq z_{\boldsymbol{x}} \leq z_1$ **do**
    $l \leftarrow \mathcal{L}$
    $\boldsymbol{x} \leftarrow \boldsymbol{x} + l\vec{v}$
    $d \leftarrow \dfrac{z_1 - z_{\boldsymbol{x}}}{\vec{s}' \cdot \vec{n}}$
    $L[i] \leftarrow L[i] + \beta(d)P(\vec{v} \cdot \vec{s}')$
    $\vec{v} \leftarrow \vec{\boldsymbol{V}}$
  end while
  $L \leftarrow L + L[i]$
end for
$L \leftarrow \frac{1}{N}L$

We consider M incident directions $\vec{s}_1{}', \ldots, \vec{s}_M{}'$. We trace N paths from the viewpoint $\boldsymbol{p}$ in the view direction $\vec{s}$. Each path i consists in a succession of free paths and scattering events. The length l of each free path is determined stochastically by using the extinction function β as the probability density function. The phase angle Θ of each scattering events is determined stochastically using the phase function $P(\Theta)$ as the probability density function.

At each scattering location $\boldsymbol{x}[i][o]$ corresponding to each order of scatter o, we compute the radiance $L[j][i][o]$ due to this order of scattering for this path for each incident direction $\vec{s}_j{}'$. This is done by tracing a ray from $\boldsymbol{x}[i][o]$ to the lit boundary of the slab in direction $\vec{s}_j{}'$. The intersection of this ray with the boundary is written $\boldsymbol{c}[j][i][o] = (x[j][i][o], y[j][i][o], z_1)$ and is the *entry point* of light for the order o of the path i and for the incident direction $\vec{s}_j{}'$.

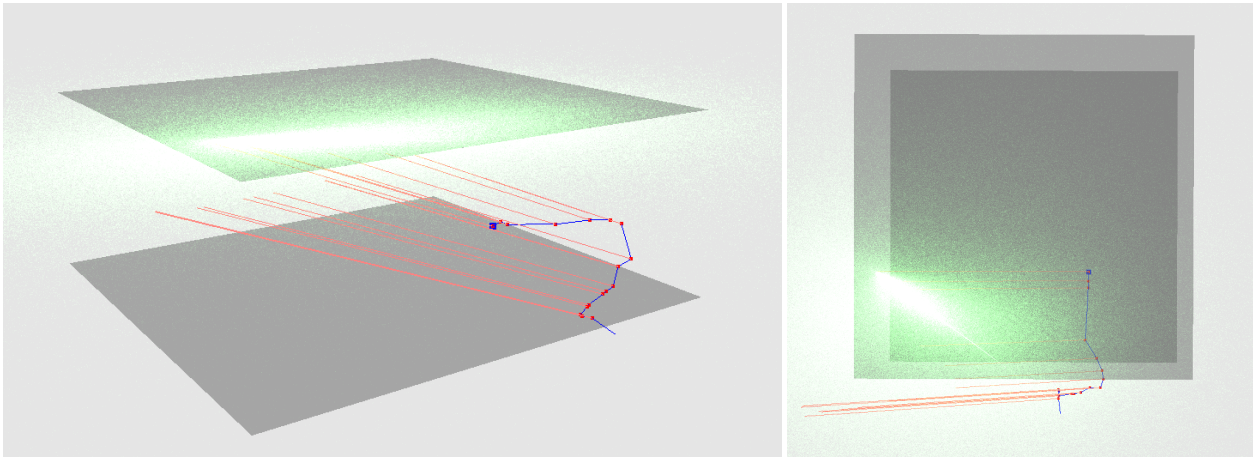
We write $L[j][o]$ the mean radiance due to paths of order o for the incident direction $\vec{s}_j{}'$ calculated by averaging of the radiance of each path

$$L[j][o] = \frac{1}{N}\sum_{i=1}^{N} L[j][i][o]. \tag{4.26}$$

The mean entry point $\boldsymbol{c}[j][o] = (x[j][o],\ y[j][o],\ z_1)$ of light paths of order o for the incident direction $\vec{s}_j{}'$ is the weighted average of the entry point of order o for all light paths i, *i.e.,*

$$x[j][o] = \frac{1}{L[j][o]}\sum_{i=1}^{N} x[j][i][o]\ L[j][i][o], \tag{4.27}$$

$$y[j][o] = \frac{1}{L[j][o]}\sum_{i=1}^{N} y[j][i][o]\ L[j][i][o]. \tag{4.28}$$



(a) (b)

Figure 4.10: *Screenshots of our path tracer. Starting from the viewpoint (blue dot), we trace a path (blue path) through the volume. At each scattering location (red dots) on this path, we shoot rays (red lines) in the incident direction . The intersection of these rays through the top boundary are the entry points of light paths (green dots). The green blotch represents the spatial distribution D of these entry points. In this example,* $z_1 = 300\,\text{m}, z = 100\,\text{m}, \theta = 120°, \psi = 90°, \theta' = 67°$.

Similarly, the variance $(\sigma_x[j][o],\ \sigma_y[j][o])$ of the entry points of order o is

$$\begin{aligned}
\sigma_x^2[j][o] &= \frac{1}{L[j][o]} \sum_{o=1}^{\infty} (x[j][i][o] - x[j][o])^2 L[i][o] \\
&= \frac{1}{L[j][o]} \left[\sum_{o=1}^{\infty} (x[j][i][o])^2\ L[j][i][o] - \left(\sum_{o=1}^{\infty} x[j][i][o]\ L[j][i][o] \right)^2 \right], \qquad (4.29) \\
\sigma_y^2[j][o] &= \frac{1}{L[j][o]} \left[\sum_{o=1}^{\infty} (y[j][i][o])^2\ L[j][i][o] - \left(\sum_{o=1}^{\infty} y[j][i][o]\ L[j][i][o] \right)^2 \right]. \qquad (4.30)
\end{aligned}$$

This algorithm is highly parallelizable. The tracing of one path (lines 8 – 28) is an independent operation. We implemented this algorithm on a supercomputer. Each node of the cluster traces numerous paths (lines 8 – 28). A master node gathers the results an performs the final computation (lines 30 – 38). The average time complexity of the serial algorithm $O(MNX)$ where M is the number of incident directions, N the number of paths traced and X is the average order of scattering of light paths. The network complexity of the parallel algorithm is $O(MX)$. Thus the total time complexity of our parallel algorithm is $O\left(\frac{MNX}{K}\right)$ where K is the number of nodes in the cluster. Note that this implies that the computation time increases as the average order of scattering of light paths increases. Since this average order is greater in thick slabs, as we will see in §4.4, it means computing Monte-Carlo light transport is more computationally expensive in thick slabs than in thin slabs.

Algorithm 4.2 Our parallel Monte-Carlo path tracing algorithm.

for $j = 1 \dots M$ **do**
  for $o = 1 \dots O$ **do**
    $L[j][o] \leftarrow 0, \quad x[j][o] \leftarrow 0, \quad y[j][o] \leftarrow 0$
    $XL2[j][o] \leftarrow 0, \quad YL2[j][o] \leftarrow 0$ {Intermediate variables for the calculation of σ_x, σ_y}
  end for
end for

for $i = 1 \dots N$ **do**
  $\boldsymbol{x} \leftarrow \boldsymbol{p}$
  $\vec{\boldsymbol{v}} \leftarrow \vec{\boldsymbol{s}}$
  $o \leftarrow 1$
  while $0 \leq z_{\boldsymbol{x}} \leq z_1$ **do**
    $l \leftarrow \mathcal{L}$
    $\boldsymbol{x} \leftarrow \boldsymbol{x} + l\vec{\boldsymbol{v}}$
    for $j = 1 \dots M$ **do**
      $d \leftarrow \dfrac{z_1 - z_{\boldsymbol{x}}}{\vec{\boldsymbol{s}}'_j \cdot \vec{\boldsymbol{n}}}$
      $L[j][i][o] \leftarrow \beta(d) P(\vec{\boldsymbol{v}} \cdot \vec{\boldsymbol{s}}'_j)$
      $L[j][o] \leftarrow L[j][o] + L[j][i][o]$
      $\boldsymbol{c}[j][i][o] = \boldsymbol{x} + d\vec{\boldsymbol{s}}'_j$
      $x[j][o] \leftarrow x[j][o] + x[j][i][o]\ L[j][i][o]$
      $y[j][o] \leftarrow y[j][o] + y[j][i][o]\ L[j][i][o]$
      $XL2[j][o] \leftarrow XL2[j][o] + x^2[j][i][o]\ L[j][i][o]$
      $YL2[j][o] \leftarrow YL2[j][o] + y^2[j][i][o]\ L[j][i][o]$
    end for
    $\vec{\boldsymbol{v}} \leftarrow \vec{\boldsymbol{V}}$
    $o \leftarrow o + 1$
  end while
end for

for $j = 1 \dots M$ **do**
  for $o = 1 \dots O$ **do**
    $\sigma_x[j][o] \leftarrow \sqrt{\frac{1}{L[j][o]}(XL2[j][o] - x^2[j][o])}$
    $\sigma_y[j][o] \leftarrow \sqrt{\frac{1}{L[j][o]}(YL2[j][o] - y^2[j][o])}$
    $x[j][o] \leftarrow \frac{1}{L[j][o]} x[j][o]$
    $y[j][o] \leftarrow \frac{1}{L[j][o]} y[j][o]$
    $L[j][o] \leftarrow \frac{1}{N} L[j][o]$
  end for
end for

4.3.3 Output of the path tracer

The output of our path tracer is the mean radiance $L[j][o]$ (Equation 4.26), the mean entry point $\boldsymbol{c}[j][o]$ (Equations 4.27 – 4.28) and the standard deviation of entry points ($\sigma_x[j][o]$, $\sigma_y[j][o]$) (Equations 4.29 – 4.30) for each incident direction $\vec{s}_j{}'$ and each order o. From these, we can directly estimate the functions we need in our study. Indeed, we have

$$C(o, z_1, z, \theta, \psi_j, \theta'_j) \approx \frac{L[j][o]}{\cos\theta'}, \tag{4.31}$$

$$\boldsymbol{c}(o, z_1, z, \theta, \psi_j, \theta'_j) \approx \boldsymbol{c}[j][o], \tag{4.32}$$

$$\sigma_x(o, z_1, z, \theta, \psi_j, \theta'_j) \approx \sigma_x[j][o], \tag{4.33}$$

$$\sigma_y(o, z_1, z, \theta, \psi_j, \theta'_j) \approx \sigma_y[j][o]. \tag{4.34}$$

with $z = z_{\boldsymbol{p}}$, $\cos\theta = (\vec{s}\cdot\vec{n})$, $\cos\theta'_j = (\vec{s}_j{}'\cdot\vec{n})$ and $\psi_j = |\arccos(\vec{s}\cdot\vec{x}) - \arccos(\vec{s}_j{}'\cdot\vec{x})|$.

This gives us the solution for the order-dependent BSDF (Equations 4.6) and the first two moment of the order-dependent surface distribution (Equations 4.8 – 4.10). All the other functions that we need in our study (§4.2.5) except D can be deduced from these through Equations 4.11 – 4.24.

To compute the surface distribution D, one solution would be to store the values $L[j][o][i]$ in a photon map [Jen01] at location $\boldsymbol{c}[j][o][i]$. The issue is that we do not know the distribution beforehand, thus we do not know where to place this photon map on the surface nor what its resolution should be.

Although there exist adaptive methods overcoming this problem (which is inherent to photon maps), we prefer to use a more naive method. We simply store all entry points $\boldsymbol{c}[j][o][i]$ and their associated radiance $L[j][o][i]$ on disk, not using any photon map. After the simulation has ended, we can analyze and plot the distribution D through the stored values. The obvious drawback of this method is that it uses a lot of storage space as opposed to photon maps. However, as we will see in §4.4.3, the general shape of the spatial distribution does not vary much with respect to the input parameters. As a result, it will be sufficient for us to store these entry points only for a few cases, yielding decent disk usage.

Note that this storing is only necessary to compute the spatial distribution D. For all other values (*i.e.,* reflectance, BSDF, moments of D, etc.), we do not need to store each traced path, we only store the resulting output values described by Equations 4.31 – 4.34.

4.3.4 Running the experiments

One run our path tracer gives us the result for various incident directions $\vec{s}_j{}'$ and all orders o. It takes as input the viewpoint height z, view angle θ and slab thickness z_1 as well as the various optical parameters of the slab. We store the output $L[j][o]$, $\boldsymbol{c}[j][o]$, $\sigma_x[j][o]$, $\sigma_y[j][o]$ of our simulations in an SQL database for easy read and write access.

We choose $M = 231$ incident directions $\vec{s}_j{}'$ sampling the hemisphere geodesically. This way, one simulation gives us the results for a decent number of incident directions sampling homogeneously the range of possible incident directions. We ran 10,000 simulations with various values of z_1, z and θ, resulting in 20GB of data.

We only calculated the distribution D (*i.e.,* storing all computed entry points) in a few cases. The storing of these entry points resulted in 10GB of data.

These experiments were run on the clusters of INRIA (200-node Itanium cluster) and LLNL (various powerful computer clusters[3]) and took a few weeks of computation[4].

4.3.5 Accuracy and validation

For each simulation, we choose N so as to have a confidence interval of 5% and a confidence level of 95% (see §3.2). That is, we chose N so that

- We have 95% probability that $L[j][o] = L \pm 5\%$.
- We have 95% probability that $\boldsymbol{c}[j][o] = \boldsymbol{c} \pm (5\%\sigma)$.
- We have 95% probability that $\sigma[j][o] = \sigma \pm 5\%$.

Note that, as we have seen in §3.2, this criterion implies that the smaller L is, the more paths we need to compute, hence the more computationally expensive the simulations will be.

To validate our results, we ran simulations with an isotropic phase function and compared the results with theoretical values computed with the X and Y functions [DGS82]. We found the results for luminance to be within the confidence interval.

Note that since we found no available data for these values with the Mie phase function and for the spatial distribution, we could not validate these results.

Summary of §4.3 : We implemented an optimized parallel Monte-Carlo volume path-tracer computing light transport in a slab. It is designed to compute the luminance and the first moments of the spatial distribution given the input parameters. The error of its results are bound thanks to interval estimation. These results are stored in an SQL database. We ran simulations for numerous values of the input parameters.

Nous avons implanté un algorithme de tracé de chemin de Monte-Carlo dans une dalle, optimisé pour une architecture parallèle. Cet algorithme est conçu pour calculer les luminances et les premiers moments de la distribution spatiale, en fonction des paramètres d'entrée. L'erreur des résultats obtenus est bornée par des intervalles de confiance. Ces résultats sont stockés dans une base de données SQL. Nous avons exécuté des simulations pour de nombreuses valeurs différentes des paramètres d'entrée.

4.4 Study analysis

4.4.1 Introduction

Our Monte-Carlo simulations (§4.3) provide us with all the experimental data necessary to study light transport. In this section, we use each of the tools described in §4.2.5 to analyze light transport.

[3] Amusingly, although LLNL hosts a gigantic amount of computing resources (including BlueGene/L, the fastest supercomputer in the world), the demand on these resources is quite high. As a result, using the (rather old) INRIA cluster was more efficient for us than waiting for a free slot in the LLNL queue.

[4] Not counting development time and debugging.

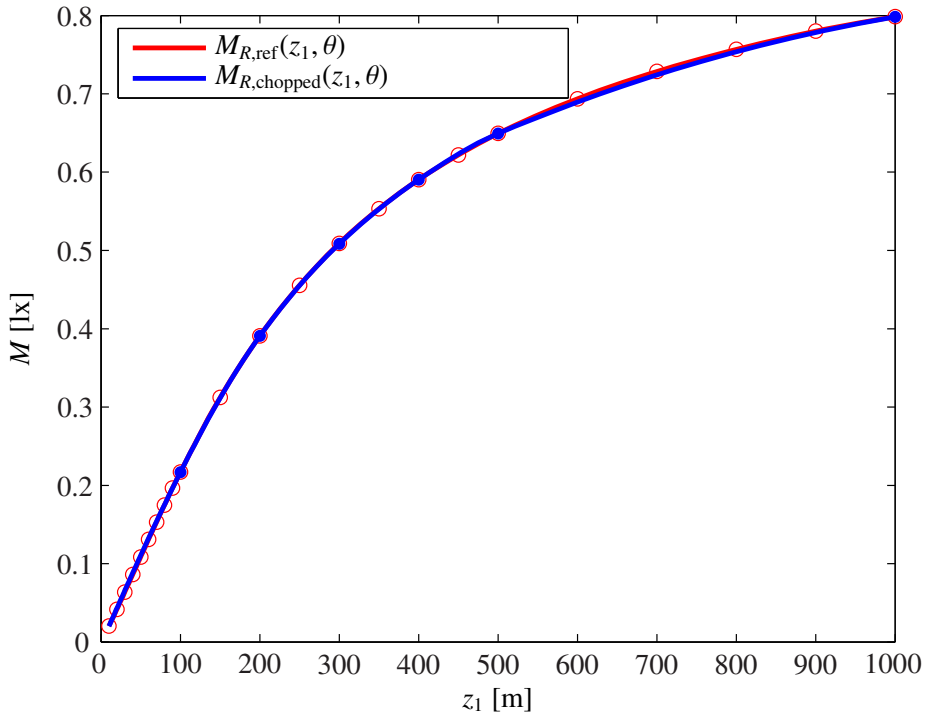


Figure 4.11: *Comparison of the reflected exitance M_R against z_1 between the chopped and the reference model. The dots indicate the sampling points. $\psi = 0°, \theta' = 75°$.*

We begin with analyzing the effect of using the chopped model instead of the reference model (§4.4.2). We then analyze the spatial distribution (§4.4.3) and its moments (§4.4.4). Then we analyze the exitances (§4.4.5), the reflectance and transmittance (§4.4.6) and finally the BSDF (§4.4.7). We summarize the order-dependence of all these functions in §4.4.8. Finally, these analyses allow us to explain the origin of the visual features of real clouds in §4.4.9.

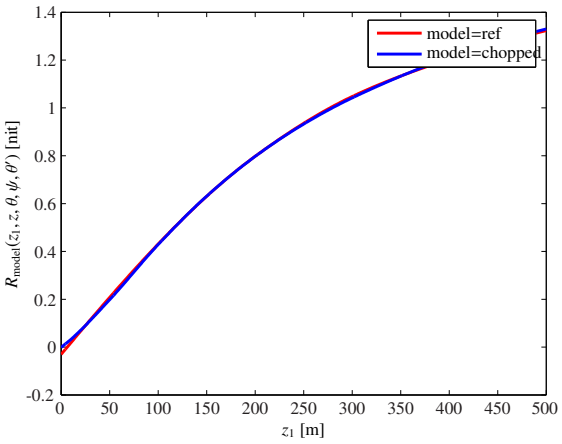
4.4.2 Analysis of the chopped model

4.4.2.1 Validity

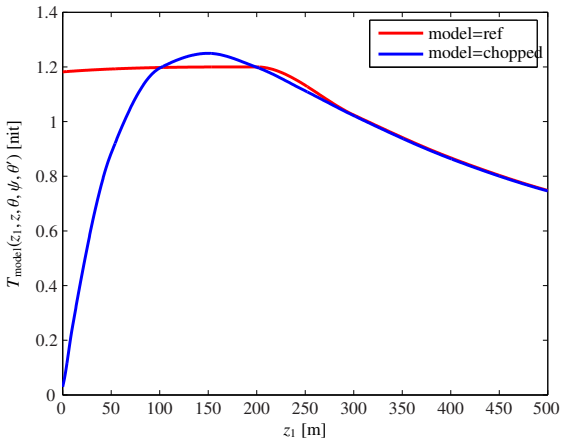
We begin by comparing the accuracy of the chopped model against the results of the reference model. To do so, we use the various tools described in §4.2.5 with both models.

First, we can examine the exitance (§4.2.5.1) with the two models. Figure 4.11 shows these comparisons. We see that both models result in exactly the same reflected exitance. This means the total flux reflected by the slab with the chopped model is the same as with the reference model.

Similarly, we can examine the reflectance and transmittance (§4.2.5.2) against the slab thickness z_1. Figure 4.12 shows this analysis. Like M_R, the reflectance R is the same with both models. Regarding the transmittance T, we can see a strong difference for $z_1 < 200\,\mathrm{m}$ (see Figure 4.12(b)). We can further investigate that difference by plotting T against θ for low values of z_1. Figure 4.13 shows this comparison.

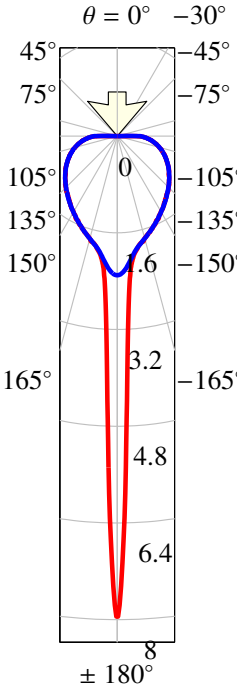


(a) Comparison of the reflectance R against z_1 between the chopped and the reference model. $\theta = 45°, \psi = 0°, \theta' = 0°$.

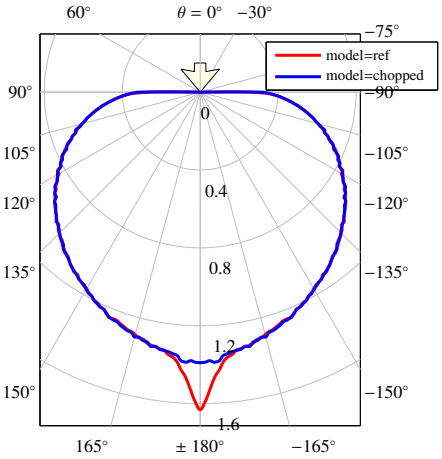


(b) Comparison of the transmittance T against z_1 between the chopped and the reference model. $\theta = 135°, \psi = 0°, \theta' = 0°$.

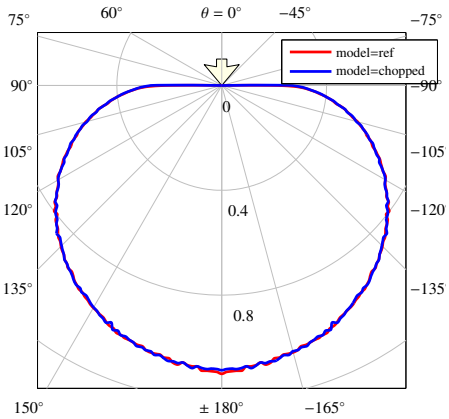
Figure 4.12: *Comparison of results between the reference model and the chopped model for various functions.*



(a) $z_1 = 200$ m.

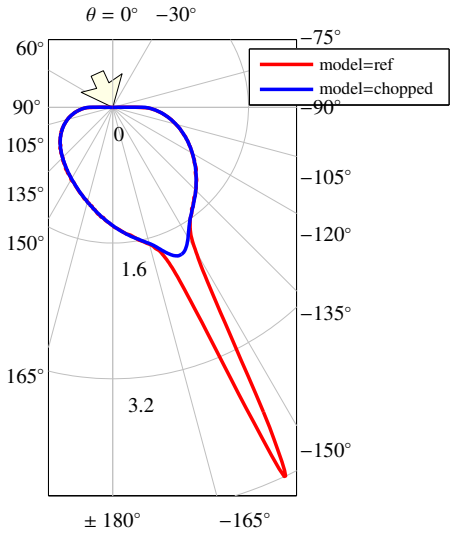


(b) $z_1 = 300$ m.

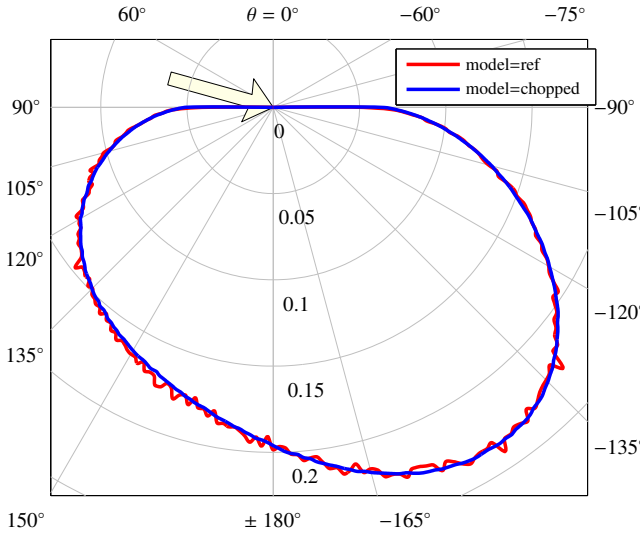


(c) $z_1 = 400$ m.

Figure 4.13: *Comparison of the BTDF T for low values of z_1. We can see that the BTDF with the chopped model lacks a strong forward peak that disappears for $z_1 > 400$ m. $\psi = 0°, \theta' = 0°$.*



(a) $\theta' = 25°$.



(b) $\theta' = 75°$. The high frequencies are due to noise from the Monte-Carlo computations.

Figure 4.14: *Comparison of the BTDF T for varying values of θ'. We can see that the peak that disappears with grazing illumination angles.* $z_1 = 200\,\text{m}, \psi = 0°$.

The error we can see is that the BTDF with the chopped model lacks a strong peak in the forward direction. This error is expected since we removed the strong forward peak of the phase function. This makes the scattered transmittance due to scattering lower than when using the reference phase function. Note that the total transmitted exitance is correct, though, because the missing peak is compensated by a higher transparency since $\kappa_{\text{chopped}} < \kappa_{\text{ref}}$. In a way, the peak is transformed into a dirac delta function in the direction $\vec{s} = -\vec{s}'$. Since we only plot scattered light, this transparency is not shown on the graphs.

We can also see that this peak is only present in very specific cases. These cases arise when the viewpoint depth z' is small enough so that low orders of multiple forward scattering dominate (*i.e.,* the point of view is close to the lit surface and the view direction points towards the light source). We can see it disappears with grazing illumination angles (see Figure 4.14). For this error to be significant, we measured that the input parameters must be in the range

$$\frac{z'}{\mu'} < 400\,\text{m}. \tag{4.35}$$

In all other cases, all results are equivalent with both models. We measured that the width of this peak is at most 8°, *i.e.,* the width of the peak we chopped from the reference phase function. This means that this peak is not significantly spread by multiple scattering.

It is also possible to determine exactly the power of this missing peak. Indeed, the radiance reaching the viewer by transparency with the reference model is

$$\tau_{\text{ref}} = e^{-\kappa_{\text{ref}} z' \cos\theta'}.$$

With the chopped model it is

$$\tau_{\text{chopped}} = e^{-\kappa_{\text{chopped}} z' \cos\theta'}.$$

The difference between this two values is the power of that peak, thus it is the error between the chopped BSDF and the reference BSDF

$$\epsilon = \tau_{\text{chopped}} - \tau_{\text{ref}}. \tag{4.36}$$

Equation 4.36 allows us to estimate the error we make when using the chopped model instead of the reference model. We can also compute the relative error

$$\epsilon_r = \frac{\epsilon}{\int_0^{2\pi}\int_{-\pi}^{\pi} C_{\text{chopped}}(z_1, z, \theta, \psi, \theta') \sin\theta\cos\theta d\theta d\psi}. \tag{4.37}$$

Equation 4.37 allows us to know whether the error is negligible or not. Interestingly, we do not need to compute the whole reference solution to evaluate this relative error. Only the reference transparency τ_{ref} is needed.

Again, note that this error is compensated by an increased transparency. This means the total exitance is correct, but its angular distribution differs from the reference in that what should be a peak is transformed into a dirac delta function.

We consider this error acceptable for various reasons. First, this error is located in a very small portion of the study space. Second, it is delimited well enough (we have an exact estimation of the error, we know it is bounded within $\Theta < 8°$ and $\frac{z'}{\mu'} < 400\,\text{m}$) so that we can hope to find ways to recover from it later. Moreover, there exist many multiple forward scattering approximations that might help us compensate this error. For example, since we know that what is missing is a peak and we know its weight (ϵ), we could add a simple procedural peak to the BSDF. In §6.3.2.1 we show how we recover this peak in our rendering models.

4.4.2.2 Order-dependence

The other interesting observation we can make is that the contribution of each order is doubled when using the chopped model. Indeed, roughly speaking, the chopped model considers two scattering events at once (a strong forward scattering one and a non strong forward scattering one). We can see on Figure 4.15 that

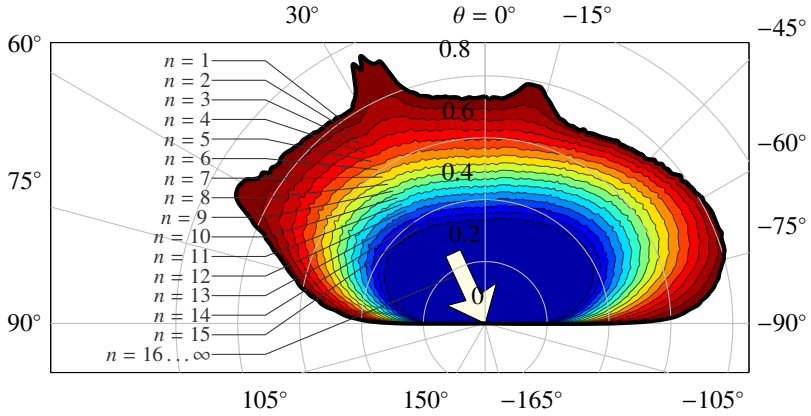
$$R_{\text{chopped}}(1, z_1, z, \theta, \psi, \theta') \approx R_{\text{ref}}([1,2], z_1, z, \theta, \psi, \theta'), \tag{4.38}$$
$$R_{\text{chopped}}(2, z_1, z, \theta, \psi, \theta') \approx R_{\text{ref}}([3,4], z_1, z, \theta, \psi, \theta'), \tag{4.39}$$
$$R_{\text{chopped}}(3, z_1, z, \theta, \psi, \theta') \approx R_{\text{ref}}([5,6], z_1, z, \theta, \psi, \theta'), \tag{4.40}$$

and so on. The same holds for C and T.

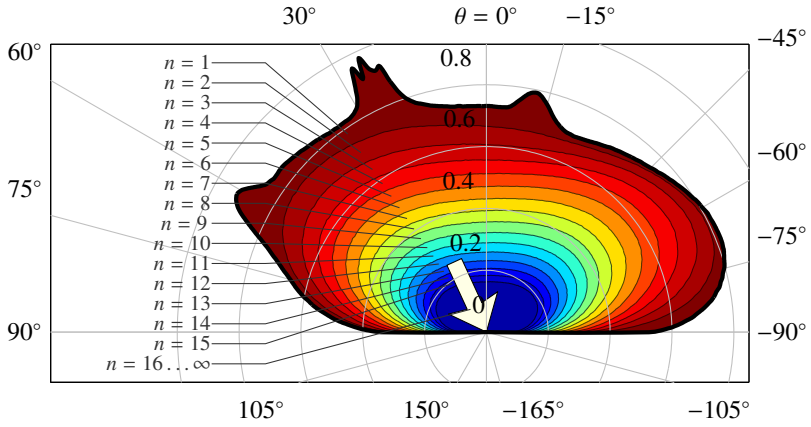
This observation is very interesting in terms of computation costs. Indeed, it means that one can compute the contribution of two orders of scattering of the reference model by computing the contribution of only one order of the chopped model. That means that an approximation can double either its quality or its computation efficiency. For example, a single scattering approximation using the chopped model corresponds to a double scattering approximation using the reference model.

Moreover, since the computation cost of our simulations is inversely proportional to L (§4.3.5) this implies that we need to trace twice as less paths with the chopped model than with the reference model to obtain the same accuracy.

Finally, we can clearly see on Figures 4.15 and 4.16 that when using the chopped model the anisotropy of the angular distribution of C fades much faster with n than when using the reference model. We discuss the behavior of this anisotropy in more details in §4.4.7.



(a) Reference model. The high frequencies are due to noise from the Monte-Carlo computations.



(b) Chopped model.

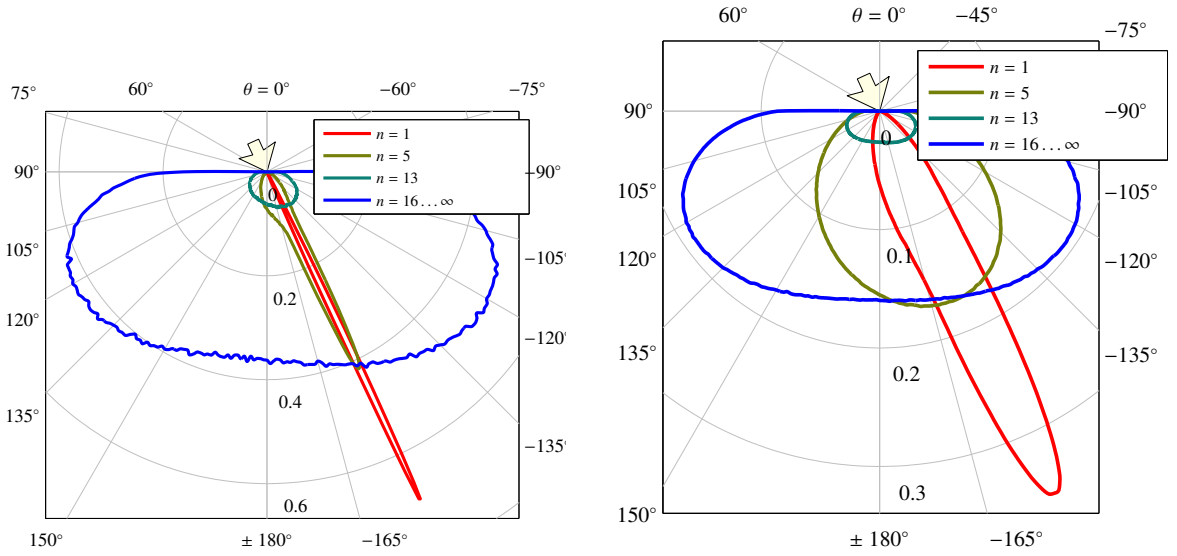
Figure 4.15: *Comparison of the order-dependence of the BRDF R with both models. We can see that with the chopped model each order contributes two times more to the radiance than with the reference model.* $z_1 = 200\,\mathrm{m}, \psi = 0°, \theta' = 25°$.

4.4.2.3 Conclusions

From the above observations, we see that multiple strong forward scattering is only a very specific mode. It happens only in restricted conditions. While this does not mean it should be ignored, it is definitely not the main regime in the light transport of clouds.

From the above observations, we have seen that the error induced by using the chopped model is limited and potentially recoverable. Moreover, using this model gives advantages in terms of speed and accuracy for our Monte-Carlo computations. Finally, using this model may be more practical in the future, since it results in less anisotropic order-dependent angular distributions.

As a consequence, in the rest of this study we only compute light transport with the chopped model as described in §4.2.3.2. For the rest of this thesis we always use the chopped model, unless otherwise specified. As a result, we have now removed one degree of freedom for the rest of the study.



(a) Reference model. The high frequencies are due to noise from the Monte-Carlo computations.

(b) Chopped model.

Figure 4.16: *Comparison of the order-dependence of the BTDF T with both models. We can see that with the chopped model the anisotropy of higher orders fades faster than with the reference model.* $z_1 = 200\,\mathrm{m}, \psi = 0°, \theta' = 25°$.

In Part III, we will show that it is possible to use this chopped model for rendering while rebuilding the missing peak (§6.3.2.1). We call this approach (using the chopped model plus the recovered peak) the "modified-Mie" model in [BNL06].

modified-Mie model

4.4.3 Spatial distribution

The study of the spatial distribution (see §4.2.5.4) consists in analyzing D to see whether light transport is restricted to a small portion of the slab and to see whether its shape can be approximated as one of the standard existing distributions (*e.g.,* a gaussian distribution).

4.4.3.1 Zero and single scattering

Concerning orders 0 and 1, there is no need for Monte-Carlo ray-tracing as this distribution can be computed analytically (§2.5.5.4). For order 0 (transparency), D is a dirac located at the intersection between the view direction and the lit surface

$$D(0, z_1, z, x', y', \theta, \psi, \theta') = \delta_{\theta'}(\theta)\delta_{\phi'}(\phi)\delta\left(x' - z' \sin\theta' \cos\phi'\right)\delta\left(y' - z' \sin\theta' \sin\phi'\right). \tag{4.41}$$

For order 1 (single scattering), S is distributed with a 1D exponential distribution (see Figure 4.17) as

$$S(1, z_1, z, x', y', +\mu, \psi, \mu') = \begin{cases} P(\Theta)\kappa_e e^{-\kappa_e\left(l + l\frac{1-\mu}{\mu'}\right)} & , \quad x' = l\sin\theta\cos\phi, y' = l\sin\theta\cos\phi, 0 \le l \le \frac{z_1 - z}{\mu}, \\ 0 & \text{otherwise,} \end{cases} \tag{4.42}$$

$$S(1, z_1, z, x', y', -\mu, \psi, \mu') = \begin{cases} P(\Theta)\kappa_e e^{-\kappa_e\left(l + \frac{z_1 - z + l\mu}{\mu'}\right)} & , \quad x' = l\sin\theta\cos\phi, y' = l\sin\theta\cos\phi, 0 \le l \le \frac{z}{\mu}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.43}$$

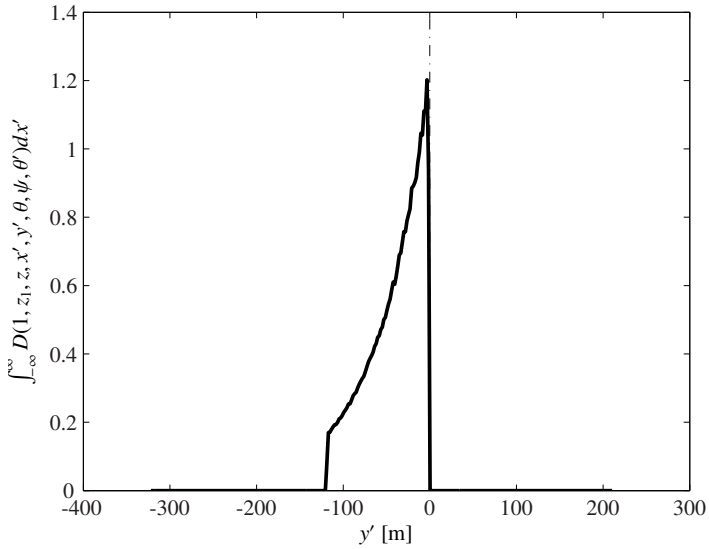


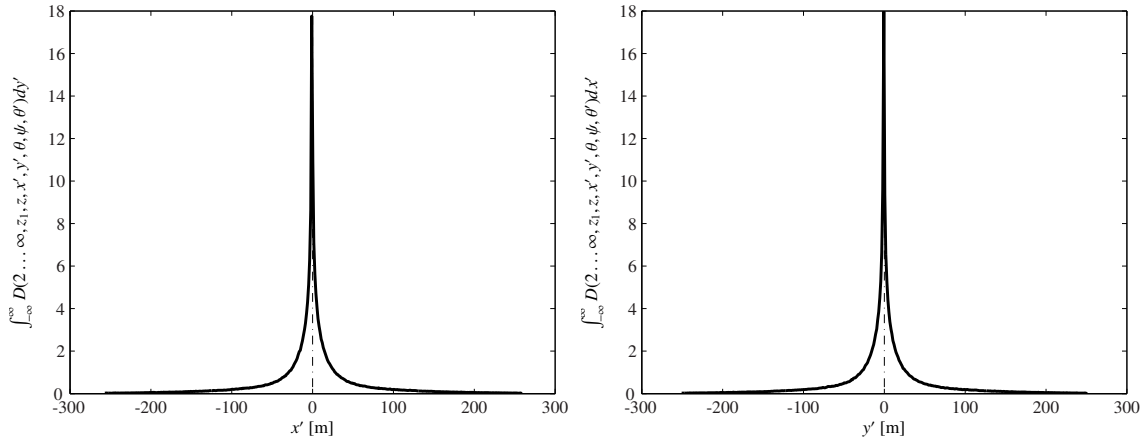
Figure 4.17: *A typical distribution for single scattering (see Equation 4.43).* $z_1 = 50\,\text{m}, z = 0\,\text{m}, \theta = 180°, \psi = 90°, \theta' = 67°$.

4.4.3.2 Multiple scattering

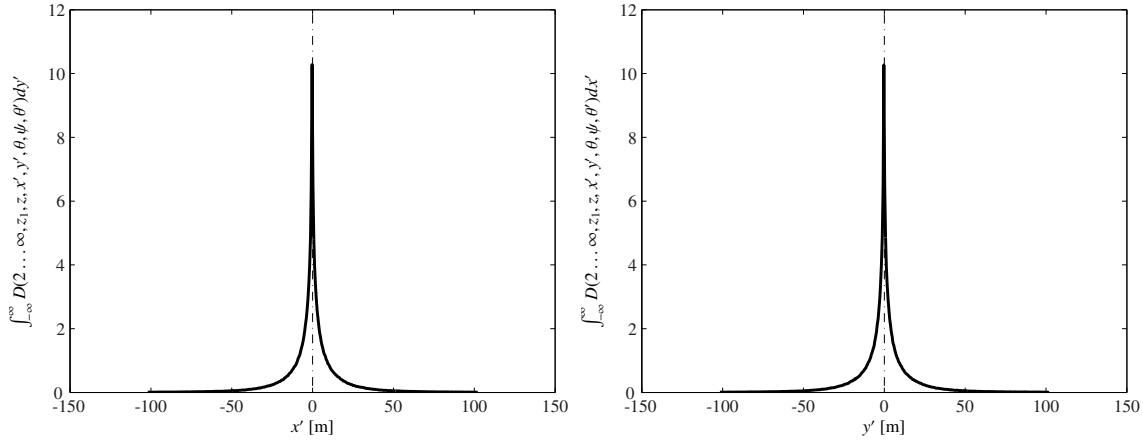
For multiple scattering, we used our Monte-Carlo ray tracer to compute $D([2,\infty], z_1, z, x', y', \theta, \psi, \theta')$ for sparse values of $z_1, z, \theta, \psi, \theta'$ and a good sampling of x' and y'. Figures 4.18 page 140 and 4.19 page 141 show some results of this study.

Due to the very high computational and memory cost of computing D (§4.3), we were limited in the number of cases we could compute and in the error allowed. However, the diagrams on Figures 4.18 and 4.19 allow us to draw several conclusions on the distribution of entry paths, that we summarize here.

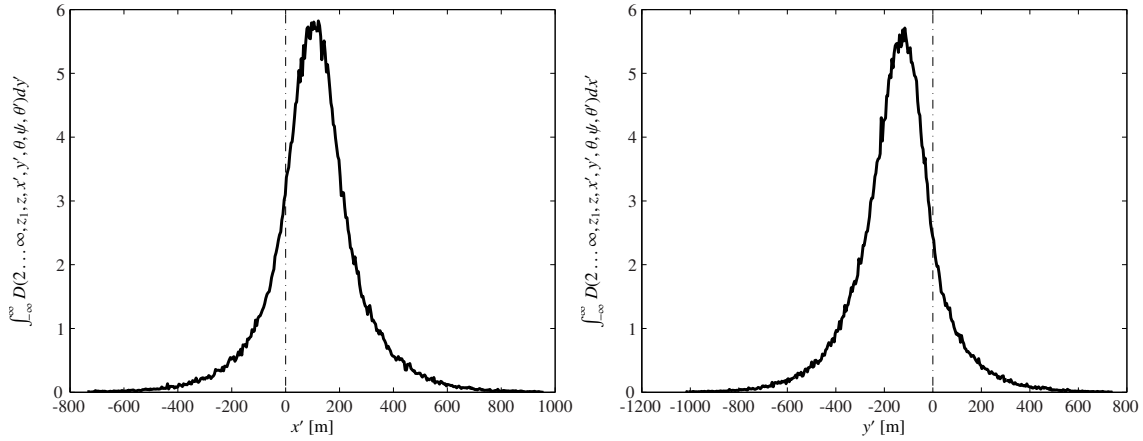
- For slabs of low thickness, D is extremely peaked, resembling a Laplace distribution, *i.e.,* a 2D exponential distribution (Figures 4.18(a) and 4.18(b)).
- As the thickness z_1 increases, the peak of the distribution is smoothed out, and D seems to resemble more and more a Gaussian distribution (Figures 4.18(c) and 4.19). The width of this distribution stays reasonable.
- The shape of the distribution is also strongly dependent on the viewpoint depth. For viewpoints near the lit boundary, the distribution is more peaked than for a viewpoint deep inside the slab (Figure 4.19).
- The center and asymmetry of the distribution depends on the view and light directions (Figure 4.19).
- The distribution seems to be very little "skewed" (*i.e.,* the ratio between σ_x and σ_y is about 1).
- The width of the distribution seems to always be of the order of the slab thickness or much lower for low orders and low viewpoint depths.



(a) $z_1 = 100\,\text{m}, z = z_1, \theta = 0°, \psi = 0°, \theta' = 0°$.

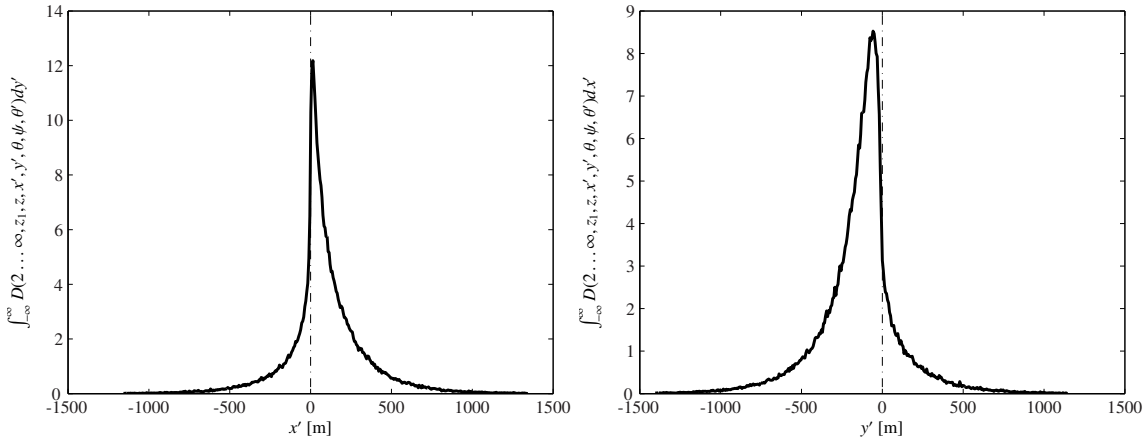


(b) $z_1 = 100\,\text{m}, z = 0\,\text{m}, \theta = 0°, \psi = 0°, \theta' = 0°$.

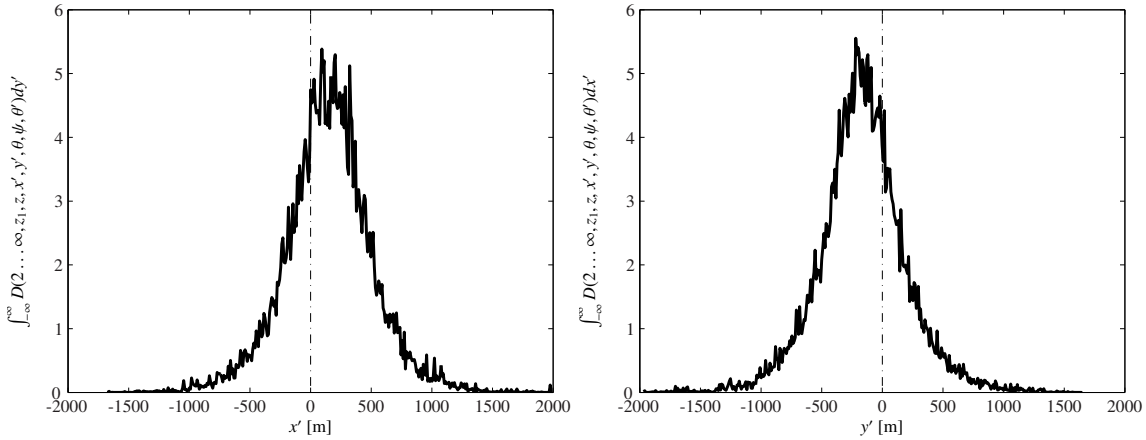


(c) $z_1 = 200\,\text{m}, z = 0\,\text{m}, \theta = 135°, \psi = 90°, \theta' = 67°$. The high frequencies are due to noise from the Monte-Carlo computations.

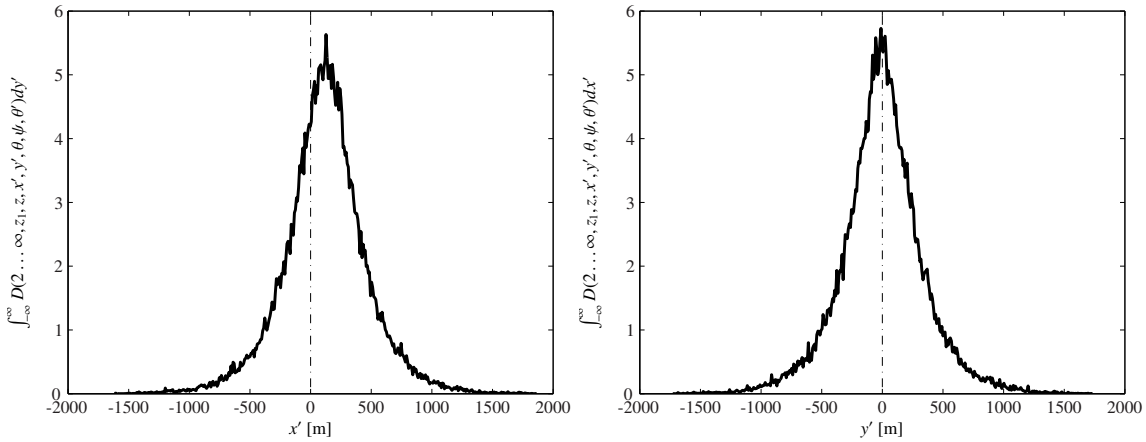
Figure 4.18: *Distribution D of the entry points of light paths on the slab surface.*



(a) $z_1 = 500\,\text{m}, z = z_1, \theta = 135°, \psi = 90°, \theta' = 67°$.

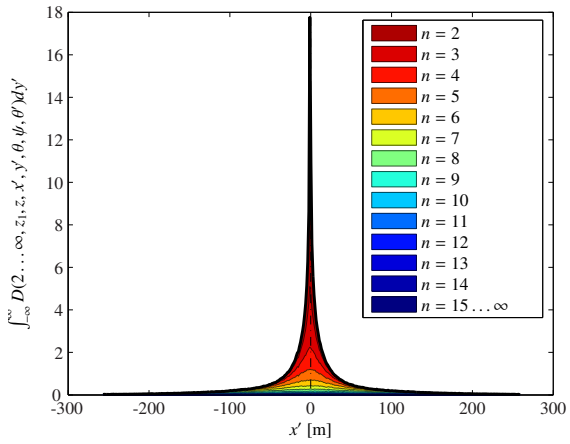


(b) $z_1 = 500\,\text{m}, z = 200\,\text{m}, \theta = 90°, \psi = 90°, \theta' = 67°$. The high frequencies are due to noise from the Monte-Carlo computations.

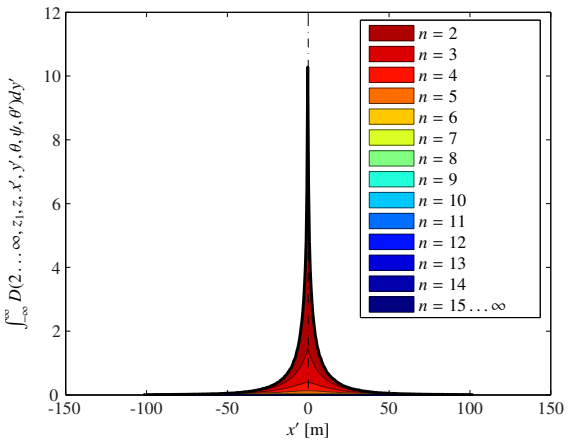


(c) $z_1 = 500\,\text{m}, z = 0\,\text{m}, \theta = 135°, \psi = 0°, \theta' = 45°$. The high frequencies are due to noise from the Monte-Carlo computations.

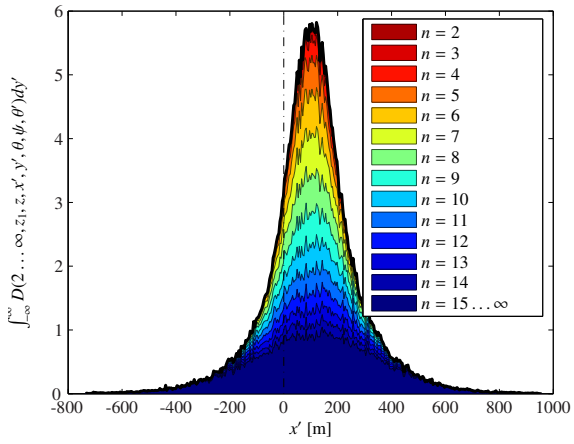
Figure 4.19: *Distribution D of the entry points of light paths on the slab surface.*



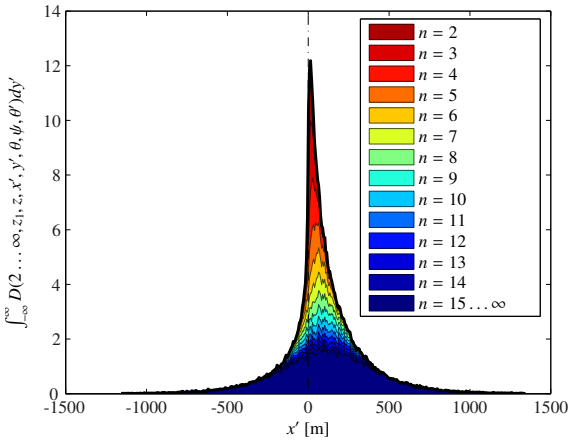
(a) $z_1 = 100\,\text{m}, z = z_1, \theta = 0°, \psi = 0°, \theta' = 0°$.



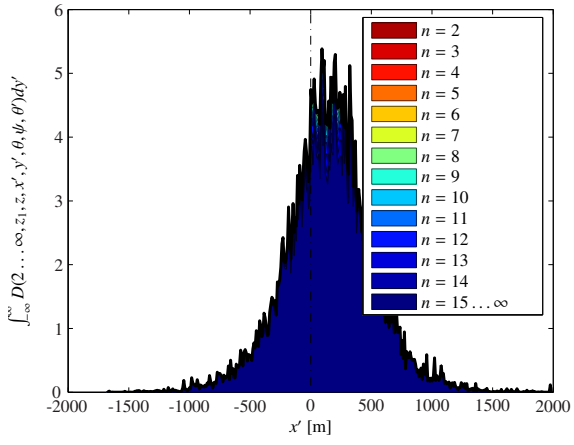
(b) $z_1 = 100\,\text{m}, z = 0\,\text{m}, \theta = 0°, \psi = 0°, \theta' = 0°$.



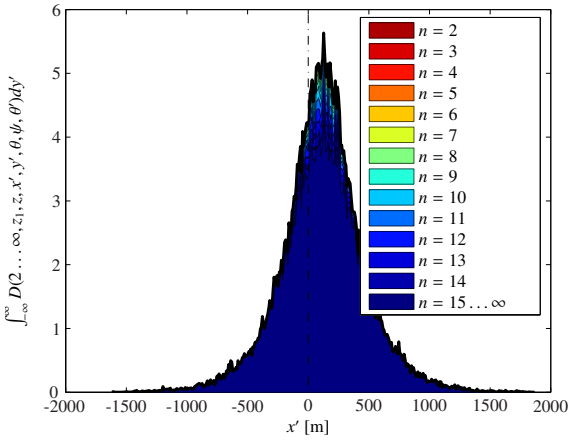
(c) $z_1 = 200\,\text{m}, z = 0\,\text{m}, \theta = 135°, \psi = 90°, \theta' = 67°$. The high frequencies are due to noise from the Monte-Carlo computations.



(d) $z_1 = 500\,\text{m}, z = z_1, \theta = 135°, \psi = 90°, \theta' = 67°$. The high frequencies are due to noise from the Monte-Carlo computations.

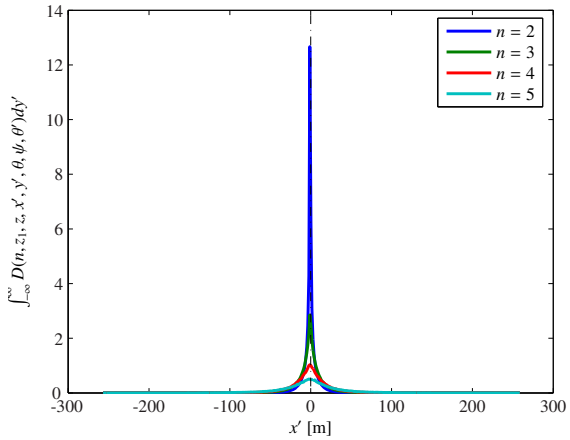


(e) $z_1 = 500\,\text{m}, z = 200\,\text{m}, \theta = 90°, \psi = 90°, \theta' = 67°$. The high frequencies are due to noise from the Monte-Carlo computations.

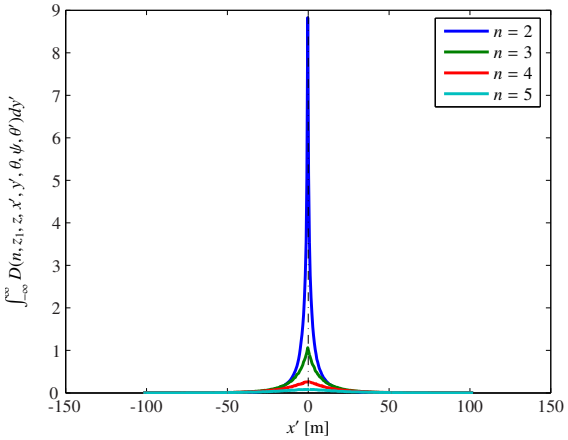


(f) $z_1 = 500\,\text{m}, z = 0\,\text{m}, \theta = 135°, \psi = 0°, \theta' = 45°$. The high frequencies are due to noise from the Monte-Carlo computations.

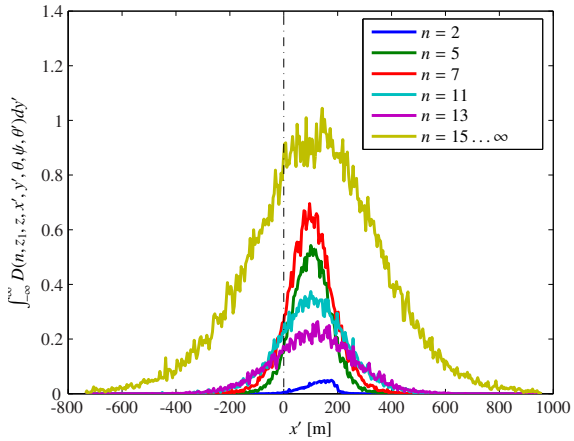
Figure 4.20: *Area plots of the distribution D of the entry points of light paths on the slab surface.*



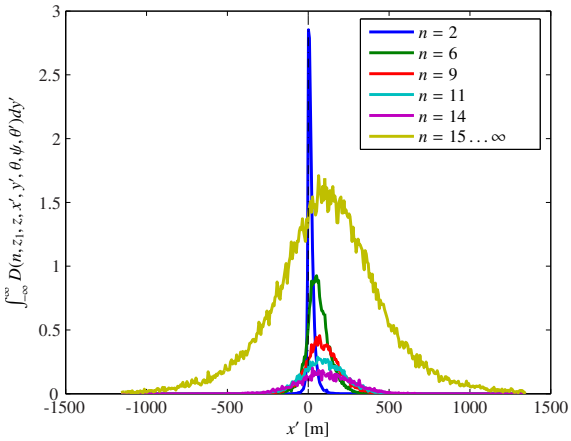
(a) $z_1 = 100\,\text{m}, z = z_1, \theta = 0°, \psi = 0°, \theta' = 0°$.



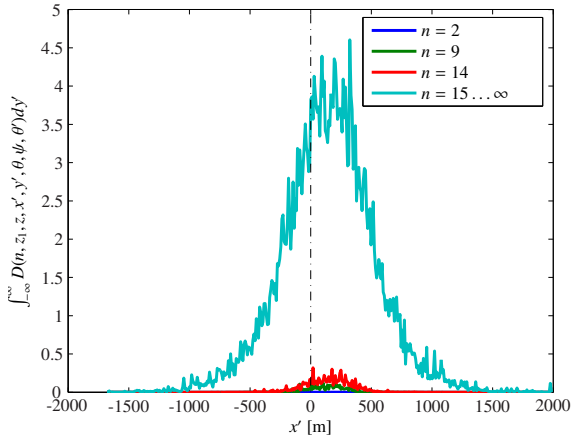
(b) $z_1 = 100\,\text{m}, z = 0\,\text{m}, \theta = 0°, \psi = 0°, \theta' = 0°$.



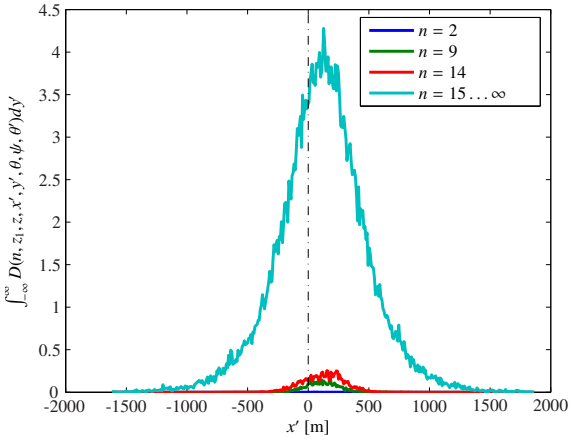
(c) $z_1 = 200\,\text{m}, z = 0\,\text{m}, \theta = 135°, \psi = 90°, \theta' = 67°$. The high frequencies are due to noise from the Monte-Carlo computations.



(d) $z_1 = 500\,\text{m}, z = z_1, \theta = 135°, \psi = 90°, \theta' = 67°$. The high frequencies are due to noise from the Monte-Carlo computations.



(e) $z_1 = 500\,\text{m}, z = 200\,\text{m}, \theta = 90°, \psi = 90°, \theta' = 67°$. The high frequencies are due to noise from the Monte-Carlo computations.



(f) $z_1 = 500\,\text{m}, z = 0\,\text{m}, \theta = 135°, \psi = 0°, \theta' = 45°$. The high frequencies are due to noise from the Monte-Carlo computations.

Figure 4.21: *Order-dependent plots of the distribution D of the entry points of light paths on the slab surface.*

4.4.3.3 Order dependence

By plotting order-dependent plots of D as seen on Figures 4.20 – 4.21 (pp. 142 – 143), we can analyze the shape of D depending on the order n. The observations we have made are

- For low orders, D resembles an narrow exponential distribution. As n increases, the peak of the distribution is smoothed out, and D seems to increasingly resemble a gaussian distribution more than an exponential distribution (Figure 4.21(d)).
- For a given order, the distribution is smoothed as the viewpoint depth increases (*e.g.,* compare Figures 4.21(a) and 4.21(c) for $n = 2$).
- The asymmetry of the distribution decreases with n (Figure 4.21(d))
- The center of the distribution does not vary much with n (Figures 4.21(c), 4.21(d))
- For thin slabs, orders that contribute to radiance are principally low, thus the total distribution is mainly narrow and highly peaked (Figures 4.20(a), 4.20(b)).
- For points of view located under a thick slab, orders that contribute to radiance are mainly high, thus the total distribution is wider and smoother (Figures 4.20(c), 4.20(d), 4.20(e)).
- Orders that contribute significantly to radiance for a point of view close to the lit surface of a thick slab are both low and high, thus the total distribution is a combination of both peaked and smooth distributions. These kinds of distributions have a wide base and high peak (Figure 4.20(d)).

4.4.3.4 Conclusions

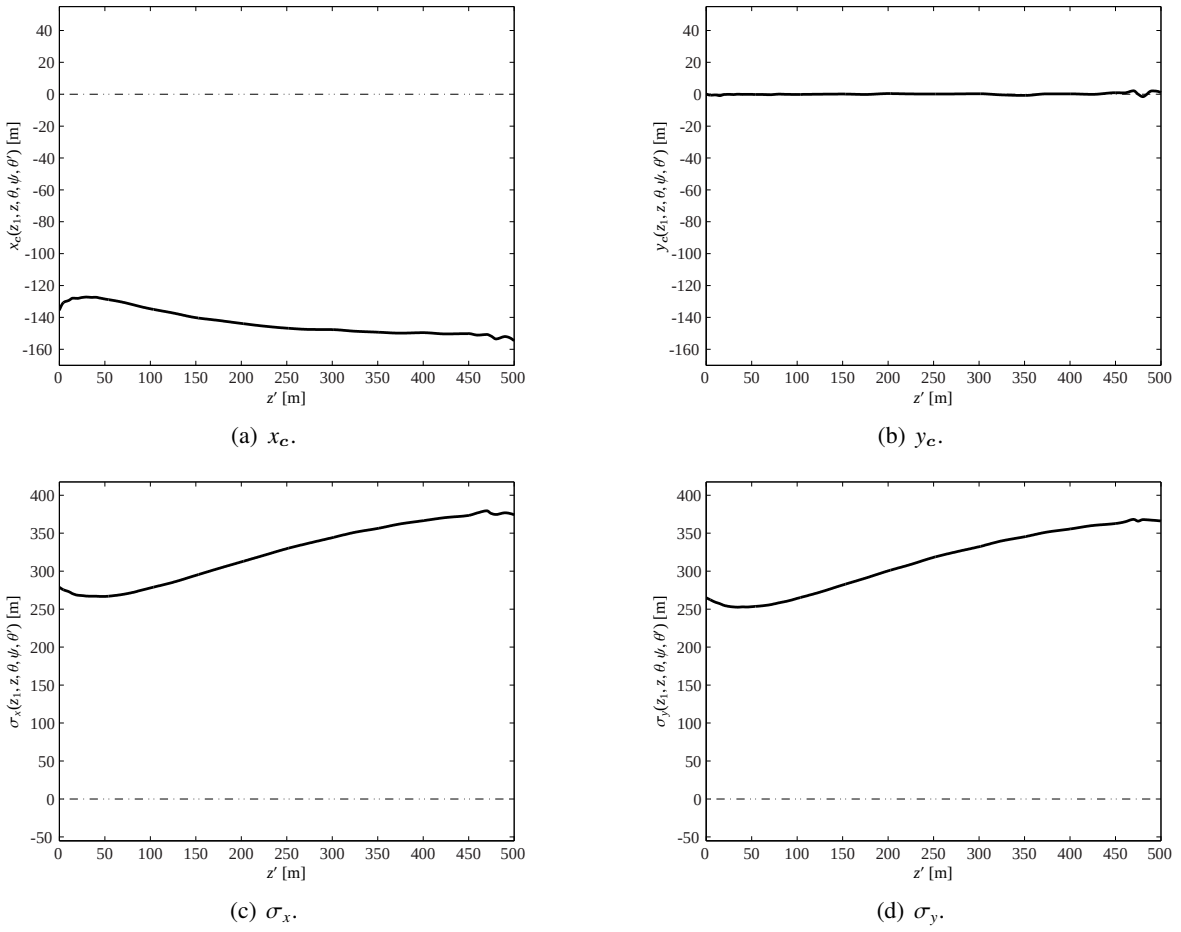
What we retain from these observations is that the *shape* of D (*i.e.,* whether it looks more like an exponential or gaussian or another distribution) varies only slightly with the input parameters. Since it is quite symmetric around its mean $\boldsymbol{c}$ and resembles an exponential or gaussian distribution of standard deviation σ, we consider the entry points of the most significant paths of light as located inside a disk of center $\boldsymbol{c}$ and radius σ. This is an important information that we will use in our rendering model of Chapter 7.

Moreover, since the shape of D varies only slightly with the input parameters we believe it should be sufficient to approximate it with a procedural function such as the gaussian or the exponential distribution functions. As a consequence, in the rest of this study we only compute its first two moments $\boldsymbol{c}$ and $\boldsymbol{\sigma}$ instead of computing the full distribution D. We believe this is sufficient information to reproduce a good approximation of D if needed. Thus in the rest of this study we only analyze the BSDF $C(n, z_1, z, \theta, \psi, \theta')$ and the first two moments $\{\boldsymbol{c}, \sigma_x, \sigma_y\}(n, z_1, z, \theta, \psi, \theta')$ of D. That is, we study 6-DOF functions instead of the original 9-DOF one. This allows us to compute and analyze many more sample points and to study more understandable functions.

Other main observations regarding this distribution are:

- It mainly depends on the order n, slab thickness z_1 and viewpoint depth z',
- Its width is of the order of z_1 or less,
- Its center and asymmetry vary slightly with the view and light directions.

In the next section, we further analyze the behavior of its center and width by studying its first two moments.



(a) x_c. (b) y_c.

(c) σ_x. (d) σ_y.

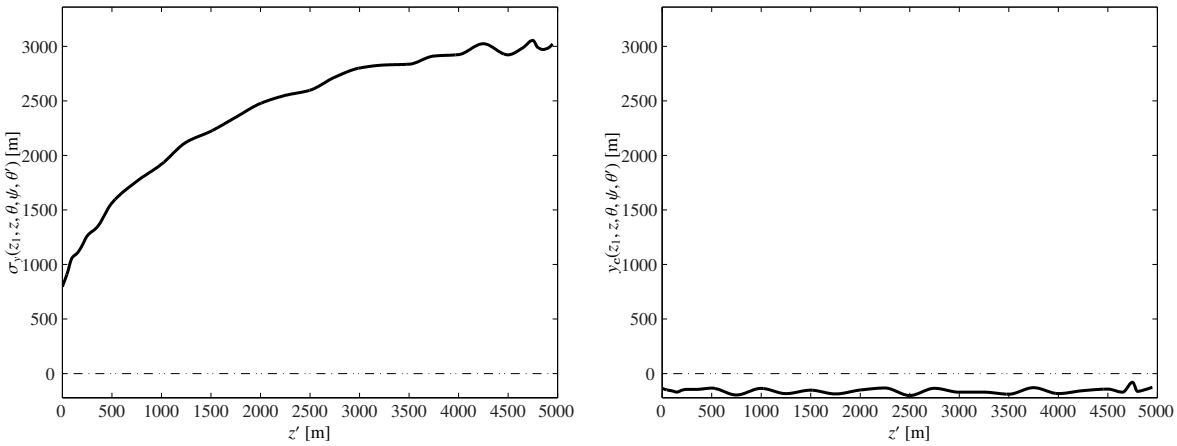
Figure 4.22: *Behavior of the moments of the surface distribution against z' at low scales.* $z_1 = 500, \theta = 90°, \psi = 0°, \theta' = 0°$. *The high frequencies are due to noise from the Monte-Carlo computations.*

4.4.4 Moments of the spatial distribution

The first moment $\boldsymbol{c}$ of D represents the most probable entry point. The second moment (σ_x, σ_y) represents its standard deviation. When analyzing these moments, we observe that their behavior is rather simple. The first moment $\boldsymbol{c}$ depends only very slightly on z_1 and z, as we can see on Figures 4.22 and 4.22. It is primarily dependent on the view and incident angles. It is also the case for the order-dependent formulation. We observe that this most probable entry point is always quite relatively close to the vertical of the viewpoint. As an example, on Figure 4.23, we observe that the standard deviation σ_y ranges from 500 m to 3000 m but the mean entry point stays around $y = -200$ m.

The second moment has a behavior that is almost independent of the view and incident angles. It seems to vary logarithmically with respect to z' (see Figures 4.22 and 4.22). The behavior is similar for the order-dependent second moment. For lower orders, as we already observed in §4.4.3, the second moment decreases with n. The difference between σ_x and σ_y is very low.

The most important information we retain from the analysis of these moments is that the standard deviation of D is always of the order of the viewpoint depth z' or lower. That is, for viewpoints close



(a) [σ_y. (b) y_e.

Figure 4.23: *Behavior of the moments of the surface distribution against z' at high scales.* $z_1 = 5000\,\text{m}$, $\theta = 45°$, $\psi = 90°$, $\theta' = 67°$. *The high frequencies are due to noise from the Monte-Carlo computations.*

to the lit surface, spatial spreading is small. This gives us valuable knowledge about the scale at which light transport takes place.

4.4.5 Transmitted and reflected exitances

In this section we study the reflected and transmitted exitances M_R and M_T (see §4.2.5.1).

4.4.5.1 Behavior with respect to the slab thickness

When plotting the reflected exitance $M_R(z_1, \theta')$ against z_1 (see Figure 4.26(a)), we can see that this exitance increases as z_1, as expected from theory (*e.g.,* the doubling-adding method, §2.5.5.6). The explanation of this phenomenon is rather intuitive. The thicker the slab, the more times a photon will have to be scattered for it to reach the bottom. Since each scattering events is a chance more to turn back upwards, it diminishes the chances for the photon to traverse the slab.

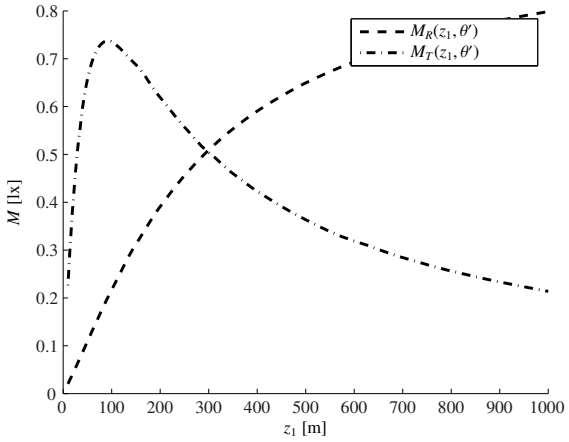
What is less intuitive is that this is the case *despite the strong forward anisotropy of the phase function*. The only difference is that M_R increases less steeply than if $P(\Theta)$ was isotropic. As a matter of fact, as it has been described before, the multiply-scattered phase function converges to isotropy with n. Multiple scattering in a dense isotropic medium can be seen as a diffusion process and a diffusive medium will only be more reflective and less transmissive with respect to its thickness.

Since a cloud is nonabsorbent in the visible spectrum, what is not reflected is transmitted. This means that

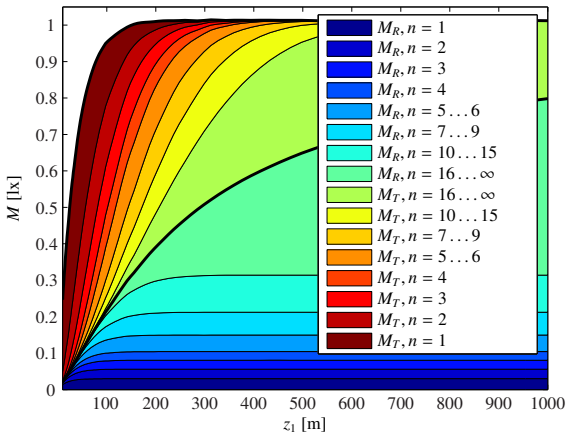
$$M_T([0, \infty], z_1, \theta') = 1 - M_R([0, \infty], z_1, \theta').$$

Thus, M_T is the complement of M_R and has the complement behavior (*i.e.,* decreases with z_1 and with θ'). What we are interested in, though, is the scattered transmitted exitance, that is, the transmitted exitance due to scattering. The transmitted exitance that is not due to scattering is the transparency and is computed analytically by

$$M_T(0, z_1, \theta') = e^{-\kappa_e \frac{z_1}{\cos\theta'}}.$$



(a) Total scattered reflected and transmitted exitances.



(b) Area plot of the exitances M_R and M_T. Areas below the black line represent the reflected exitance M_R, areas above represent the transmitted exitance M_T. The sum of all exitances equals one, except when the slab is thin enough to let some of the light exit via transparency.

Figure 4.24: *Reflected and transmitted exitances.* $\theta' = 0°$.

The scattered transmittance is

$$M_T(z_1, \theta') = M_T([1, \infty], z_1, \theta') = M_T([0, \infty], z_1, \theta') - e^{-\kappa_e \frac{z_1}{\cos\theta'}}.$$

Figure 4.24(a) shows $M_T(z_1, \theta')$ against z_1. What we can see is that the scattered transmittance actually first *increases* with the thickness of a slab, up to a critical thickness $z_{M_T \max}(\theta')$ (about 100 m in our example) where it reaches its maximum $M_{T\max}(\theta')$, then decreases toward 0. We call this behavior the "*lognormal behavior*", since it resembles the lognormal function.

lognormal behavior

This behavior can be explained as follows. If the thickness z_1 is too low, there is not enough matter to intercept the photons. Most of them go through the slab without hitting any droplets. As z_1 increases, there is more matter to intercept and scatter light, and forward scattering makes light continue through the slab and eventually exit it from the bottom. When z_1 increases even more, it reaches a point where more matter means more chances for light to bounce backwards rather that forward.

If we were studying the total exitance including unscattered exitance, M_T would always decrease with z_1. But since we are only considering the scattered exitance, we see this lognormal behavior.

The important observation we can make when comparing M_R and M_T is that the reflected exitance becomes larger than the transmitted exitance quickly. Indeed, Figure 4.24 shows that more light is reflected from $z_1 = 300$ m on, which is a relatively small thickness (as an example, the average thickness of a stratocumulus is about 400-1000 m). For $z_1 = 1000$ m, we see that only 20% of the light traverses the slab. These observations mean that *despite the strong forward anisotropy of the Mie phase function*, a cloud can easily be highly reflective. As a result, a method assuming that multiple scattering is mainly forward will only be valid for thin clouds.

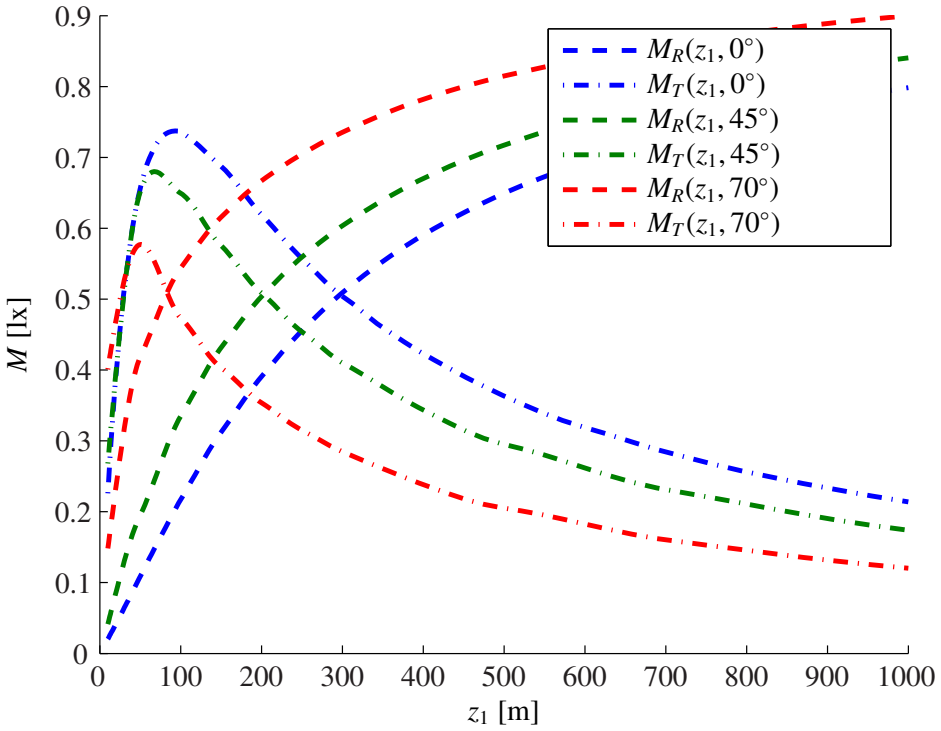


Figure 4.25: *Comparison of exitances with different incident angles. More light is reflected off the slab for grazing angles.*

4.4.5.2 Behavior with respect to the incident angle

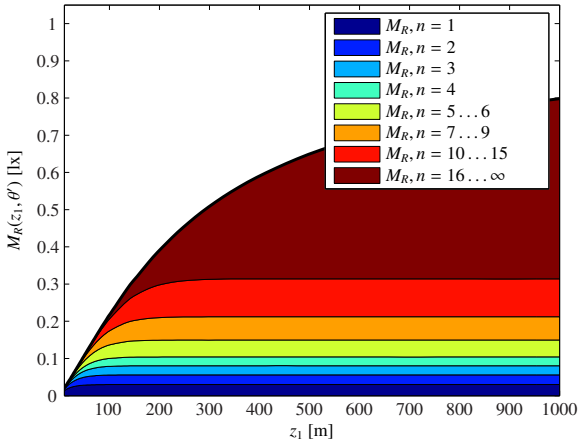
The behavior described above happens for all incident angles. Moreover, as we can see on Figure 4.25, the ratio of reflected over transmitted exitance is even higher with grazing incident angles. Indeed, forward scattering gives more chances for a photon with a grazing incident angle to escape the slab through the top surface, while it makes a photon with an orthogonal incident angle enter deeper in the slab. If the phase function $P(\Theta)$ was isotropic, this behavior would not happen.

4.4.5.3 Order-dependence

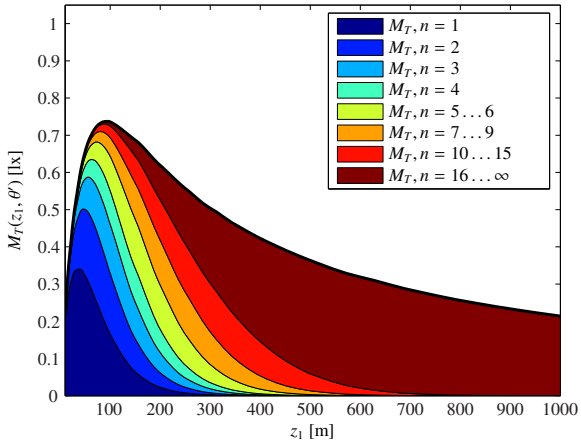
When analyzing the order-dependent plot of the reflected exitance $M_R(n, z_1, \theta')$ against z_1 (Figure 4.26(c)), we observe a few very interesting properties. The contribution $M_R(n, z_1, \theta')$ to the exitance of any given order n has an asymptotic behavior. Indeed, $M_R(n, z_1, \theta')$ rises quickly to an asymptotic value $M_{R\max}(n, \theta')$.

We have already seen (Figure 4.24(a)) that the total scattered exitance (*i.e.,* due to all orders of scattering) has an asymptotic value of 1. However, its converges to this value slowly, and reaches it only for $z_1 = \infty$. On the other hand, the convergence for a given order of scattering is much faster, as we can see on Figure 4.26(c). As an example, $M_R(2, z_1, 0°)$ reaches its asymptotic value for $z_1 = 100\,\text{m}$, and $M_R([10, 15], z_1, 0°)$ reaches its asymptotic value for $z_1 = 400\,\text{m}$.

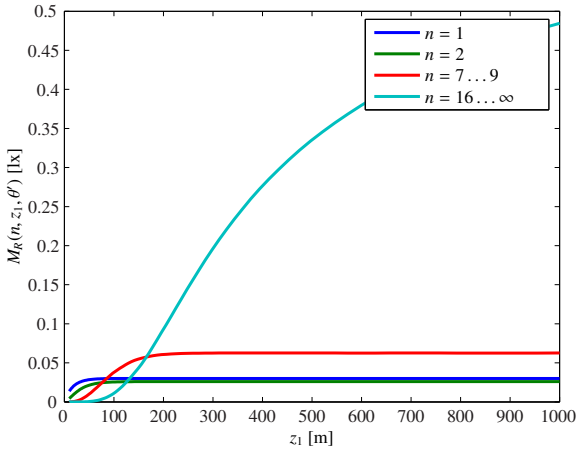
Second, orders "pile up" one after the other as the slab increases. That is, for very thin slabs, only order 1 contributes to M_R. As z_1 increases, $M_R(2, z_1, \theta')$ begins to be significant. As z_1 continues



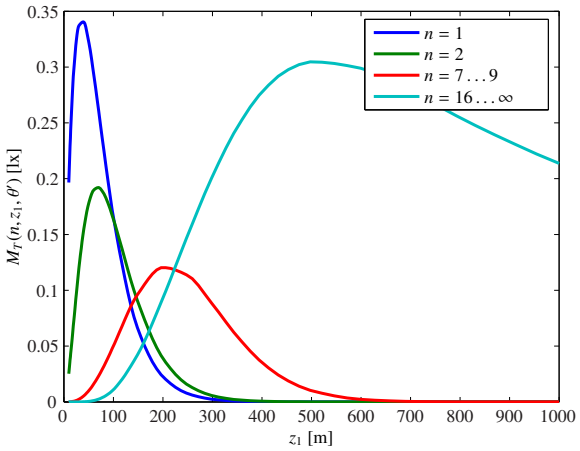
(a) Area plot of the reflected exitance M_R against z_1.



(b) Area plot of the transmitted exitance M_T against z_1.



(c) Order-dependent plot of the reflected exitance M_R against z_1. We can clearly an asymptotic behavior.



(d) Order-dependent plot of the transmitted exitance M_T against z_1. We can clearly an lognormal behavior.

Figure 4.26: *Order-dependent plots of reflected and transmitted exitances.* $\phi = 0°, \theta' = 0°$.

to increase, $M_R(3, z_1, \theta')$ then $M_R(4, z_1, \theta')$ become significant, one after the other, and so on. At the same time, lower orders reach their asymptotic value. This is clearly visible on Figure 4.26(a).

When studying the order-dependent transmitted exitance $M_T(n, z_1, \theta')$ against z_1 (Figure 4.26(d)), we can see it has the same lognormal behavior as for the total scattered exitance. That is for any given order n, M_T increases first to a maximum value at a critical thickness, then decreases.

There is an intuitive explanation of this phenomenon in the case of single scattering ($n = 1$). The mean free path $l_0 = \frac{1}{\kappa_e}$ is the mean distance at which a photon traverses between two scattering events (see §2.5.2.3). If we search for the critical thickness $z_{M_T \max}(1, \theta')$ for which single scattering will be the strongest, we find $z_1 = \frac{l_0}{\cos\theta'}$. Indeed, if z_1 is lower than this value, photons will have more chances to traverse the slab without hitting any droplet, thus fewer chances to hit one single droplet. If z_1 is higher than this value, photons will have more chances to hit a droplet, but also more chances to hit a second one after the first scattering event. Thus, $z_{M_T \max}(1, \theta') = \frac{l_0}{\cos\theta'}$ is the thickness for which $M_T(1, z_1, \theta')$ is maximum. We can see on Figure 4.26(d) that this critical thickness $z_{M_T \max}(1, 0°)$ is about 50 m, which corresponds with the optical value we used: $l_0 = \frac{1}{\kappa_{\text{chopped}}} = 46\,\text{m}$.

The same explanation holds for higher orders of scattering. The difference is that the critical thickness increases with n, and the maximum exitance of a given order decreases with n.

Finally, we can see on Figure 4.26(b) that the orders that contribute significantly to the transmitted exitance increase with the slab thickness. As an example, orders $n = [1, 3]$ are the main contributors to $M_T(50\,\text{m}, 0°)$. For $M_T(200\,\text{m}, 0°)$ we measure $n = [3, 15]$. For $M_T(700\,\text{m}, 0°)$, only orders $n > 15$ contribute.

4.4.5.4 Conclusions

From our observations of exitance, we retain the following important conclusions.

- $M_R(z_1, \theta')$ tends towards 1 with z_1.
- $M_T(z_1, \theta')$ has a lognormal behavior. That is, it increases and then decreases with z_1.
- $M_R(z_1, \theta')$ (respectively $M_T(z_1, \theta')$) increases (respectively decreases) with θ'.
- All these properties hold for a given order (or group of orders) of scattering, *i.e.,*
  - $M_R(n, z_1, \theta')$ tends towards a maximum $M_{R\max}(n, \theta')$ at a critical thickness $z_1 = z_{M_{R\max}}(n, \theta')$.
  - $M_T(n, z_1, \theta')$ has a lognormal behavior. That is, it reaches a maximum value $M_{T\max}(n, \theta')$ at a critical thickness $z_1 = z_{M_{T\max}}(n, \theta')$, then tends towards zero.
  - $M_R(n, z_1, \theta')$ (resp. $M_T(n, z_1, \theta')$) increases (resp. decreases) with θ'.
- $M_{T\max}(n, \theta')$ decreases with n.
- $z_{M_{T\max}}(n, \theta')$ and $z_{M_{R\max}}(n, \theta')$ increase with n.
- Only thin slabs ($z_1 < 300\,\text{m}$ in the case of this study) are more transmissive than reflective.

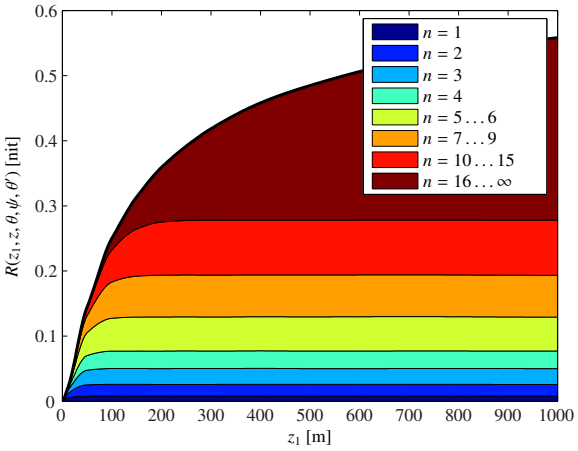
4.4.6 Transmittance and reflectance

In this section we study the reflectance $R(z_1, \theta, \psi, \theta')$ and transmittance $T(z_1, \theta, \psi, \theta')$ (see §4.2.5.2).

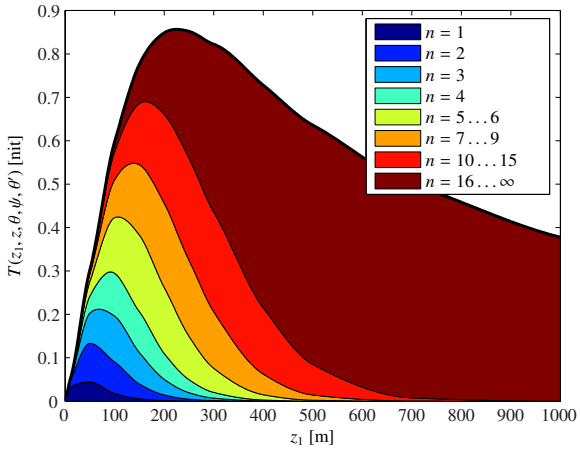
4.4.6.1 Behavior with respect to the slab thickness

The behavior of reflectance and transmittance with respect to z_1 is exactly the same as for the exitances M_R and M_T, as we can see in Figure 4.27. That is,

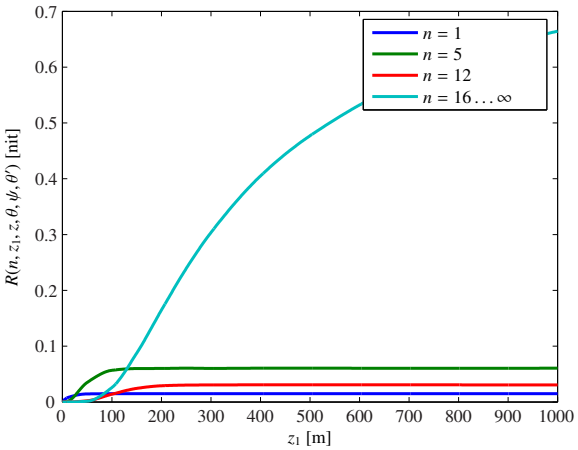
- $R(z_1, \theta, \psi, \theta')$ tends towards a maximum value $R_{\max}(\theta, \psi, \theta')$ with z_1.
- $T(z_1, \theta, \psi, \theta')$ has a lognormal behavior. That is, it increases reaches a maximum value $T_{\max}(\theta, \psi, \theta')$ for a critical thickness $z_1 = z_{T\max}(\theta, \psi, \theta')$, then tends towards zero.
- All these properties hold for a given order (or group of orders) of scattering, *i.e.,*
  - $R(n, z_1, \theta, \psi, \theta')$ tends towards a maximum $R_{\max}(n, \theta, \psi, \theta')$ at a critical thickness $z_1 = z_{R\max}(n, \theta, \psi, \theta')$.
  - $T(n, z_1, \theta, \psi, \theta')$ has a lognormal behavior. That is, it reaches a maximum value $T_{\max}(n, \theta, \psi, \theta')$ at a critical thickness $z_1 = z_{T\max}(n, \theta, \psi, \theta')$, then tends towards zero.
- $T_{\max}(n, \theta, \psi, \theta')$ decreases with n.
- $z_{T\max}(n, \theta, \psi, \theta')$ and $z_{R\max}(n, \theta, \psi, \theta')$ increase with n.



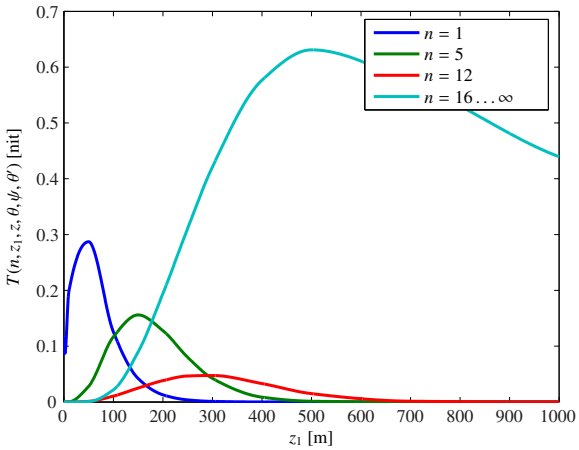
(a) Area plot of the reflectance R against z_1. $\theta = 45°, \psi = 0° \theta' = 25°$.



(b) Area plot of the transmittance T against z_1. $\theta = 135°, \psi = 90° \theta' = 67°$.



(c) Order-dependent plot of the reflectance R against z_1. We can clearly see an asymptotic behavior. $\theta = 45°, \psi = 180° \theta' = 45°$.



(d) Order-dependent plot of the transmittance T against z_1. We can clearly see a lognormal behavior. $\theta = 135°, \psi = 0° \theta' = 0°$.

Figure 4.27: *Order-dependent plots of reflectance and transmittance. Note the similitude with Figure 4.26.*

When considering $R(1, z_1, \theta, \psi, \theta')$ and $T(z_1, \theta, \psi, \theta')$, we can use the exact analytical formulation to put a mathematical explanation on the intuitive explanation we gave for M_R and M_T.

The analytical formulation $R(1, z_1, \theta, \psi, \theta')$ is a special case of Equation 2.33:

$$R(1, z_1, \mu, \psi, \mu') = P(\Theta) \frac{1}{\mu' + \mu} \left[1 - \beta \left(\frac{z_1}{\mu} + \frac{z_1}{\mu'} \right) \right]. \tag{4.44}$$

Similarly, $T(1, z_1, \theta, \psi, \theta')$ is a special case of Equation 2.31:

$$T(1, z_1, \mu, \psi, \mu') = P(\Theta) \frac{1}{\mu' - \mu} \left[\beta \left(\frac{z_1}{\mu'} \right) - \beta \left(\frac{z_1}{\mu} \right) \right]. \tag{4.45}$$

We can see that R is of the form $1 - e^{-x}$. We can compute analytically its derivative and its asymptotic value

$$\frac{d}{dz_1} R(1, z_1, \mu, \psi, \mu') = P(\Theta) \frac{\kappa_e}{\mu\mu'} \beta\left(\frac{\mu + \mu'}{\mu\mu'} z_1\right), \tag{4.46}$$

$$R_{\max}(1, \mu, \psi, \mu') = \lim_{z_1 \to \infty} R(1, z_1, \mu, \psi, \mu') = P(\Theta) \frac{1}{\mu' + \mu}. \tag{4.47}$$

We can also see that T is of the form $e^{-ax} - e^{-bx}$. Thus its derivative is

$$\frac{d}{dz_1} T(1, z_1, \mu, \psi, \mu') = P(\Theta) \frac{\kappa_e}{(\mu' - \mu)\mu\mu'} \left[\beta\left(\frac{z_1}{\mu}\right) - \beta\left(\frac{z_1}{\mu'}\right)\right]. \tag{4.48}$$

We can thus compute the critical thickness $z_{T\,\max}$ and the maximum transmittance $T_{\max}$:

$$z_{T\,\max}(1, \mu, \psi, \mu') = \frac{\ln(\frac{\mu'}{\mu})\mu\mu'}{\kappa_e(\mu - \mu')} \tag{4.49}$$

$$T_{\max}(1, \mu, \psi, \mu') = T(1, z_{T\,\max}, \mu, \psi, \mu') = P(\Theta) \frac{1}{\mu - \mu'} \left[\left(\frac{\mu'}{\mu}\right)^{\left(\frac{\mu'}{\mu - \mu'}\right)} - \left(\frac{\mu'}{\mu}\right)^{\left(\frac{\mu}{\mu - \mu'}\right)}\right]. \tag{4.50}$$

Single scattering is the only case for which there is an analytical solution for these values. It can give us hints on how we could approximate R and T for higher orders. For example, since Equation 4.44 is of the form e^{-x} it seems reasonable to try to approximate M_R and $R(n, z_1, \mu, \psi, \mu')$ with a function of the form e^{-x}. Similarly, since Equation 4.45 is of the form $e^{-ax} - e^{-bx}$ it seems also logical to approximate M_T and $T(n, z_1, \mu, \psi, \mu')$ with a function of the form $e^{-ax} - e^{-bx}$.

4.4.6.2 Other behaviors

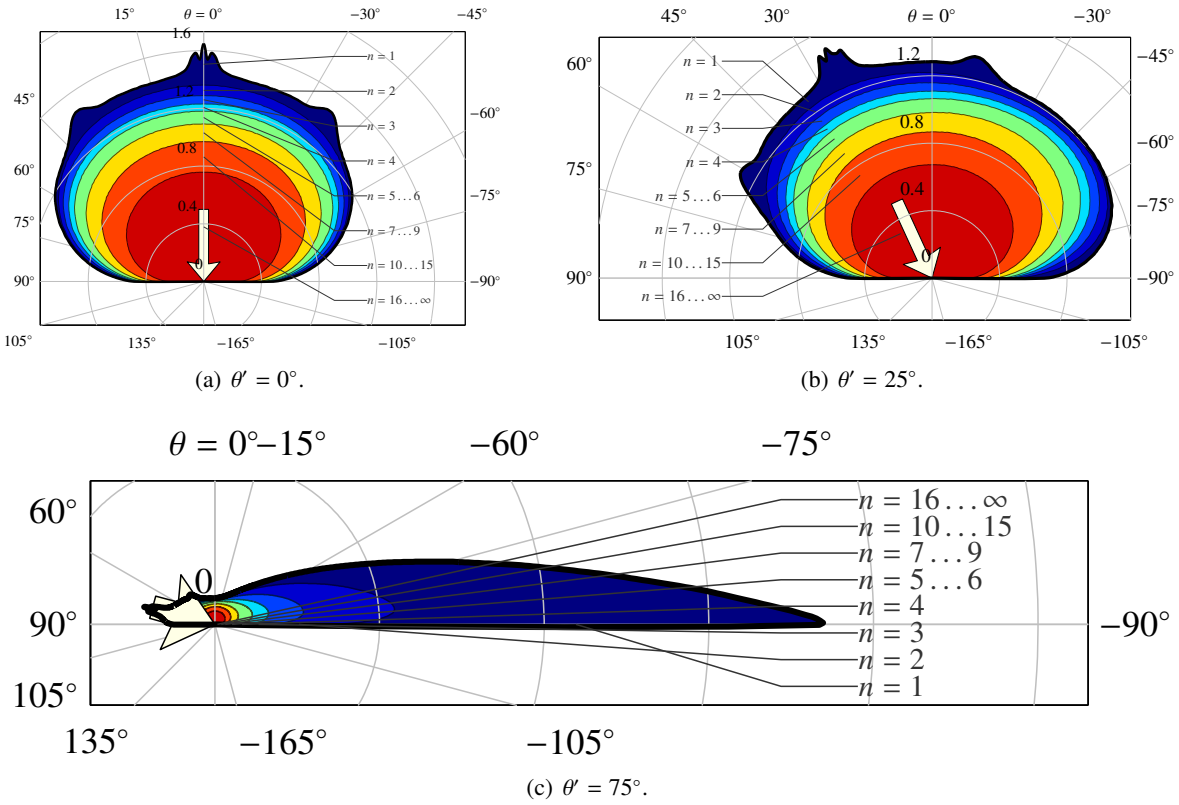
Since R and T are particular cases of the BSDF C, the behavior of their angular distribution is exactly the same as for C. The same holds for the order dependence. We study these behaviors in §4.4.7.

4.4.6.3 Conclusions

We have observed that the behavior of the reflectance R and transmittance T with respect to z_1 is very similar to that of the reflected and transmitted exitances. In the case of single scattering, we actually have an analytical formulation of this behavior. This is a very interesting information because it gives us a hint on what kind of procedural function may mimic the behavior of higher orders.

4.4.7 BSDF

In this section we finally study the 6-DOF BSDF $C(z_1, z, \theta, \psi, \theta')$ (see §4.2.5.3). C can be seen as a generalization of the reflectance R and transmittance T. More precisely, R and T are special cases of C: R is C for $z = z_1$ and T is C for $z = 0$.



(a) $\theta' = 0°$.

(b) $\theta' = 25°$.

(c) $\theta' = 75°$.

Figure 4.28: *Angular distribution of the reflectance R at a given viewpoint depth and slab thicknesses for varying incident angles.* $z_1 = 500\,\text{m}, z = z_1, \psi = 0°$.

4.4.7.1 Angular distribution

We begin by studying the behavior of C against the input angles. What is interesting to analyze here is the anisotropy of this directional distribution. The anisotropy is linked with the view-dependent effects we can observe by eye. If we are able to characterize the anisotropy of C, *i.e.,* to determine what are the conditions for C to be anisotropic, it might help us determine where the view-dependent effects come from. The other analysis we can do is to observe whether this angular distribution is axially symmetric. Indeed, even if it is not isotropic, being axially symmetric means it does not depend on ψ, thus removing one DOF in its expression. This can help us finding phenomenological models for C.

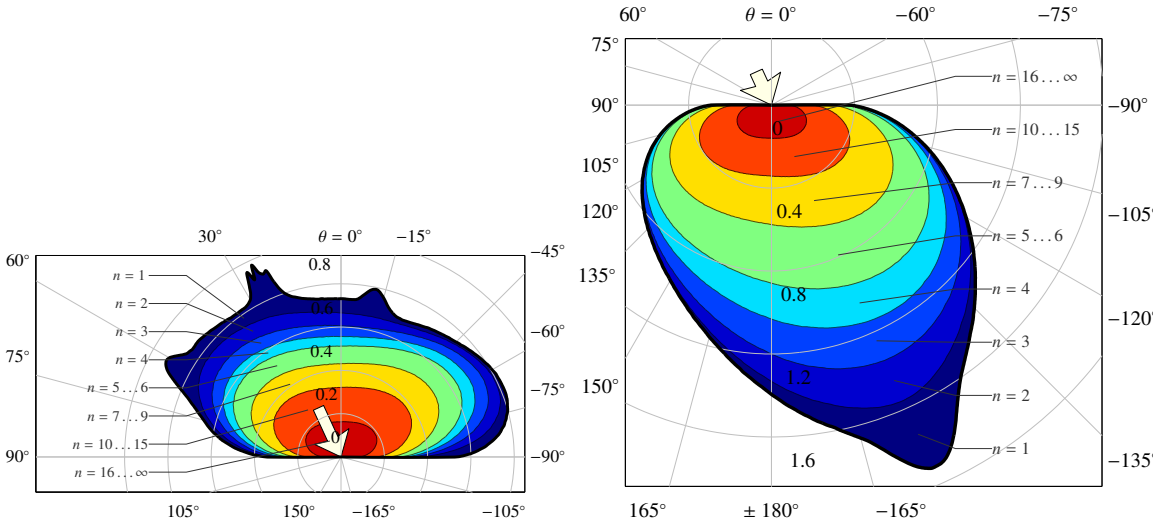
We start by looking at the reflectance R and the transmittance T, which are understandable subsets of the BSDF C. Figure 4.28 shows the reflectance R for various incident angles. As we can expect from our study of exitance and reflectance, more light is reflected as θ' increases. What is interesting to notice is that this light is reflected in a highly anisotropic way. A strong lobe appears in R in the approximate direction of reflection (Figure 4.28(c)). This lobe corresponds to the pseudo-specular effect we describe in §2.2.2.2. It is present for any given thickness. Regarding T, its anisotropy diminishes as θ' increases.

Figure 4.29 shows the BRDF R and BTDF T of a slab for varying thicknesses. Here are the things we can observe.

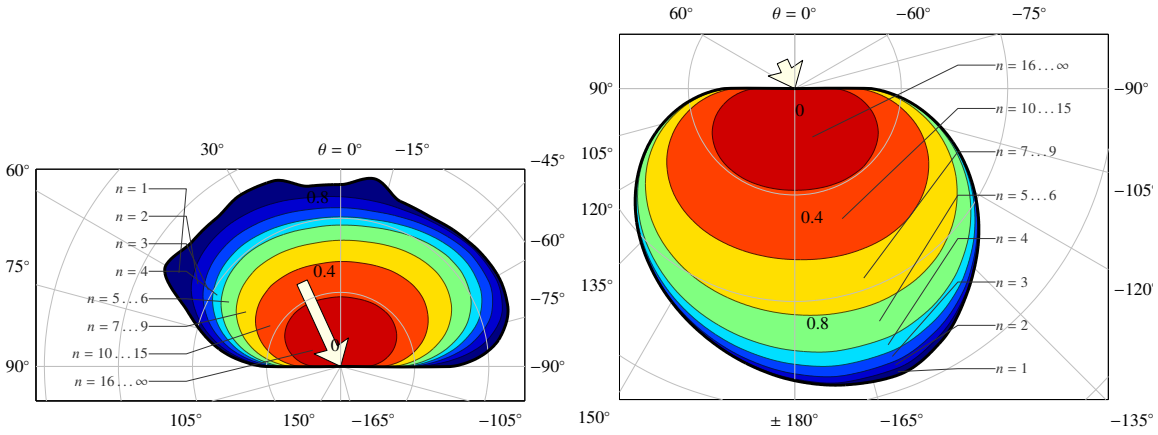
- For thin slabs, we can see an anisotropy in the forward direction for T (Figure 4.29(a), right).
- For all thicknesses, we can see small peaks in R in the direction of $\vec{s}'$ and around it (Figure 4.29, left). These correspond to the glory and fogbow described in §2.2.2.2. They are caused by single scattering (order 1).
- The higher the order, the more axially symmetric R and T are (Figure 4.29(c)).
- Since the contribution of paths of high orders increases with the slab thickness z_1 (see §4.4.6), R and T tend to be more axially symmetric for thick slabs (Figure 4.29(c).
- Since R is a sum of low and high-order paths (see §4.4.6), it grows axially symmetric with thickness but always keeps the anisotropic features brought by low orders (Figure 4.29(c), left).
- Since the significant orders for T are higher as z_1 increases, it becomes totally axially symmetric for thick slabs (Figure 4.29(c), right).
- R and T are never isotropic. They are axially symmetric at best. However, the BTDF T tends towards isotropy for thick slabs (Figure 4.29(c), right).

Figure 4.30 shows the BSDF C for various viewpoint depths. It shows how this BSDF passes from the reflectance R (Figure 4.30(a)) to the transmittance T (Figure 4.30(f)) as the viewpoint depth increases. We observe that a forward peak appears as the viewpoint enters the cloud, while the glory and fogbow peaks disappear (Figure 4.30(b)). As the viewpoint goes deeper, the forward peak disappears (Figures 4.30(c), 4.30(d)). Then the backward part decreases, leaving only the axially symmetric contribution of high orders in the bottom direction (Figures 4.30(e)).

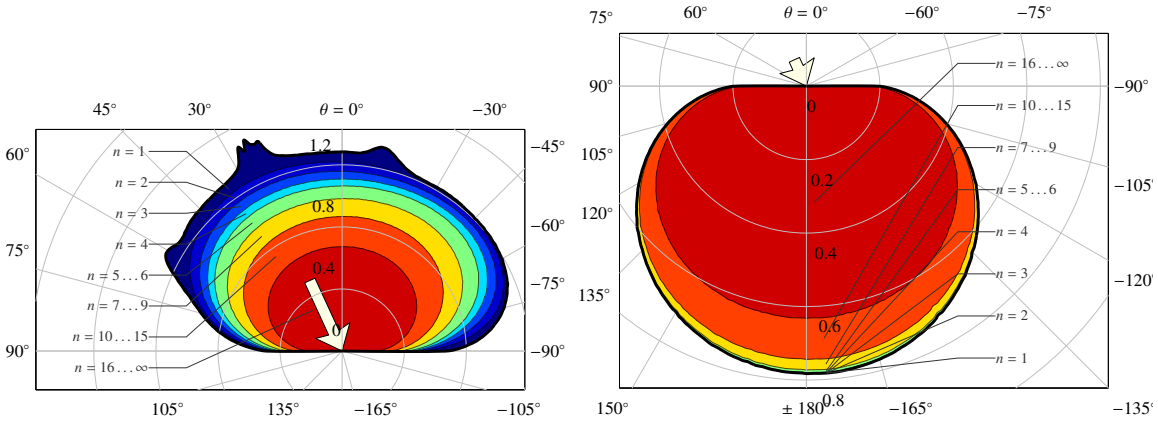
Figure 4.31 shows the BSDF C at a given viewpoint depth and for varying slab thicknesses. We can see that the contribution of the higher orders increase while that of the lower orders stays the same. As a result, the overall BSDF becomes slightly less anisotropic.



(a) $z_1 = 200\,\text{m}$.

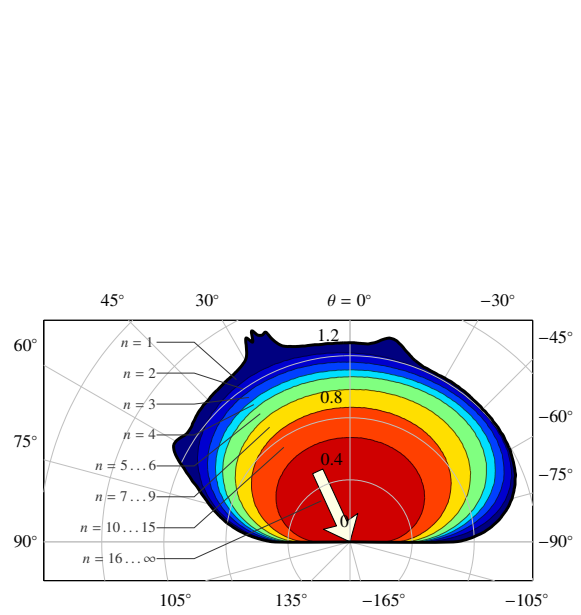


(b) $z_1 = 500\,\text{m}$.

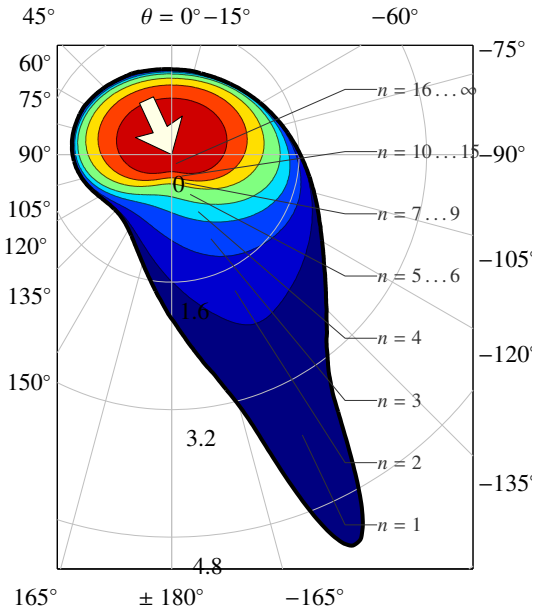


(c) $z_1 = 1000\,\text{m}$.

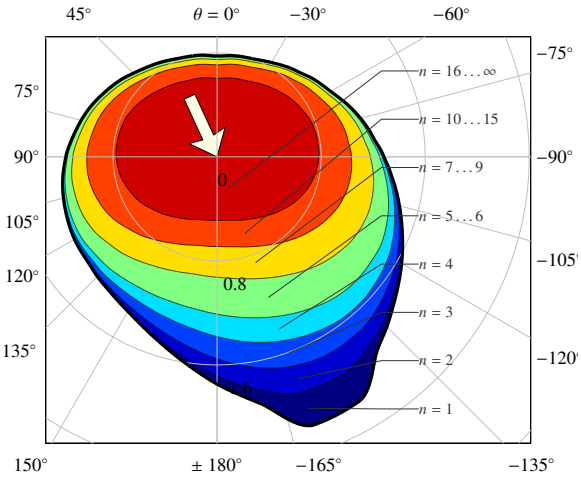
Figure 4.29: *Angular distribution of the BSDF for varying slab thicknesses.* $\psi = 0°, \theta' = 25°$. Left: $z = z_1$ *(reflectance).* Right: $z = 0$ *(transmittance).*



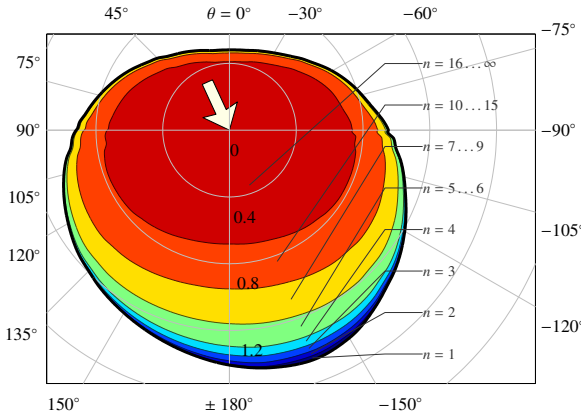
(a) $z' = 0\,\text{m}$.



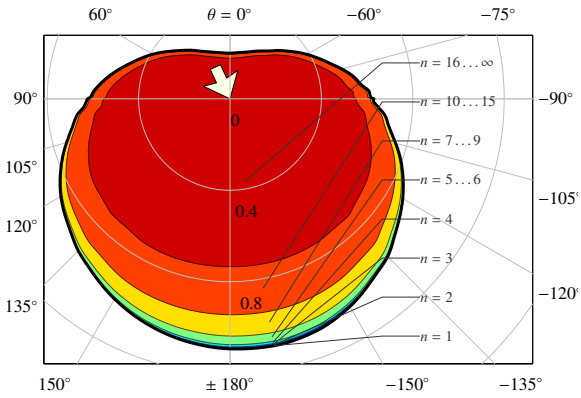
(b) $z' = 100\,\text{m}$.



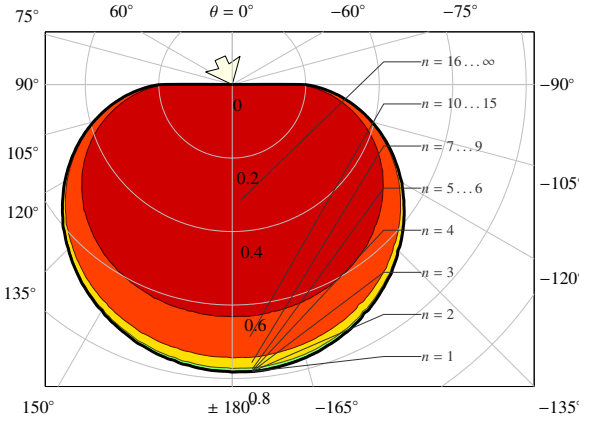
(c) $z' = 200\,\text{m}$.



(d) $z' = 300\,\text{m}$.

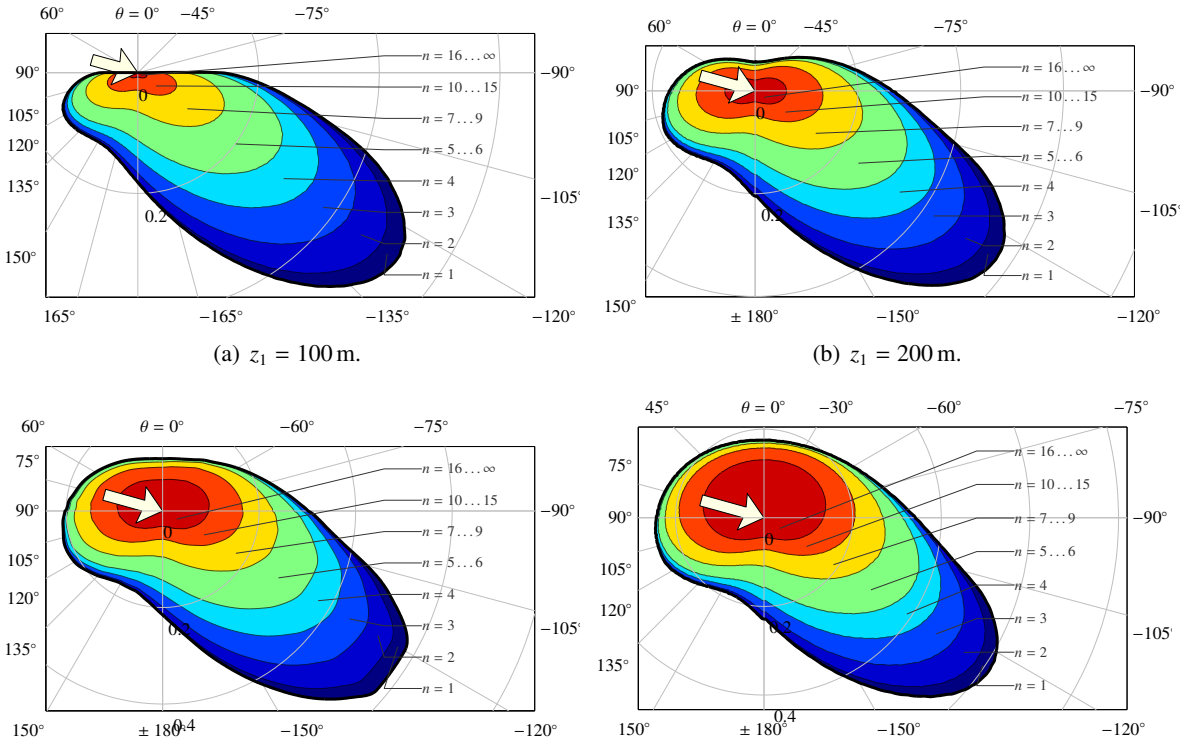


(e) $z' = 400\,\text{m}$.



(f) $z' = 500\,\text{m}$.

Figure 4.30: *Angular distribution of the BSDF for a given thickness and varying viewpoint depths.* $z_1 = 500\,\text{m}, \psi = 0°, \theta' = 25°$.



(a) $z_1 = 100\,\text{m}$. (b) $z_1 = 200\,\text{m}$.

(c) $z_1 = 300\,\text{m}$. (d) $z_1 = 500\,\text{m}$.

Figure 4.31: *Angular distribution of the BSDF at a given depth for varying slab thicknesses.* $z' = 100\,\text{m}, \psi = 0°, \theta' = 75°$.

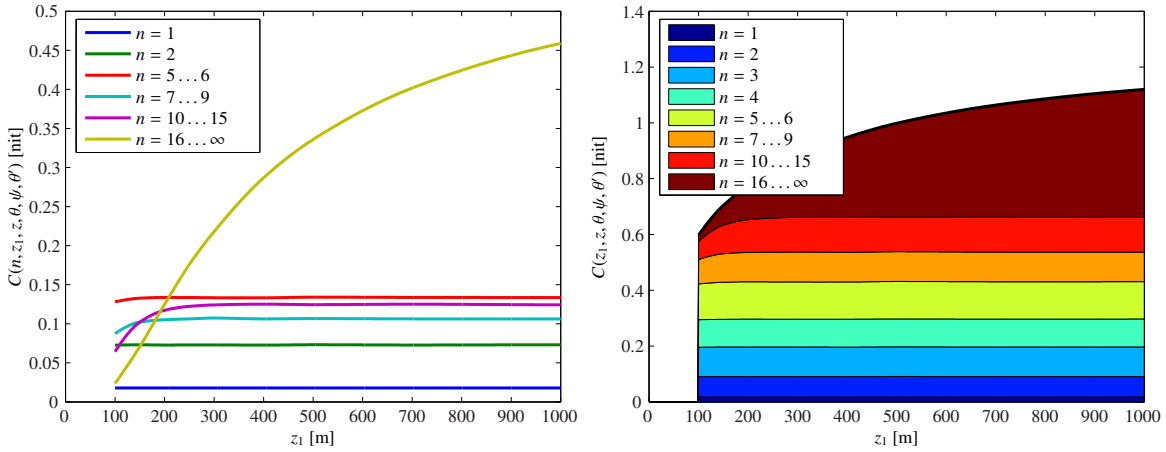
4.4.7.2 Behavior with respect to depth and thickness

We observe from the two previous analyzes that

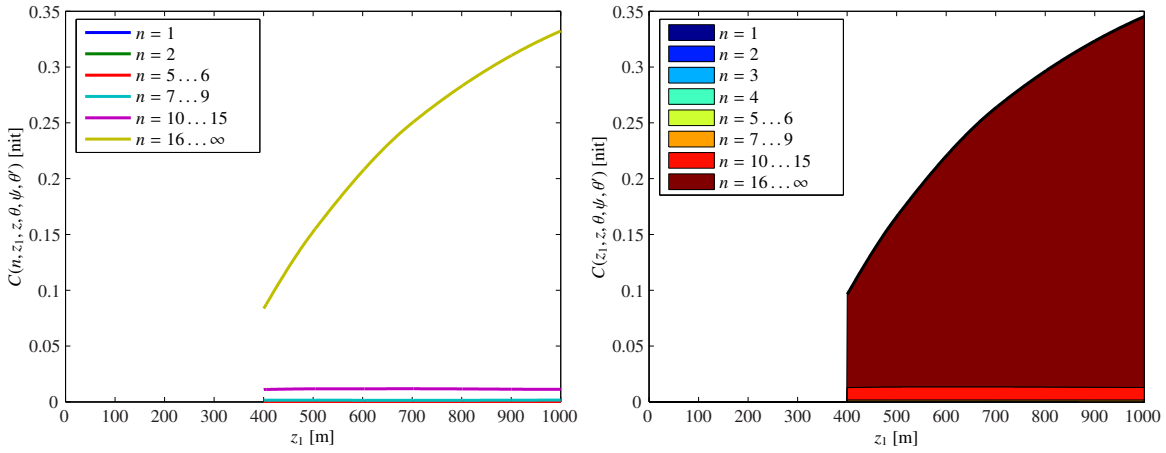
- When increasing the viewpoint depth, lower orders contribute less to the BSDF C while high orders contribute more. This is also the behavior of the transmittance with respect to the slab thickness.
- When increasing the slab thickness, lower orders keep their contribution to the BSDF C while the contribution of high orders increases. This is also the behavior of the reflectance with respect to the slab thickness.

These observations prompt us to study C against z' and z_1. Figures 4.32 and 4.33 show these behaviors. We observe that, indeed, they are very similar to that of the reflectance R and T. More precisely,

- All other parameters being constant, for a given order, the BSDF can only increase or stay constant when the thickness z_1 increases. As a result, the order-dependent has the same asymptotic behavior with z_1 than the one we observed for R. This observation generalizes this behavior for any viewpoint depth z' inside the cloud.
- A corollary of this observation is that the asymptotic value may even be reached for the lowest possible value of z_1. This is the case for low orders and/or high viewpoint depths (*e.g.*, Figure 4.32(a), orders 1 and 2).



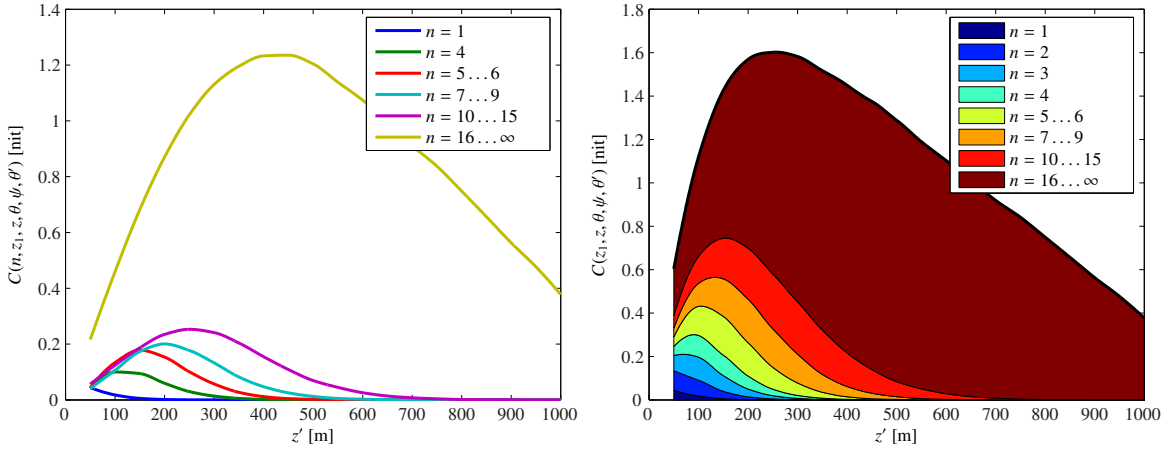
(a) $z' = 100\,\text{m}, \theta = 135°, \psi = 0°, \theta' = 25°$.



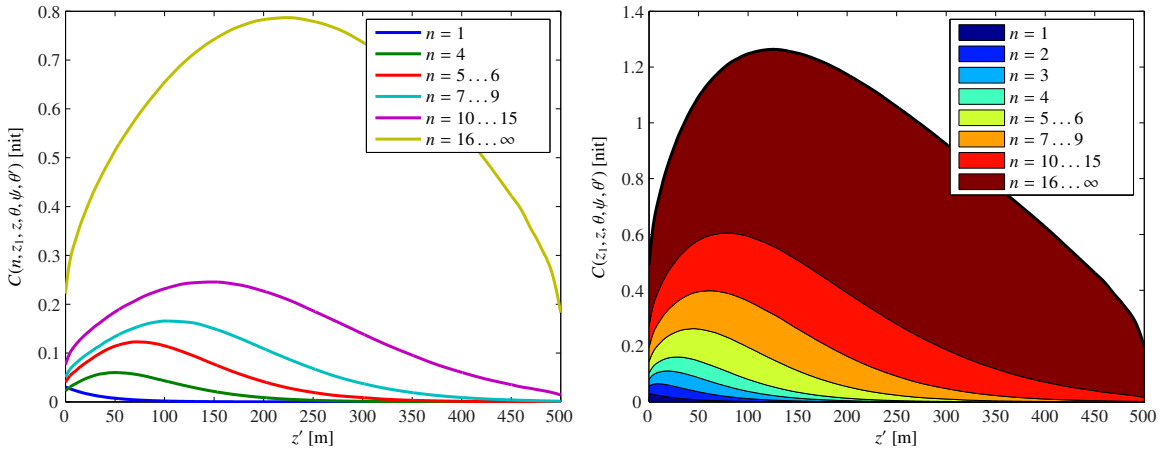
(b) $z' = 300\,\text{m}, \theta = 45°, \psi = 90°, \theta' = 67°$.

Figure 4.32: *Behavior of the BSDF C at a given viewpoint depth against the slab thickness z_1. Note the similarity with the behavior of reflectance (Figure 4.27). Values for $z_1 < z'$ do not exist since in these cases the point of view is outside the slab.*

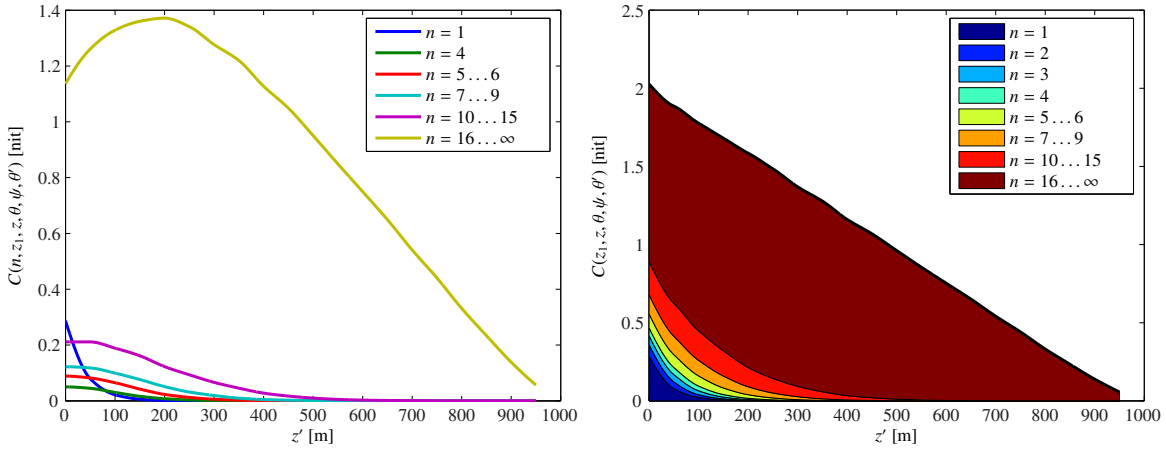
- All other parameters being constant, for a given order, the BSDF has roughly a lognormal behavior with respect to the viewpoint depth z'. That is, it increases up to a maximum value at a critical depth, then decreases. In some cases, it may decrease right from the start (*e.g.,* Figure 4.33, order 1).
- The decreasing rate of the total BSDF C against z' sometimes seems to be linear (*e.g.,* Figure 4.33(c), right).



(a) $z_1 = 1000\,\text{m}, \theta = 135°, \psi = 0°, \theta' = 25°$.



(b) $z_1 = 500\,\text{m}, \theta = 90°, \psi = 0°, \theta' = 25°$.



(c) $z_1 = 1000\,\text{m}, \theta = 0°, \psi = 0°, \theta' = 0°$.

Figure 4.33: *Behavior of the BSDF C in a slab of given thickness against the viewpoint depth z'. Note the similarity with the behavior of transmittance (Figure 4.27).*

4.4.7.3 Conclusions

The study of the BSDF C finally allows us to understand the angular distribution of light in a slab, and to relate it to real clouds. Here are the important observations we retain.

- The angular distribution of the BSDF is highly anisotropic for low orders. For high orders it is axially symmetric and tends towards isotropy. We measured that for $n > 15$, C is always axially symmetric.
- Since the reflectance of a slab is a mix of high and low orders paths, its angular distribution is a mix of diffusive reflection and anisotropic features.
- Since the transmittance of a slab is caused by low order paths for thin slabs and high order paths for thick slabs, its angular distribution is strongly anisotropic for thin slabs and strongly diffusive for thick slabs.
- The BSDF behavior against slab thickness is the same asymptotic behavior than that of the reflectance R.
- The BSDF behavior against viewpoint depth is almost the same lognormal behavior than that of the transmittance T.

Moreover, we can decompose the angular distribution in various components, more or less present depending on the conditions:

- A narrow forward peak (which is not captured by the chopped model) and a wider forward peak, whose contributions decrease with viewpoint depth.
- A pseudo-specular component appearing in the reflectance for grazing incident angles, which may be seen as a piece of the wide forward peak.
- The glory in direction $\Theta \approx 0^\circ$ and fogbow in direction $\Theta \approx 40^\circ$ due to single scattering, present only in the reflectance.
- A diffusive (*i.e.,* almost isotropic) part brought by paths of high orders, whose contribution increases with thickness.

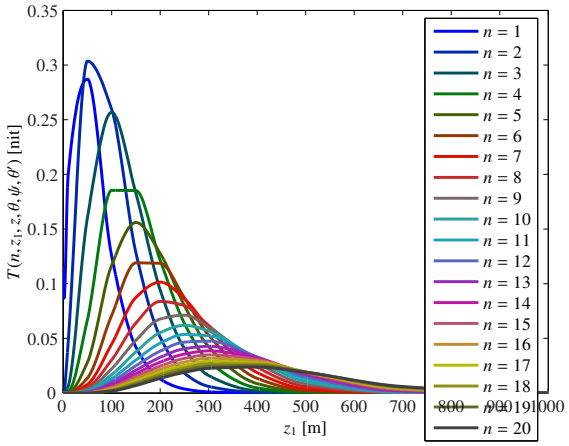
4.4.8 Order dependence

In each of the previous sections we have described the order dependence of the function we where studying. We recall in this section our observation on order dependence.

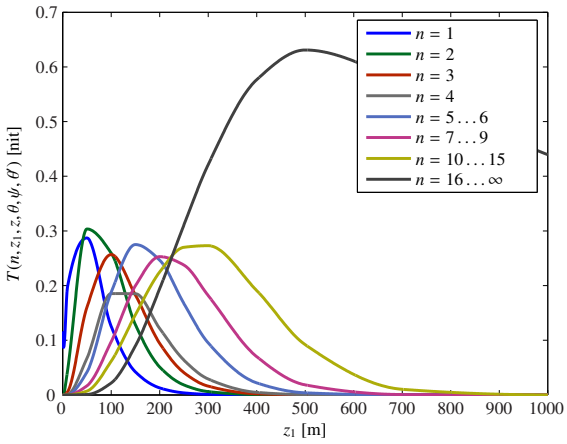
- The behavior of a group of consecutive orders is similar to the behavior of a single order of scattering.
- The width of the spatial distribution D of entry points increases with the order n.
- The contribution of high orders increases with the slab thickness z_1.
- The contribution of low orders decreases with the viewpoint depth z'.
- Paths of low order are mainly anisotropic in the forward direction.
- Paths of high order are mainly diffusive (*i.e.,* axially symmetric and close to isotropic).

As a result,

- In thin slabs, reflectance and transmittance are due to paths of low order. Reflectance is low and anisotropic, and transmittance is high and anisotropic.



(a) Note the asymptotic behavior of order-dependence. High orders are closer to each other than low orders.



(b) By grouping consecutive orders and making these groups larger for higher orders, we can see significant differences of behavior.

Figure 4.34: *Order-dependent behavior of the transmittance T against the slab thickness z_1.*

- In thick slabs, reflectance is due to paths of all orders and transmittance are due to paths of high order. Thus, reflectance is high and anisotropic, and transmittance is low and isotropic.
- For point of views inside the slab, the BSDF varies between these extrema.

Moreover, the order dependence itself is asymptotic. That is, the difference of behavior between two paths of consecutive order decreases with n. As an example, orders 1 and 2 have a very distinguishable behavior, whereas orders 14 and 15 behave very similarly (see Figure 4.34(a)).

From these observations, we deduced that we could study the behavior of groups of consecutive orders and make these groups larger for higher orders. We empirically chose the following groups

- $n = 1$,
- $n = 2$,
- $n = 3$,
- $n = 4$,
- $n = [5, 6]$,
- $n = [7, 9]$,
- $n = [10, 15]$,
- $n = [16, \infty]$.

The result of this grouping can be seen on Figure 4.34(b). Contrary to Figure 4.34(a), we can see significant differences between each group of orders. We have used this grouping in most of the figures shown in this chapter. Note that we grouped all diffusive orders ($n > 15$) in one single group.

4.4.9 Relation to real clouds

Thanks to this study, we can now explain the origin of the visual features of clouds.

- The view-dependent effects on cloud edges are the caused by the same phenomena that happens in thin slabs. That is,
  - Paths of low order dominate.
  - These path cause the light to exit rather forward than backward.
  - The angular distribution of these paths is highly anisotropic and forward dominant.
  - The spatial spreading of these paths is very small.

  As a result, the edges of clouds are much more bright on their unlit side than on their lit side, especially in the direction of the sun. The inhomogeneities of the edges affect light transport and convey details.
- The appearance of the base of clouds are caused by the same phenomena that happen at the bottom of thick slabs. That is,
  - Transmittance is low.
  - Paths of high order dominate.
  - The angular distribution of these paths is highly diffusive.
  - The spatial spreading of these paths is of the order of the cloud thickness.

  As a result, the base of a cloud is darker than the top. It has roughly the same luminance in all directions. The inhomogeneities in the cloud's density are smoothed by spatial spreading, blurring details and reducing contrast.
- The appearance of the top of clouds are caused by the same phenomena that happens at the top of thick slabs. That is,
  - Reflectance is high.
  - Light paths are of all orders.
  - The angular distribution is a mix of anisotropic (due to low orders) and diffusive (due to high orders) features.
  - The spatial spreading of these paths is small for low orders and large for high orders.

  As a result, the top of a cloud is much brighter than the bottom. It is generally bright in all directions, but the luminance varies with respect to the view direction. Specifically, the pseudo-specular effect of stratiform clouds and the dark edges of cumuliform clouds are a result of the strong anisotropy of paths of low order. The details in the clouds density are made visible by paths of low order.
- The anisotropy and power of the BSDF decreases quickly with the viewpoint depth. This is why we see a uniform grey-to-white color when flying through clouds.

Summary of §4.4 : We have analyzed the light transport in a slab against all input parameters. We have observed many remarkable behaviors that allow us to explain the origin of the visible features of clouds. We have seen that these behaviors seem rather simple and might be reproducible with a phenomenological approach.

Nous avons analysé le transport de la lumière dans une dalle en fonction de tous les paramètres d'entrée. Nous avons observé de nombreux comportements remarquables qui nous permettent d'expliquer l'origine des caractéristiques visuelles des nuages. Nous avons vus que ces comportements semblent relativement simples et pourraient être reproduites avec une approche phénoménologique.

4.5 Summary of Chapter 4

In this chapter, we designed (§4.2) a study of light transport in order to understand the mesoscopic behavior of light in clouds and find a simple model describing it. We developed (§4.3) an efficient light transport simulator to realize this study. By analyzing (§4.4) the result of our simulations, we managed to observe remarkable behaviors and link them with the visual features observed in Chapter 2. From our analysis we deduced the following general rules for light transport in clouds:

- Thin slabs ($z_1 < 500\,\mathrm{m}$) are more transmissive than reflective. Thicker slabs are more reflective than transmissive.
- We can decompose the BSDF in terms of contributions due to each order of scattering.
- Orders can be grouped in two main categories: low orders ($n < 16$) and high orders ($n \geq 16$).
- Low orders contribute to the anisotropic features in the BSDF.
- High orders contribute to the diffusive features in the BSDF.
- Paths of low order are short and their entry point on the cloud surface is located in a small area (local behavior).
- Paths of high order are long and their entry point on the cloud surface is located in a wide area (of the order of the slab thickness).
- The chopped-peak model approximates the reference model very well except in the particular case of multiple strong forward scattering.

These behaviors are simple, understandable, and it seems possible to model them phenomenologically. We can relate visual features we see in real clouds to the BSDF as follows:

- The view-dependent features we see in cloud edges are due to low-order paths, which make them appear dark on the lit side and bright on the unlit side. The local behavior of these paths makes these features detailed and contrasted.
- The visual aspect of the base of clouds is due to high order paths, that bring blurred, attenuated and diffusive light.
- The visual aspect of the top of clouds is a combination of both low and high order paths. High order bring an overall bright diffusive reflectance. Low orders bring details and view-dependent features.

Thanks to this analysis, we now know what are the limits of the multiple forward scattering approach and the diffusion approximation approach. The multiple forward scattering approach models low-order, anisotropic paths. The diffusive approach models high-order, diffusive paths. Both types of paths are significant in clouds. In reality, light is due to paths of very different types. In the edges of

cloud they are short, mainly forward scattering and almost not spread. At the base of clouds they are very long, chaotic, and very widely spread. At the top of clouds they are of the both types.

Finally, thanks to this analysis we also know what we have to do in order to reproduce faithfully all the visual features of clouds: model correctly the effect of these significant light paths. This is what we do in the rest of this thesis. In Chapter 5 we propose phenomenological models for the light transport in a slab reproducing the behaviors observed in this study. In Part III we present two models for rendering realistic clouds in realtime. The first one (Chapter 6) is designed for stratiform clouds, whose shape is close to that of a slab, and serves as a validation of all the deductions we draw here. The second one (Chapter 7) attempts to cover all kinds of cloud shapes by exploiting as much as possible the data generated by our study.

It is to note that similar studies have been done in previous works. In particular, there exist data and models for the directional radiance distribution in oceans [Tyl60, Mob94], which are often considered as semi-infinite slabs. Much work exist on the radiative transfer in a slab in the general case (see §2.5.5). However, we do not know of works on the light transport in a slab of cloud. In particular, we did not find any study on the spatial distribution of the entry point of light paths in slabs of clouds, let alone the order-dependence of this distribution. In addition, works on slabs in the general case are usually restricted to the order-independent exiting radiance (*i.e.,* they do not provide models for the radiance distribution inside the slab). As a result, we believe that the study presented in this chapter may contribute to the existing theoretical research on radiative transfer.

Dans ce chapitre, nous avons conçu (§4.2) une étude du transport de la lumière afin de comprendre le comportement mésoscopique de la lumière dans les nuages et de trouver un modèle simple le décrivant. Nous avons développé (§4.3) un simulateur efficace du transport de la lumière pour réaliser cette étude. En analysant (§4.4) les résultats de nos simulations, nous avons réussi à observer des comportements remarquables et à les relier aux caractéristiques visuelles observés en chapitre 2:

- *Les dalles fines ($z_1 < 500\,\mathrm{m}$) sont plus transmissives que réflectives. Les dalles plus épaisses sont plus réflectives que transmissives.*
- *On peut décomposer la BSDF en termes de contributions apportées par chaque ordre de diffusion.*
- *Les ordres peuvent être regroupés en deux grandes catégories: les petits ordres ($n < 16$) et les grands ordres ($n \geq 16$).*
- *Les parties anisotropes de la BSDF sont dues aux petits ordres de diffusion.*
- *Les parties isotropes de la BSDF sont dues aux grands ordres de diffusion.*
- *Les chemins dus aux petits ordres sont courts et leurs points d'entrée sur la surface du nuage est situé dans une aire réduite (comportement local).*
- *Les chemins dus aux grands ordres sont longs et leurs points d'entrée sur la surface du nuage est situé dans une aire large (de l'ordre de l'épaisseur de la dalle).*
- *Le modèle* chopped-peak *approxime le modèle de référence très bien mis à part dans le cas particulier de la diffusion multiple fortement en avant.*

Ces comportements sont simples, compréhensibles, et il semble possible de les modéliser de façon phénoménologique. Nous pouvons relier les caractéristiques visuelles que nous voyons dans les nuages réels avec la BSDF comme suit:

- *Les caractéristiques dépendant du point de vue que nous voyons sur les bords des nuages sont dus aux chemins de petits ordres, ce qui les rend sombre du côté illuminé et lumineux du côté non-illuminé. Le comportement local de ces chemins rend ces caractéristiques détaillées et contrastées.*
- *L'aspect visuel de la base des nuages est dû à des chemins d'ordre élevé, qui apportent une lumière floue et atténuée.*
- *L'aspect visuel du haut des nuages est une combinaison de chemins de tous ordres. Les grands ordres apportent une réflectance globalement forte et dispersée. Les petits ordres apportent les détails et les caractéristiques dépendant du point de vue.*

Grâce à cette analyse, nous savons maintenant quelles sont les limites de l'approche multiple forward scattering *et de l'approche par diffusion. L'approche* multiple forward scattering *modélise les chemins d'ordres faibles et anisotropes. L'approche par diffusion modélise les chemins d'ordre élevés et diffusifs. Ces deux types de chemins sont significatifs dans les nuages. En réalité, l'apparence des nuages est due à des chemins de types très différents. Sur les bords des nuages ils sont court, principalement diffusant vers l'avant et très peu étendus. A la base des nuages ils sont très longs, chaotiques et très étendus. En haut des nuages ils sont des deux types.*

En conséquence, grâce à cette analyse nous savons également ce que nous avons à faire afin de reproduire fidèlement toutes les caractéristiques visuelles des nuages: modéliser correctement les effets de ces chemins significatifs. C'est ce que nous faisons dans le reste de cette thèse. Dans le chapitre 5 nous proposons des modèles phénoménologiques du transport de la lumière dans une dalle reproduisant les comportements observés dans cette étude. Dans la partie III nous présentons deux modèles de rendu réaliste de nuages en temps-réel. Le premier (chapitre 6) est destiné aux nuages stratiformes, dont la forme est proche de celle d'une dalle. Le second (chapitre 7) tente de couvrir tous les types de nuages en exploitant autant que possible les données générées par notre étude.

Il est à noter que des études similaires ont été réalisées dans des travaux précédents. En particulier, il existe des données et des modèles concernant la distribution directionelle de la radiance dans les océans [Tyl60, Mob94], qui sont souvent considérés comme des dalles semi-infinies. Beaucoup de travaux existent sur le transfert radiatif dans une dalle dans le cas général (voir §2.5.5). Cependant, nous ne connaissons pas de travaux sur le transport lumineux dans une dalle de nuage. En particulier, nous n'avons pas trouvé d'étude sur la distribution spatiale des points d'entrée des chemins lumineux, sans compter la dépendance à l'ordre de diffusion de cette distribution. De plus, les travaux existants sur les dalles dans le cas général sont habituellement restreints à la radiance quittant la dalle (i.e., ils ne fournissent pas de modèle de la distribution de la radiance à l'intérieur de la dalle) et ne s'intéressent pas à la dépendance à l'ordre de diffusion. Ceci nous mène à penser que l'étude présentée dans ce chapitre peut être une contribution à l'état actuel de la recherche théorique sur le transfert radiatiff.

Chapter 5

Phenomenological models of light transport in a slab of cloud

Contents

5.1 Introduction

In Chapter 4, we have defined a case to study, in the form of light transport in a slab. We have computed and analyzed the output functions, which represent the luminous exitance, the surface distribution of entry points of light paths and its moments, the reflectance, the transmittance and the BSDF. We have seen that some of the behaviors of these functions seem to follow simple and invariable rules (*e.g.,* reflected exitance always increases with slab thickness).

In this chapter, we propose procedural formulations for these functions by making use of these analyses. The goal of these formulations is to reproduce efficiently the output of our simulations. They are phenomenological, *i.e.,* we do not try to rely on any microphysical formula to find them. However, since they are designed to match the output of our microphysics-based simulations, we can see them as empirical physical models.

Ideally, our models should be fast to compute and rely on as little precomputed data as possible, if no data at all. They should reproduce the result of our simulations faithfully (*i.e.,* with little error) for any given input parameters.

In this chapter, we propose three new models aimed at reproducing three of the functions we studied. In §5.2, we propose a new model for the luminous exitance M of a slab. In §5.3, we propose a new model for the first two moments of the spatial distribution D. In §5.4, we propose a new model for BSDF of a slab.

We will use these models in Part III where we propose two new approaches for the rendering of clouds. The first approach, aimed at stratiform clouds (Chapter 6), will make use of the exitance model (§5.2). The second approach, aimed at cumuliform clouds (Chapter 7), will make use of the models of §5.3 and §5.4.

5.2 Models of exitance

In this section we propose a model to compute the total exitance of a slab of cloud (§5.2.1), and a model to compute the order-dependent exitance (§5.2.2). These models are based on our study of exitance described in §4.4.5.

5.2.1 Total exitance

The doubling-adding method (§2.5.5.6) gives an analytical solution for the reflected and transmitted exitance of a slab. However, it is suited only for particles with an isotropic phase function and lit by a diffuse source. We propose a modification of the doubling-adding approach that approximates the exitances of a slab with the Mie phase function and under directional lighting conditions.

The original doubling-adding scheme is

$$\begin{aligned} M_R([1,\infty], z_1, \mu') &= 1 - \gamma(z_1), \\ M_T([1,\infty], z_1, \mu') &= \gamma(z_1), \\ \gamma(z_1) &= \frac{\gamma_{\text{ref}}}{z_1 - (z_1 - z_{\text{ref}})\gamma_{\text{ref}}}, \end{aligned}$$

from Equations 2.38, 2.39 and 2.40. The parameter γ_{ref} is measured for a slab of given thickness z_{ref}. This reference thickness can *normally* be any thickness.

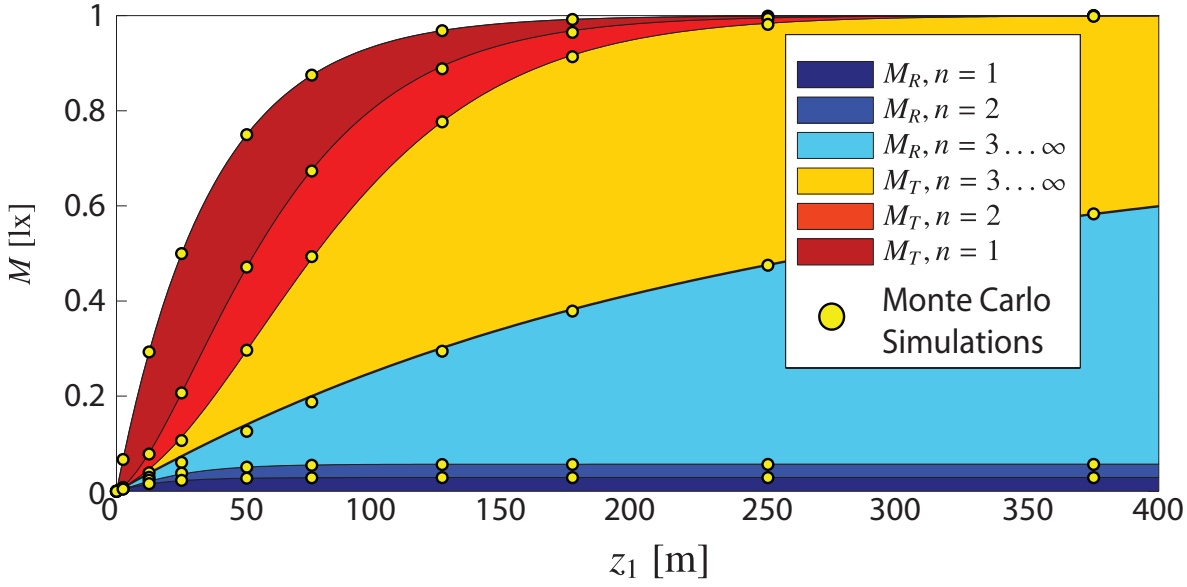


Figure 5.1: *Area graph of our procedural model for the exitances of a slab. The dots represent the values obtained with our Monte Carlo simulations.* $\theta' = 30°$.

As we have seen in our study (§4.4.5), the exitance in a slab of cloud differs from the exitance in a slab of particles of isotropic phase function in the following ways:

- If γ_{ref} is measured on a thick slab ($z_{\text{ref}} > 500\,\text{m}$), then in thin slabs the transmitted exitance is higher than expected.
- Conversely, if γ_{ref} is measured on a thin slab ($z_{\text{ref}} < 500\,\text{m}$), then in thick slabs the transmitted exitance is lower than expected.
- For grazing incident angles, the transmitted exitance is lower and the reflected exitance is higher than for an orthogonal incident angle.

We propose to add a correction function deduced from these observations as follows. We measure γ_{ref} on a thin slab ($z_{\text{ref}} = 1\,\text{m}$) and we define γ by

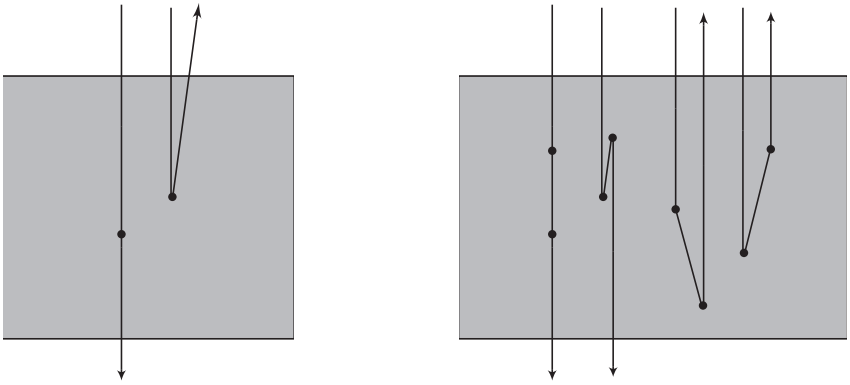
$$\gamma(z_1, \mu') = F(z_1, \mu') \frac{\gamma_{\text{ref}}}{z_1 - (z_1 - z_{\text{ref}})\gamma_{\text{ref}}}, \tag{5.1}$$

with the correction function F defined by

$$F(z_1, \mu') = b(\mu') + (1 - b(\mu'))\, e^{-z_1 c(\mu')}, \tag{5.2}$$

where $b(\mu')$ and $c(\mu')$ are the parameters of the correction function. The correction function reads as follows. The first term b corrects the error for thick slabs between the exitance predicted by the original doubling-adding function and the exitance measured in our study. Since this error decreases on thin slabs, we add a second term $(1 - b(\mu'))\, e^{-z_1 c(\mu')}$. This second term overrides the first term for low values of z_1 and decreases to 0 with high values of z_1. The rate at which this second term decreases is determined by c. Since the behavior of a slab varies with the incident angle, we make these parameters dependent on θ'.

We fit these parameters against the output $M_T(z_1, \theta')$ and $M_R(z_1, \theta')$ of our study. The results are shown in Table 5.1. We also find $\gamma_{\text{ref}} = 0.9961$ for $z_{\text{ref}} = 1\,\text{m}$ and $\theta' = 0°$ by fitting. Figure 5.1 shows



(a) Single scattering exitance.

(b) Double scattering exitance.

Figure 5.2: *In a slab, we consider either upward fluxes or downward fluxes.*

the result of our model compared with the result of our Monte-Carlo simulations. As we can see, our procedural model fits the reference model quite well (compare with Figure 4.24(b)).

As we can see on Table 5.1, $b > 1$ when $\theta' < 35°$ and $b < 1$ when $\theta' > 35°$. Physically, this means that if the slab is lit from an orthogonal incident angle, more light will traverse it than expected by the doubling-adding method. Conversely, if the slab is lit from a grazing angle, more light will be reflected. This corresponds to what we observed in our study.

5.2.2 Order-dependent exitance

In this section we propose a model to compute the exitance of orders 1 (single scattering) and 2 (double scattering). These models follow the doubling-adding approach in the sense that a slab can be considered as a 1D column of medium with only two existing directions: up and down.

When considering a flux, we consider its direction as being either upward or downward. Similarly, when hitting a droplet, scattering deviates this flux either up or down. Using this paradigm, we apply the laws of the successive orders of scattering (§2.5.4.3) and find analytical solutions.

By placing ourselves in one dimension, the reflected and transmitted exitance functions are rather simple to compute. Let us consider a 1D column of scattering medium of extinction coefficient κ_c. A flux in this column can either be scattered backward with a coefficient r or scattered forward with a

Table 5.1 Values of the parameters for our procedural exitance models.

θ_l	b	c	κ_c	t	r
0°	1.1796	0.0138	0.0265	0.8389	0.0547
10°	1.1293	0.0154	0.0262	0.8412	0.0547
20°	1.1382	0.0131	0.0272	0.8334	0.0552
30°	1.0953	0.0049	0.0294	0.8208	0.0564
40°	0.9808	0.0012	0.0326	0.8010	0.0603
50°	0.9077	0.0047	0.0379	0.7774	0.0705
60°	0.7987	0.0207	0.0471	0.7506	0.0984
70°	0.6629	0.0133	0.0616	0.7165	0.1700
80°	0.5043	0.0280	0.0700	0.7149	0.3554
90°	0.3021	0.0783	0.0700	0.1000	0.9500

coefficient t (see Figure 5.2(a)). The exitance due to single scattering can be computed by

$$\begin{aligned} M_R(1, z_1, \mu') &= \int_0^{z_1} e^{-\kappa_c z} t \kappa_c e^{-\kappa_c (z_1 - z)} dz \\ &= t z_1 \kappa_c e^{-\kappa_c z_1}, \end{aligned} \tag{5.3}$$

$$\begin{aligned} M_R(1, z_1, \mu') &= \int_0^{z_1} e^{-\kappa_c z} r \kappa_c e^{-\kappa_c z} dz \\ &= \frac{1}{2} r (1 - e^{-2\kappa_c z_1}). \end{aligned} \tag{5.4}$$

In the case of double scattering (see Figure 5.2(b)), transmitted exitance can be the cause of either double-backward or double-forward scattering

$$\begin{aligned} M_T(2, z_1, \mu') &= \int_0^{z_1} e^{-\kappa_c z} t \kappa_c \int_0^{z_1 - z} e^{-\kappa_c z'} \kappa_c t e^{-\kappa_c (z_1 - z')} dz' dz \\ &\quad + \int_0^{z_1} e^{-\kappa_c z} r \kappa_c \int_0^{z} e^{-\kappa_c (z - z')} \kappa_c r e^{-\kappa_c (z_1 - z')} dz' dz \\ &= \frac{1}{2} (\kappa_c t z_1)^2 e^{-\kappa_c z_1} + \frac{1}{4} r^2 \left(1 + (2\kappa_c z_1 - 1) e^{-2\kappa_c z_1}\right) e^{-3\kappa_c z_1}. \end{aligned} \tag{5.5}$$

Reflected exitance from double scattering happens when the flux is scattered backward then forward, or vice-versa

$$\begin{aligned} M_R(2, z_1, \mu') &= 2 \int_0^{z_1} e^{-\kappa_c z} r \kappa_c \int_0^{z} e^{-\kappa_c (z - z')} \kappa_c t e^{-\kappa_c z'} dz' dz \\ &= \frac{1}{2} r t \left(1 - (2\kappa_c z_1 + 1) e^{-2\kappa_c z_1}\right). \end{aligned} \tag{5.6}$$

Here again r, t and κ_c depend on θ' and are found by fitting Equations 5.3 – 5.6 against the output $M_T(1, z_1, \theta')$, $M_R(1, z_1, \theta')$, $M_T(2, z_1, \theta')$ and $M_R(2, z_1, \theta')$ of our study. Table 5.1 shows the results of this fitting.

Order-dependent models for higher orders of scattering can be found in the same way, although we did not pursue this model for higher orders. Moreover, since we have a model for the total exitance (§5.2.1), we can also compute the exitance due to paths or order $n = [3, \infty]$ simply by

$$M_R([3,\infty], z_1, \mu') = M_R([3,\infty], z_1, \mu') - M_R(1, z_1, \mu') - M_R(2, z_1, \mu'), \quad (5.7)$$
$$M_T([3,\infty], z_1, \mu') = M_T([3,\infty], z_1, \mu') - M_T(1, z_1, \mu') - M_T(2, z_1, \mu'). \quad (5.8)$$

Figure 5.1 shows the result of our model compared with the result of our Monte-Carlo simulations. As we can see, our procedural model fits the reference model quite well (compare with Figure 4.24(b)).

5.3 Models for the moments of spatial distribution

In this section we propose a model to compute the first two moments of the order-dependent distribution of entry points of light paths on the lit boundary of the slab. It is based on our study of the moments described in §4.4.4.

5.3.1 First moment

We write the first moment $\boldsymbol{c}(n, z_1, z, \theta, \psi, \theta') = (x, y, z_1)$. When studying x and y, we have observed that its behavior is slightly logarithmic against the viewpoint depth z' and depends mainly on the view and incident angles. We have thus used logarithmic and trigonometric functions to fit this behavior. We found the following model:

$$x = a_x \log(1 + ez') + b_x, \quad (5.9)$$
$$y = a_y \log(1 + ez') + b_y, \quad (5.10)$$
$$a_x = f \sin\psi \sin(g\theta'), \quad (5.11)$$
$$b_x = h \sin\psi \sin(\theta'), \quad (5.12)$$
$$a_y = i + j\left(\cos\psi \sin(k\theta') + l\theta'\right), \quad (5.13)$$
$$b_y = m + o\cos\psi, \quad (5.14)$$
$$e = E(n, \mu), \quad (5.15)$$
$$f = F(n, \mu), \quad (5.16)$$
$$g = G(n, \mu), \quad (5.17)$$
$$h = H(n, \mu), \quad (5.18)$$
$$i = I(n, \mu), \quad (5.19)$$
$$j = J(n, \mu), \quad (5.20)$$
$$k = K(n, \mu), \quad (5.21)$$
$$l = L(n, \mu), \quad (5.22)$$
$$m = M(n, \mu), \quad (5.23)$$
$$o = O(n, \mu, \mu'). \quad (5.24)$$

This model is parameterized through 10 input parameters $e \ldots o$. The first nine depend only on the order n and the view angle μ while the last one depends also on the incident angle μ'. Note that the

slab thickness is not involved, since we found no strong dependence of $\boldsymbol{c}$ on it. Like for our exitance model, we fit these parameters against our simulation results.

Ironically, the behavior of this mean entry point does not seem to follow closely trigonometric rules. As a result, the model we found here is rather complicated. It is probable that a better model can be found. Nevertheless, this model fits our needs: it is fast to compute and its parameters are at most 3-dimensional.

5.3.2 Second moment

The second moment of D represent its standard deviation. We simplify the problem by considering this standard deviation is symmetric, *i.e.*, $\sigma_x = \sigma_y = \sigma$. When studying σ_x and σ_y, we observed that their behavior is logarithmic against the viewpoint depth z' and does not depend on the view and incident angles. Thus again, we have used logarithmic and trigonometric functions to fit this behavior. We found the following model:

$$\begin{aligned}
\sigma(n, z_1, z, \theta, \psi, \theta') &= o + q z_1 \log(1 + r z') + s \log(1 + t z'), && (5.25)\\
p &= P(n), && (5.26)\\
q &= Q(n), && (5.27)\\
r &= R(n), && (5.28)\\
s &= S(n), && (5.29)\\
t &= T(n). && (5.30)
\end{aligned}$$

This model is parameterized through 5 input parameters $p \dots t$. Interestingly, these parameters depend only on the order of scattering, which makes this model very light in memory usage. As for the other models, we fit these parameters against the results of our simulations.

5.4 BSDF model

In this section we propose a model to compute the 6-DOF order-dependent BSDF $C(n, z_1, z, \theta, \psi, \theta')$ of a slab. It is based on our study of the BSDF described in §4.4.7.

5.4.1 Core function

We base our model on the behavior of C against z' for a given thickness z_1 and a high order $n > 15$ (so that we are in the axially symmetric case, which allows us to ignore ψ for the moment). As described in §4.4.7, we have observed that C either first increases then decreases with z', or decreases with z' from the start. We mimic this behavior using the function

$$C(n, z_1, z, \theta, \psi, \theta') = a \frac{\log(z' + d)}{\log(b + d)} e^{-\frac{(z'-b)^2}{2c^2}}, \qquad (5.31)$$

where a, b, c, d are parameters encoding the four main aspects of the function and depend on the other input values $(n, z_1, \theta, \theta')$, *i.e.,*

$$a = A(n, z_1, \theta, \theta'), \tag{5.32}$$
$$b = B(n, z_1, \theta, \theta'), \tag{5.33}$$
$$c = C(n, z_1, \theta, \theta'), \tag{5.34}$$
$$d = D(n, z_1, \theta, \theta'). \tag{5.35}$$

The parameter b corresponds to the critical depth z' at which C starts to decrease. It can be negative or null if C is monotonic (*i.e.,* only decreases with z'). The parameter $a > 0$ encodes the maximum value of C, *i.e.,* the value $C(n, z_1, b, \theta, \psi, \theta')$. The parameter $c > 0$ encodes the broadness of the peak around b. Finally, the parameter $d \geq 1$ describes how "logarithmic" the behavior is. This function resulted in a good fitting in all cases.

5.4.2 Angle dependence

In a second step, we have observed the behavior of C against z' when keeping z_1 constant and varying θ and θ'. We have observed that the shape of the curve was smoothly varying with θ'. In particular, we have found that c and d almost did not change with θ', and that a and b were varying slowly with θ'.

As a result, we have decided to have c and d not depend on θ', and to split the expression of a and b as

$$a = A_1(n, z_1, \theta) \times A_2(n, z_1, \theta'), \tag{5.36}$$
$$b = B_1(n, z_1, \theta) - B_2(n, z_1, \theta'), \tag{5.37}$$
$$c = C(n, z_1, \theta), \tag{5.38}$$
$$d = D(n, z_1, \theta), \tag{5.39}$$

Since there can be an infinity of pairs of solutions for A_1 and A_2 and B_1 and B_2, we add the constraint

$$A_2(n, z_1, 0) = 1, \tag{5.40}$$
$$B_2(n, z_1, \theta') = 0. \tag{5.41}$$

We used non-linear least-square fitting using matlab to fit all the functions A_1, A_2, B_1, B_2, C, D against the results of our simulations for $n = [16, \infty]$, and it has shown to result in a decent fitting. As a result, we have

$$C([16, \infty], z_1, z, \theta, \psi, \theta') = a \frac{\log(z' + d)}{\log(b + d)} e^{-\frac{(z'-b)^2}{2c^2}}, \tag{5.42}$$
$$a = A_1([16, \infty], z_1, \theta) \times A_2([16, \infty], z_1, \theta'), \tag{5.43}$$
$$b = B_1([16, \infty], z_1, \theta) - B_2([16, \infty], z_1, \theta'), \tag{5.44}$$
$$c = C([16, \infty], z_1, \theta), \tag{5.45}$$
$$d = D([16, \infty], z_1, \theta). \tag{5.46}$$

This formulation is parameterized through six 2-DOF functions. The shape of the function A_1 is very similar to that of Chandrasekhar's X function [Cha60]. This is not a coincidence since both encode the amplitude of radiance against θ', all others things being equal.

5.4.3 Anisotropy

For lower orders, C also depends on ψ, which makes things more complicated. We have made the hypothesis that this anisotropy could be encoded by a function depending only on n and Θ, *i.e.,*

$$C(n, z_1, z, \theta, \psi, \theta') = pa\frac{\log(z'+d)}{\log(b+d)}e^{-\frac{(z'-b)^2}{2c^2}}, \tag{5.47}$$

$$a = A_1(n, z_1, \theta) \times A_2(n, z_1, \theta'), \tag{5.48}$$

$$b = B_1(n, z_1, \theta) - B_2(n, z_1, \theta'), \tag{5.49}$$

$$c = C(n, z_1, \theta), \tag{5.50}$$

$$d = D(n, z_1, \theta), \tag{5.51}$$

$$p = P(n, \Theta). \tag{5.52}$$

Again, we found the seven parameterizing functions $a \dots p$ by fitting them against the results of our simulations for each set of orders $n \in \{2, 3, [4, 5], [6, 8], [9, 14]\}$. We found our model to approximate the results fairly well. Note the similarity between P and the multiply-scattered phase function of Tessendorf and Wasson [TW03]. Indeed, as n increases, P becomes more isotropic. The case $n = [16, \infty]$ can be seen as a case where $P(n, z_1, \Theta) = 1$.

Figures 5.3 – 5.4 pages 176 – 177 show some results of this model compared with the results of our simulations. We can see that our procedural model fits the simulations quite well.

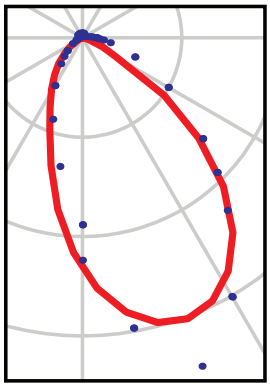
5.5 Summary of Chapter 5

In this chapter we have presented three phenomenological models for the light transport in a slab. We found these models thanks to our study of light transport presented in Chapter 4. These models include a model of the exitance of a slab (§5.2), a model of the moments of the spatial distribution of entry points of light paths on the lit surface of a slab (§5.3) and a model of the BSDF of a slab (§5.4).

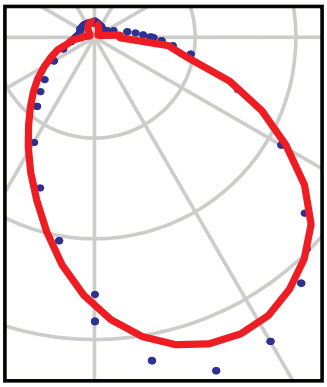
We use the first model in a new approach for the rendering of stratiform clouds in Chapter 6. The two other models are used for the rendering of cumuliform clouds in Chapter 7.

Dans ce chapitre nous avons présenté trois nouveaux modèles phénoménologiques du transport de la lumière dans une dalle. Nous avons découvert ces modèles grâce à notre étude du transport lumineux présentée dans le chapitre 4. Ces modèles décrivent l'exitance d'une dalle (§5.2), les moments de la distribution spatiale des points d'entrée des chemins lumineux sur la surface illuminées d'une dalle (§5.3) et la BSDF d'une dalle (§5.4).

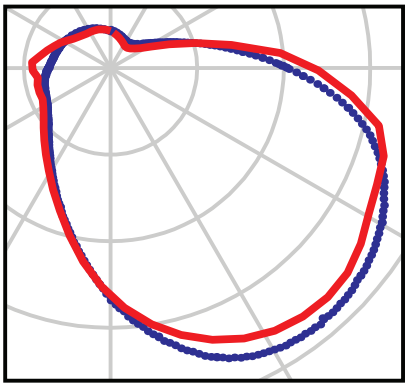
Nous utilisons le premier modèle dans une nouvelle méthode de rendu de nuages stratiformes dans le chapitre 6. Les deux autres modèles sont utilisés pour le rendu de nuages cumuliformes au chapitre 7.



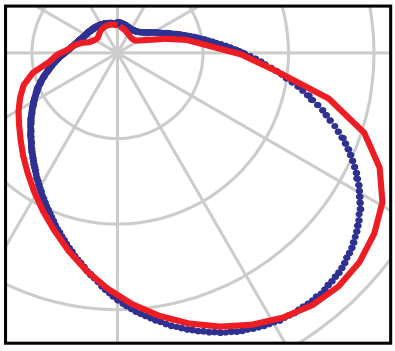
(a) $z_1 = 500\,\text{m}, z' = 100\,\text{m}, n = 2$.



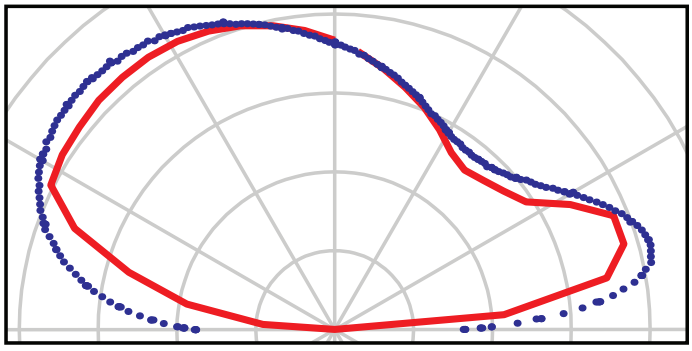
(b) $z_1 = 500\,\text{m}, z' = 100\,\text{m}, n = 3$.



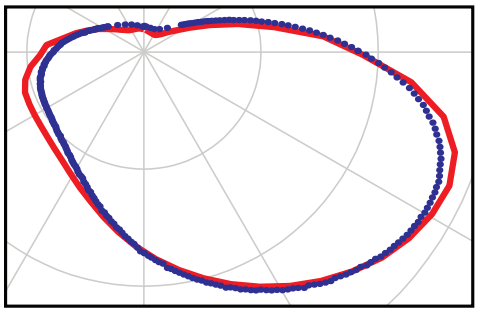
(c) $z_1 = 100\,\text{m}, z' = 50\,\text{m}, n = 3$.



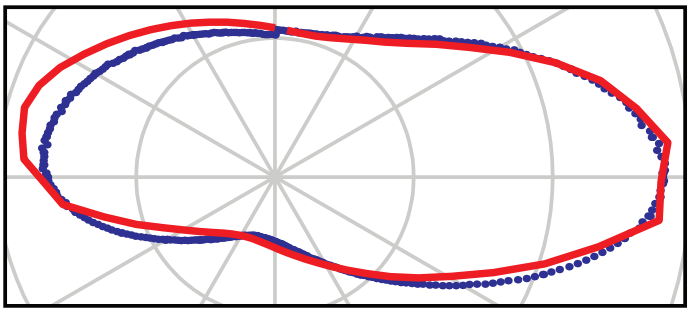
(d) $z_1 = 200\,\text{m}, z' = 100\,\text{m}, n = 4$.



(e) $z_1 = 500\,\text{m}, z' = 0\,\text{m}, n = 2$.

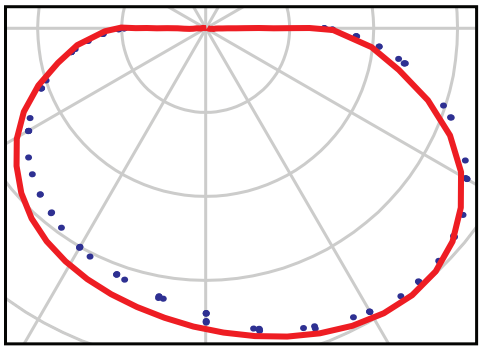


(f) $z_1 = 100\,\text{m}, z' = 70\,\text{m}, n = 4$.

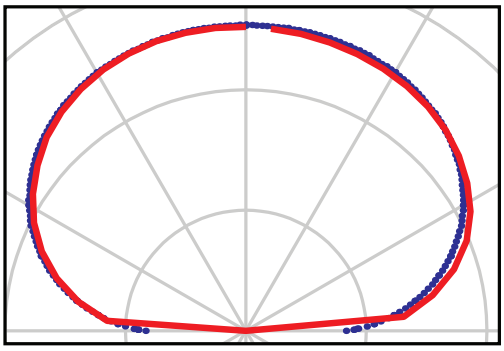


(g) $z_1 = 200\,\text{m}, z' = 50\,\text{m}, n = [5, 6]$.

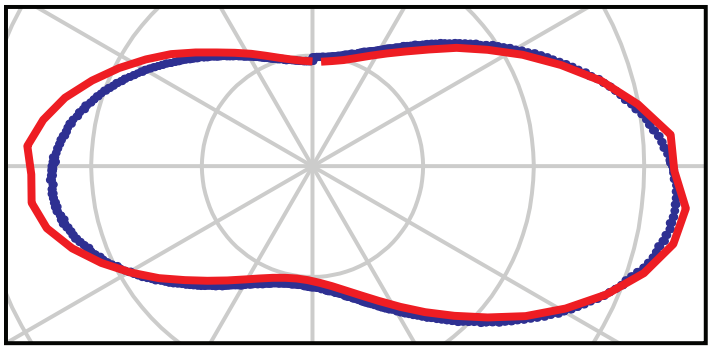
Figure 5.3: *Some results of our procedural BSDF model (in red) compared to the results of our simulations (in blue)* $\psi = 0°, \theta' = 25°$



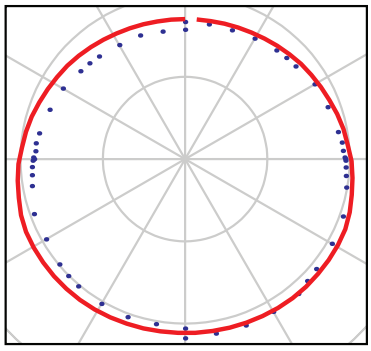
(a) $z_1 = 200\,\text{m}, z' = 200\,\text{m}, n = [7, 9]$.



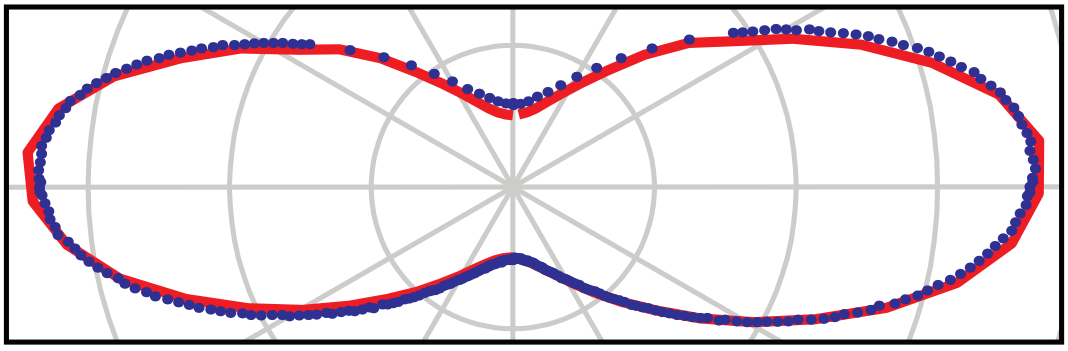
(b) $z_1 = 500\,\text{m}, z' = 0\,\text{m}, n = [16, \infty]$.



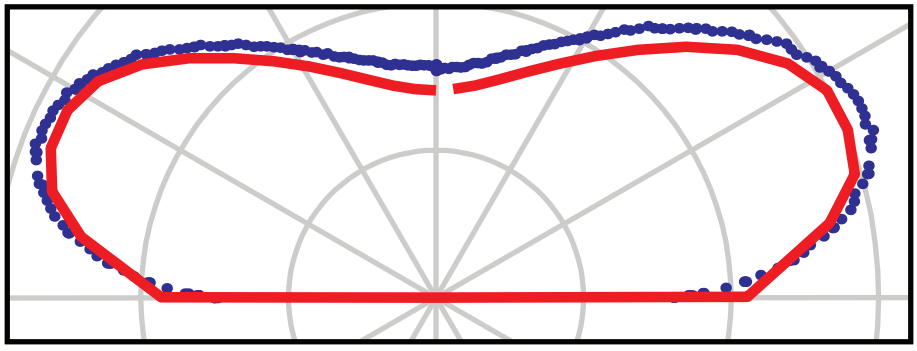
(c) $z_1 = 200\,\text{m}, z' = 100\,\text{m}, n = [7, 9]$.



(d) $z_1 = 1500\,\text{m}, z' = 600\,\text{m}, n = [16, \infty]$.



(e) $z_1 = 100\,\text{m}, z' = 50\,\text{m}, n = [10, 15]$.



(f) $z_1 = 100\,\text{m}, z' = 0\,\text{m}, n = [10, 15]$.

Figure 5.4: *Some results of our procedural BSDF model (in red) compared to the results of our simulations (in blue)* $\psi = 0°, \theta' = 25°$

Part III

Two new rendering methods for clouds


YOU KNOW YOU'VE BEEN ON THE COMPUTER TOO MUCH WHEN YOU WONDER WHICH CHEAP PHOTOSHOP FILTER WAS USED TO MAKE THE PRETTY CLOUDS IN THE SKY.

Chapter 6

A rendering method for stratiform clouds

Contents

6.1 Introduction

In this chapter we present a model for the rendering and shading of stratiform clouds that draws on the results of our study of light transport in clouds (Part II). To do so, we propose a new approach for computing light transport that reproduces both low order and high order light paths. This model is designed to run in realtime on current graphics hardware. To simplify the problem, we make several important assumptions:

- We focus on stratiform clouds, which have the shape of a layer. This way, the shape of the clouds is close to that of the reference slab used in our study. We expect from this that light transport in this cloud behave similarly to light transport in slabs.
- We assume the point of view is either above or below the clouds, and not at the same level. That is, we do not look at the clouds from the side or the inside.
- We consider that the clouds are homogeneous. This simplifies their representation, and again makes them close to the reference shape.
- We do not take spatial spreading into account. That is, we do not consider the surface distribution of entry points of light paths in this model.

The restriction on the shape will limit us in the type of clouds we can render. The other approximations (homogeneity and no spreading) are bound to have visual consequences on the result of our method. However, as we will see, they shall not prevent us from reproducing all the visual features we observe in clouds, which is our main goal. The features we reproduce are

- Highly detailed, contrasted, view-dependent cloud edges.
- Diffusive, view-independent cloud cores.
- Pseudo-specular effect on seas of clouds.
- Sky and ground illumination on clouds.
- Ground-clouds inter-reflections.

Moreover, we allow the animation of our clouds. None of the previous approaches provides all of these features, may they be offline or realtime. This model has been published in [BNL06].

We begin by explaining the way we represent the cloud shape in §6.2. We then describe our new light transport model in §6.3. We give some information on the implementation of this model on GPU in §6.4. We then show the results in §6.5. We discuss the results, limitations and applications of this model in §6.6.

6.2 Cloud shape and optical parameters

We define our clouds much like the slab of Chapter 4. The droplet concentration ρ is considered homogeneous and equals $300\,\mathrm{cm}^{-3}$. We use the modified gamma DSD with $\gamma = 2$ and $r_e = 7\,\mu\mathrm{m}$. We inferred these values from [MVC00]. Unlike a slab, here the thickness z_1 is varying along x and y. We simply assume that its variation is slow enough so that the layer can be locally considered as a slab. As a result we describe by a simple 2D height map. This representation is much more memory efficient than the classical 3D density grid and allows realtime animation. Since clouds usually have a flat bottom, we choose to fix the bottom boundary at a given constant height h above the ground, and

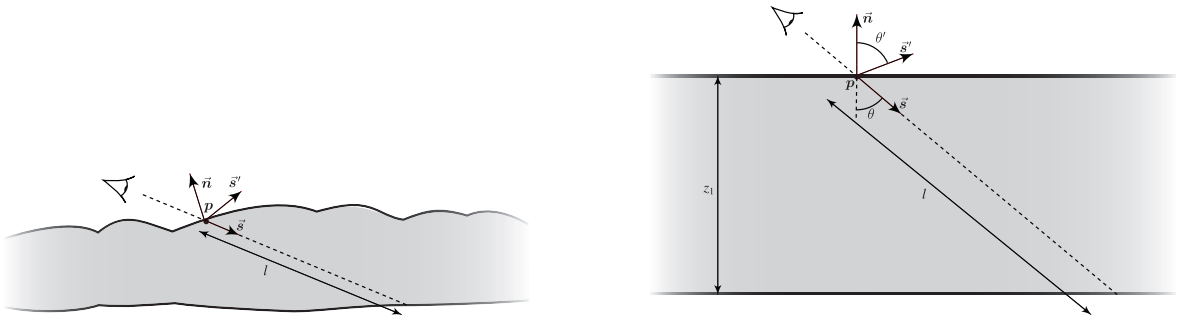


Figure 6.1: *Principle of our model. A stratiform cloud is locally considered as a slab aligned with the normal of the viewed boundary.*

the top boundary at $h + z_1(x, y)$. However, note that our method does not restrict the bottom boundary to be flat.

We use the chopped-peak model as described in §4.4.2. We show in §6.3 how we make up for the loss of the multiple strong forward scattering.

6.3 Light transport model

6.3.1 Overview

For a given viewpoint $\boldsymbol{p}$ and view direction $\vec{\boldsymbol{s}}$ for which to compute luminance $L(\boldsymbol{p}, \vec{\boldsymbol{s}})$, we measure the distance l along $\vec{\boldsymbol{s}}$ to the bottom boundary of the cloud and the normal $\vec{\boldsymbol{n}}$ of the surface at point $\boldsymbol{p}$. From these values we place ourselves in the reference shown on Figure 6.1. We consider the slab of normal $\vec{\boldsymbol{n}}$ and thickness $z_1 = l \cos\theta$. The view and illumination directions $\vec{\boldsymbol{s}}$ and $\vec{\boldsymbol{s}}'$ are represented by spherical coordinates (θ, ψ, θ') or (μ, ψ, μ'). We need to solve L for each visible point $\boldsymbol{p}$ on the boundaries of the clouds in direction $\vec{\boldsymbol{s}}$ of the camera.

Here again, we rely on the BSSSDF paradigm. That is, given the boundary conditions, which are the incoming radiance on the boundaries of the clouds, and a BSSSDF, compute the outgoing radiance. Since we ignore the spatial distribution, this BSSSDF comes down to a BSDF. And since we only consider the top and bottom boundaries for solving radiance, this BSDF is actually only a BRDF R and a BTDF T.

We represent this problem as solving

$$L_{\uparrow}(+\mu, \phi) = L(z_1, +\mu, \phi), \tag{6.1}$$

$$L_{\downarrow}(-\mu, \phi) = L(0, -\mu, \phi) \tag{6.2}$$

with the boundary conditions defined through $L(z_1, -\theta, \phi)$ and $L(0, +\theta, \phi)$. Using the BRDF and BTDF approach, this can be solved through

$$\begin{aligned} L_{\uparrow}(+\mu,\phi) &= \int_0^{2\pi}\int_0^{\pi/2} R(z_1,\theta,\psi,\theta')L(z_1,\pi+\theta',\phi')\cos\theta'\sin\theta' d\theta' d\psi \\ &+ \int_0^{2\pi}\int_0^{-\pi/2} T(z_1,\theta,\psi,\theta')L(0,\pi+\theta',\phi')\cos\theta'\sin\theta' d\theta' d\psi, \qquad (6.3) \\ L_{\downarrow}(-\mu,\phi) &= \int_0^{2\pi}\int_0^{-\pi/2} R(z_1,\theta,\psi,\theta')L(0,\pi+\theta',\phi')\cos\theta'\sin\theta' d\theta' d\psi \\ &+ \int_0^{2\pi}\int_0^{\pi/2} T(z_1,\theta,\psi,\theta')L(z_1,\pi+\theta',\phi')\cos\theta'\sin\theta' d\theta' d\psi. \qquad (6.4) \end{aligned}$$

Thus, what we have to do is to find a model for R and T and compute Equations 6.3 and 6.4 efficiently.

To do so, we decompose L into three components corresponding to the three main sources lighting the cloud (sun, sky and ground), *i.e.,* we replace Equations 6.3 and 6.4 by

$$\begin{aligned} L_{\uparrow}(\theta,\phi) &= L_{\text{sun}\uparrow}(\theta,\phi) + L_{\text{sky}\uparrow}(\theta,\phi) + L_{\text{ground}\uparrow}(\theta,\phi), \qquad (6.5) \\ L_{\downarrow}(\theta,\phi) &= L_{\text{sun}\downarrow}(\theta,\phi) + L_{\text{sky}\downarrow}(\theta,\phi) + L_{\text{ground}\downarrow}(\theta,\phi). \qquad (6.6) \end{aligned}$$

We compute separately each component by taking advantage of our a priori knowledge about each of the sources. In the next subsections, we present each of these models.

6.3.2 Sun illumination

Regarding the sun, we assume it is illuminating the top of the clouds with a radiance L_{sun} from a single direction (θ', ϕ'). That is, Equations 6.3 and 6.4 for the sun come down to

$$\begin{aligned} L_{sun\uparrow}(+\mu,\phi) &= R_{\text{sun}}(z_1,\theta,\psi,\theta')L_{\text{sun}}\cos\theta', \qquad (6.7) \\ L_{sun\downarrow}(-\mu,\phi) &= T_{\text{sun}}(z_1,\theta,\psi,\theta')L_{\text{sun}}\cos\theta'. \qquad (6.8) \end{aligned}$$

Furthermore, we compute separately the contributions of different orders of scattering (see Figure 6.2). We write

$$\begin{aligned} R_{\text{sun}}(z_1,\theta,\psi,\theta') &= R_{\text{sun}}(0,z_1,\theta,\phi,\theta') \\ &+ R_{\text{sun}}(1,z_1,\theta,\phi,\theta') \\ &+ R_{\text{sun}}(2,z_1,\theta,\phi,\theta') \\ &+ R_{\text{sun}}([3,\infty],z_1,\theta,\phi,\theta') \qquad (6.9) \\ T_{\text{sun}}(z_1,\theta,\psi,\theta') &= T_{\text{sun}}(0,z_1,\theta,\phi,\theta') \\ &+ T_{\text{sun}}(1,z_1,\theta,\phi,\theta') \\ &+ T_{\text{sun}}(2,z_1,\theta,\phi,\theta') \\ &+ T_{\text{sun}}([3,\infty],z_1,\theta,\phi,\theta') \qquad (6.10) \end{aligned}$$

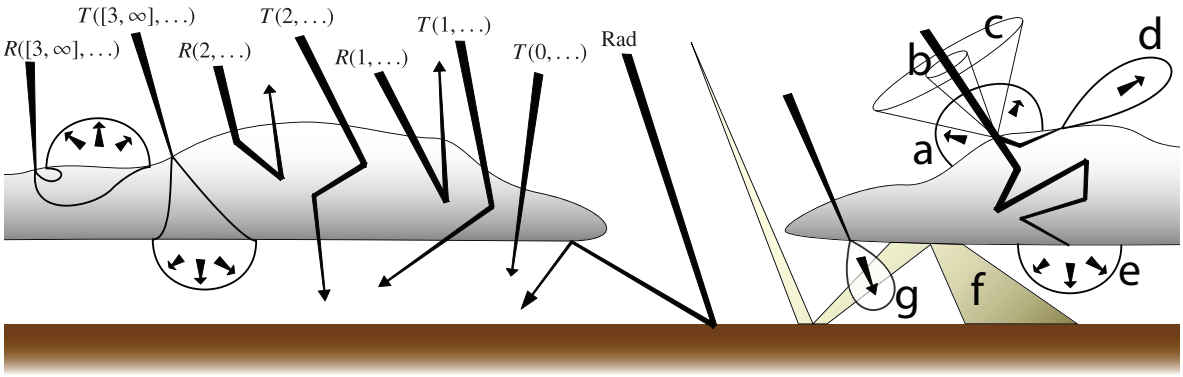


Figure 6.2: Left: *All the light path categories we simulate: orders* $n \geq 3$ *of scattering* ($R([3,\infty],\ldots)$, $T([3,\infty],\ldots)$)*, double scattering* ($R(2,\ldots)$, $T(2,\ldots)$)*, single scattering* ($R(1,\ldots)$, $T(1,\ldots)$)*, transparency* ($T(0,\ldots)$) *ground-clouds radiosity* (Rad). Right: *All the effects we reproduce:* a: *diffuse reflectance.* b: *glory.* c: *fogbow.* d: *pseudo-specular reflectance.* e: *diffuse transmittance.* f: *ground-clouds inter-reflection.* g: *forward scattering.*

where $R_{\text{sun}}(n, z_1, \theta, \psi, \theta')$ and $T_{\text{sun}}(n, z_1, \theta, \phi, \theta')$ model the contribution of light paths of order n. Since we know from our study that the reflectance behavior of each of these orders is different, we propose a separate BRDF and BTDF model for each of them.

6.3.2.1 Model for order 0

In this section we describe our model for $R_{\text{sun}}(0, \theta, \psi, \theta')$ and $T_{\text{sun}}(0, z_1, \theta, \psi, \theta')$.

Since we assume the sun to be always above the clouds, we have

$$R_{\text{sun}}(0, z_1, \theta, \psi, \theta') = 0. \tag{6.11}$$

Regarding T, as we have seen earlier transparency can be simply computed through optical thickness. However, since we use the chopped-peak model, the multiple strong forward scattering that should result in an anisotropic peak is transformed into transparency (see §4.4.2). We propose a way to recover from this approximation here.

We write

$$l = \frac{z_1}{\cos\theta} \tag{6.12}$$

the length of the paths of order 0. The optical depth with the reference model and with the chopped model are

$$\tau_{\text{ref}} = l\kappa_{\text{ref}}, \tag{6.13}$$

$$\tau_{\text{chopped}} = l\kappa_{\text{chopped}}, \tag{6.14}$$

respectively. We decompose $T_{\text{sun}}(0, \theta, \phi, \theta', \phi')$ into two components: the paths that did not encounter any droplet, *i.e.,* the "real" transparency as defined by the reference model ($n = 0_a$) and the paths that underwent multiple strong forward scattering ($n = 0_b$), *i.e.,*

$$T_{\text{sun}}(0, z_1, \theta, \psi, \theta') = T_{\text{sun}}(0_a, z_1, \theta, \psi, \theta') + T_{\text{sun}}(0_b, z_1, \theta, \psi, \theta'). \tag{6.15}$$

The transmittance due to the reference transparency is simply

$$T_{\text{sun}}(0_a, z_1, \theta, \psi, \theta') = \delta(\theta - \theta')\delta(\psi - \pi)e^{-\tau_{\text{ref}}}. \tag{6.16}$$

To compute the transmittance due to multiple strong forward scattering, we use the peak $P_{\text{peak}}(\Theta)$ of the reference phase function (Equation A.1 in Appendix A).

$$T_{\text{sun}}(0_b, z_1, \theta, \psi, \theta') = P_{\text{peak}}(\Theta)(e^{-\tau_{\text{chopped}}} - e^{-\tau_{\text{ref}}}). \tag{6.17}$$

Equation 6.17 represents a peak whose strength decreases with τ. It mimics the effect of multiple strong forward scattering that we described in our study (§4.4.2).

6.3.2.2 Model for order 1

Single scattering in an homogeneous slab admits an analytical solution, as we have seen in §2.5.5.4. We use the exact same formula. That is,

$$R_{\text{sun}}(1, z_1, \mu, \psi, \mu') = P(\Theta)\frac{1}{\mu' + \mu}\left[1 - \beta\left(\frac{z_1}{\mu} + \frac{z_1}{\mu'}\right)\right], \tag{6.18}$$

$$T_{\text{sun}}(1, z_1, \mu, \psi, \mu') = P(\Theta)\frac{1}{\mu' - \mu}\left[\beta\left(\frac{z_1}{\mu'}\right) - \beta\left(\frac{z_1}{\mu}\right)\right]. \tag{6.19}$$

6.3.2.3 Model for order 2

For double scattering, there is no analytical solution even in a slab. However, we know that in an infinite homogeneous medium, the angular distribution of radiance follows the multiply-scattered phase function, which is $P^{(2)}(\Theta)$, *i.e.,* $P(\Theta)$ convolved by itself. We propose to approximate the double scattering paths as having the same length as single scattering paths and to account for double scattering by using the multiply-scattered phase function. As a result, R and T are

$$R_{\text{sun}}(2, z_1, \mu, \psi, \mu') = P^{(2)}(\Theta)\frac{1}{\mu' + \mu}\left[1 - \beta\left(\frac{z_1}{\mu} + \frac{z_1}{\mu'}\right)\right], \tag{6.20}$$

$$T_{\text{sun}}(2, z_1, \mu, \psi, \mu') = P^{(2)}(\Theta)\frac{1}{\mu' - \mu}\left[\beta\left(\frac{z_1}{\mu'}\right) - \beta\left(\frac{z_1}{\mu}\right)\right]. \tag{6.21}$$

6.3.2.4 Model for orders $[3, \infty]$

According to our study, the angular distribution or light paths is less anisotropic as n increases. We propose to consider the reflectance and transmittance of paths of order $n = [3, \infty]$ as diffuse. That is,

$$R_{\text{sun}}([3, \infty], z_1, \mu, \psi, \mu') = \frac{1}{2\pi}M_R([3, \infty], z_1, \mu'), \tag{6.22}$$

$$T_{\text{sun}}([3, \infty], z_1, \mu, \psi, \mu') = \frac{1}{2\pi}M_T([3, \infty], z_1, \mu'), \tag{6.23}$$

where $M_{R\text{sun}}(n, z_1, \mu')$ and $M_{T\text{sun}}(n, z_1, \mu')$ are the reflected and transmitted exitances due to paths of order n, respectively. Using this approximation, we directly use the exitance model proposed in §5.2 to compute $M_{R\text{sun}}(n, z_1, \mu')$ and $M_{T\text{sun}}(n, z_1, \mu')$.

6.3.3 Sky and ground illumination

Since sky and ground are very wide sources, we represent them with a diffuse angular distribution. As a result, we can use our order-independent exitance model to compute reflection and transmission of sky and ground illumination, *i.e.,*

$$L_{\text{sky}\uparrow}(+\mu,\phi) = \frac{1}{2\pi} M_R([1,\infty], z_1, \mu') E_{\text{sky}}, \tag{6.24}$$

$$L_{\text{sky}\downarrow}(-\mu,\phi) = \frac{1}{2\pi} M_T([1,\infty], z_1, \mu') E_{\text{sky}}, \tag{6.25}$$

where E_{sky} is the illuminance due to the sky and M_R and M_T are the reflected and transmitted reflectance models described in §5.2.1.

Similarly, the luminance due to ground illumination is

$$L_{\text{ground}\uparrow}(+\mu,\phi) = \frac{1}{2\pi} M_T([1,\infty], z_1, \mu') E_{\text{ground}}, \tag{6.26}$$

$$L_{\text{ground}\downarrow}(-\mu,\phi) = \frac{1}{2\pi} M_R([1,\infty], z_1, \mu') E_{\text{ground}}, \tag{6.27}$$

where E_{ground} is the illuminance due to the ground.

The sky luminance is computed using a standard realtime algorithm [O'N05]. The ground luminance is computed by reflecting the illuminance it receives from sky, clouds and sun, as we explain in the next section.

6.3.4 Ground-clouds inter-reflection model

As we have shown in §2.2.2.1, there are inter-reflections between the ground and the clouds that are significant if the albedo of the ground is high. Thus, the illuminance E_{ground} of the ground on the clouds is affected by the radiance of the bottom of the clouds $L_\downarrow$. We propose to compute these inter-reflections using the radiosity method by considering that the ground and the clouds are two parallel planes and using the radiosity method as follows. Note that since radiosity is a very common technique, this section addresses our model somewhat quickly.

6.3.4.1 Ground radiosity

The radiosity method is an approach to compute radiance transfer between diffuse surfaces [FvDFH90]. If a surface is diffusive, then the radiance it emits is directly proportional to its exitance (actually, the term *radiosity* is a synonym for *radiant exitance* – compare with *luminosity*). Thus, assuming the ground is diffusive, the luminance leaving the ground at a given point $\boldsymbol{p}$ in direction $\vec{s}$ is

$$L(\boldsymbol{p}, \vec{s}) = \frac{1}{2\pi} M(\boldsymbol{p}), \tag{6.28}$$

where $M(\boldsymbol{p})$ is the exitance of the ground at $\boldsymbol{p}$.

In the following, we call $\boldsymbol{p}'$ the point on the clouds located between a given point $\boldsymbol{p}$ on the ground and the sun, *i.e.,* it is the intersection point between the clouds and the ray of direction $\vec{s}'$ starting from $\boldsymbol{p}$. This exitance can be computed by reflecting the irradiance on the ground. This irradiance

is classically decomposed into two components: a diffuse component $E_d(\boldsymbol{p})$ (light coming from diffuse surfaces) and a directional component $E_s(\boldsymbol{p})$ (light coming from directional light sources). Given these two values, M can be computed by

$$M(\boldsymbol{p}) = \varpi(\boldsymbol{p})(E_d(\boldsymbol{p}) + E_s(\boldsymbol{p})), \tag{6.29}$$

where $\varpi(\boldsymbol{p})$ is the albedo of the ground at $\boldsymbol{p}$.

We consider that the light coming from the sun brings directional irradiance. This translates to

$$E_s(\boldsymbol{p}) = E_{\text{sun}}\beta(\boldsymbol{p}'), \tag{6.30}$$

where $\beta(\boldsymbol{p}')$ is the extinction function of the clouds at $\boldsymbol{p}'$.

The diffuse irradiance $E_d(\boldsymbol{p})$ on the ground is due to the sky and the radiance of clouds due to high order paths (orders $n \geq 3$). This translates to

$$E_d(\boldsymbol{p}) = \iint dE_d(\boldsymbol{p}, \theta', \phi')F(\theta')d\theta' d\phi', \tag{6.31}$$

$$dE_d(\boldsymbol{p}, \theta', \phi') = E_{\text{sun}}M_{T\text{clouds}}([3, \infty], \boldsymbol{p}') + E_{\text{sky}}\beta(\boldsymbol{p}'), \tag{6.32}$$

where $M_{T\text{clouds}}([3, \infty], \boldsymbol{p}')$ is the transmitted exitance of clouds, computed as described in §6.3.2.4. $F(\theta')$ is called the *form factor* between two elements of surfaces. It defines how the radiance of one element of surface illuminates the other and it is one of the root elements of the radiosity method.

form factor

In our case (two parallel planes), F depends only on θ' thus F is constant over a ring of directions. In the spirit of [SP89], we consider 3 areas of similar contribution on which we assume F is constant. We improve their method by choosing 3 square rings S_1, S_2, S_3 of exterior size $h, 2h, 8h$ centered on $\boldsymbol{p}$. The form factors for these areas are $F(\boldsymbol{p}, S_1) = 0.5, F(\boldsymbol{p}, S_2) = 0.3, F(\boldsymbol{p}, S_3) = 0.18$. In fact we use $F_1 = 0.5, F_2 = 0.3, F_3 = 0.2$ to avoid loosing energy. Equation 6.31 transforms to

$$E_d(\boldsymbol{p}) = F_1 \int_{S_1} dE_d(\boldsymbol{p}, \theta', \phi')d\theta' d\phi' + F_2 \int_{S_2} dE_d(\boldsymbol{p}, \theta', \phi')d\theta' d\phi' + F_3 \int_{S_3} dE_d(\boldsymbol{p}, \theta', \phi')d\theta' d\phi'. \tag{6.33}$$

Graphics hardware allows one to quickly compute an integral over a square through MIP-mapping. We take advantage of this by remembering that a square ring is the difference between two squares. Let A_1, A_2, A_3 be squares of size $h, 2h, 8h$. We have $S_1 = A_1, S_2 = A_2 - A_1, S_3 = A_3 - A_2$. The integral $dE_i = \int_{A_i} dE_d(\boldsymbol{p}, \theta', \phi')d\theta' d\phi'$ is obtained through graphics hardware. As a result, Equation 6.33 simplifies to

$$\begin{aligned} E_d(\boldsymbol{p}) &= F_1 dE_1 + F_2(\frac{4}{3}dE_2 - \frac{1}{3}dE_1) + F_3(\frac{64}{60}dE_3 - \frac{4}{60}dE_2) \\ &= \nu_1 dE_1 + \nu_2 dE_2 + \nu_3 dE_3, \end{aligned} \tag{6.34}$$

with $\nu_1 = 0.4, \nu_2 = 0.387, \nu_3 = .213$.

Thanks to all this, we are now able to compute the radiance of the ground in all points. This radiance illuminates the clouds. The irradiance on the base of the clouds due to the ground is computed in the exact same manner as the irradiance on the ground described above.

As a result, we are provided with the radiosities of both the ground plane and the clouds plane. We then use the classical iterative radiosity algorithm [FvDFH90] to compute the inter-reflections between these planes. We use 2 to 8 iterations depending on the albedo of the ground (we used 8 in Figure 6.14(b)).

Summary of §6.3 : Thanks to the starting hypothesis of locally considering the cloud as a slab, we found models to compute all the orders of multiple anisotropic scattering in clouds. We use the chopped-peak model for which we correct the error. We use an analytical solution for single scattering. We use an approximation of double scattering through phase function convolution. The corrected doubling-adding model of §5.2 provides us with higher orders of multiple scattering. We also propose an optimized GPU-friendly solution of the radiosity problem between two parallel surfaces.

Grâce à l'hypothèse initiale de considérer localement un nuage comme une dalle, nous avons trouvé des modèles permettant de calculer tous les ordres de la diffusion anisotrope multiple dans les nuages. Nous avons utilisé le modèle chopped-peak *de la fonction de phase dont nous corrigeons l'erreur. Nous utilisons une solution analytique pour la diffusion primaire. Pour la diffusion secondaire, nous utilisons une approximation via une convolution de la fonction de phase. Le modèle corrigé de* doubling-adding *(§5.2) nous fournit les ordres supérieurs de la diffusion multiple. Nous présentons également une solution optimisée pour le GPU pour le calcul de la radiosité entre deux surfaces parallèles.*

6.4 Implementation of our model on GPU

We implemented all the formulas of §6.3 in shaders on the GPU. The heightfield $z_1(\boldsymbol{p})$ is stored into a texture, as well as the albedo of the ground and the parameters of our multiple scattering model (phase function, parameters for the corrected doubling-adding model, etc.). For the cloud field, we rely on 512×512 animated textures precomputed using Perlin noise [Per85] on the CPU and animated using advected textures [Ney03]. We display the cloud surface as a geometric patch that is built on the fly on the CPU from the heightfield. The clouds height vary from 0 m to 1000 m and they span several km.

At rendering time, at each frame, we first compute the inter-reflections between the ground and the clouds in a shader (§6.3.4). Then we compute the final luminance at each visible pixel of the clouds (§6.3).

The implementation is done on a PC with an NVIDIA Quadro FX 1400. At resolution 1024×768, an image in which 50% of the pixels are covered by clouds runs at 18 FPS. One major bottleneck during clouds animation is the update of the 512×512 height field geometry. When clouds are not animated, the frame rate increases to 40 FPS.

6.5 Results

Figure 6.3 shows a comparison between our BRDF model (§6.3.2) and the results of our simulations. We can see that our approximation fits the simulations quite well.

Our results are illustrated in Figures 6.4 – 6.14. They show the various expected features including pseudo-specular reflection (Figure 6.5), silver lining (Figures 6.4, 6.8, 6.10), fogbow and glory (Figure 6.13), and the ground-clouds inter-reflections including water sky and ice blink(Figures 6.12 and 6.14). Figures 6.7 and 6.9 show the contribution of different sources and orders for Figures 6.6 and 6.8, respectively.

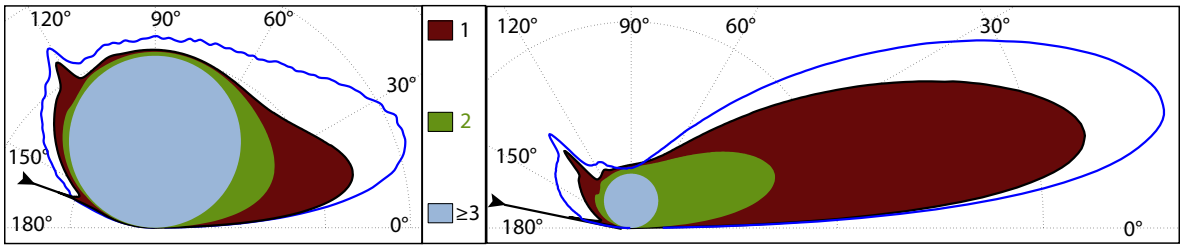


Figure 6.3: *Comparison of our BRDF model against the result of our simulations.* Left: $z_1 = 100\,\mathrm{m}$, $\theta' = 70°$. Right: $z_1 = 5\,\mathrm{m}$, $\theta' = 80°$. Areas: *Contribution of each order of scattering.* Blue line: *Monte-Carlo result.*

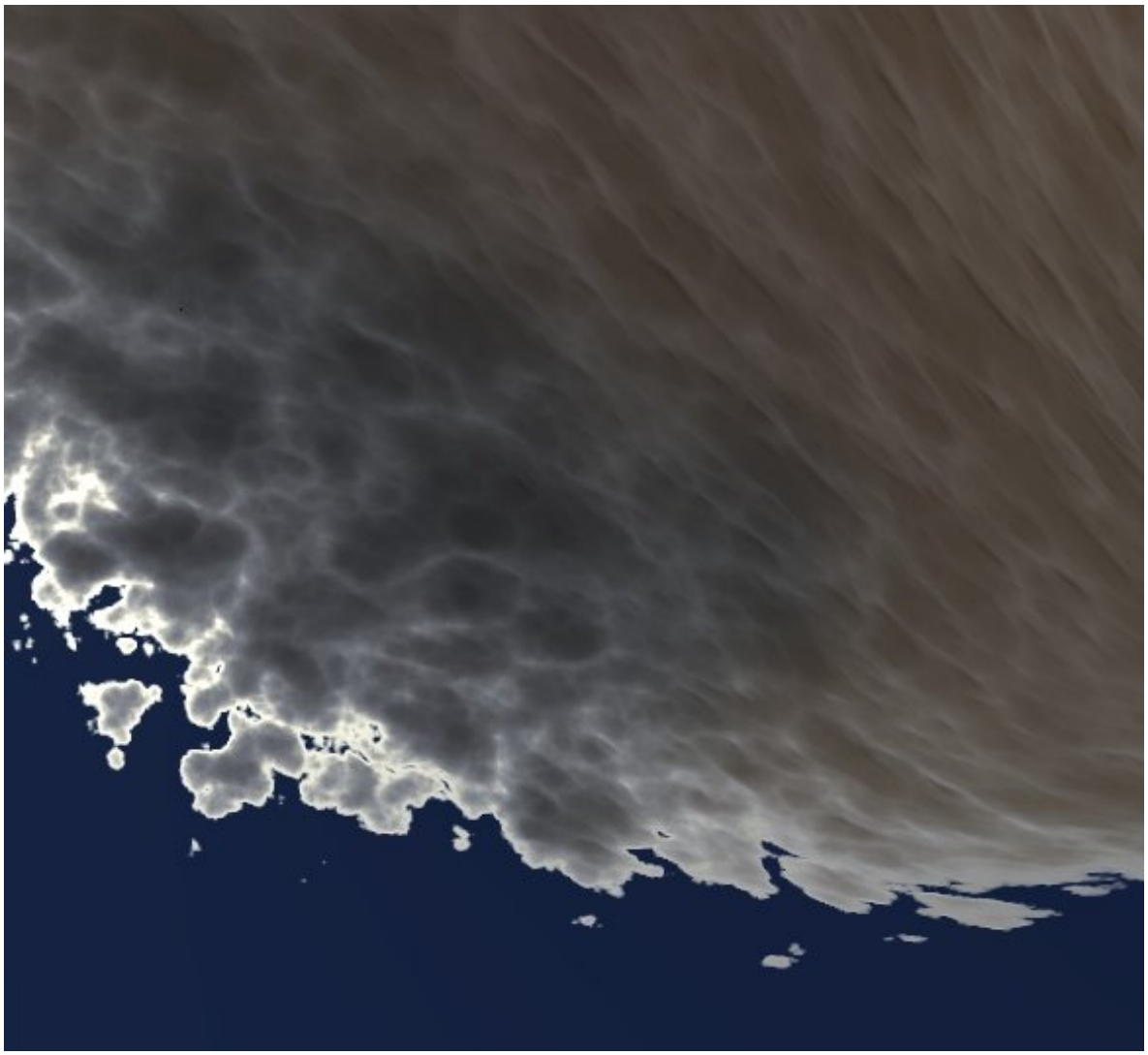
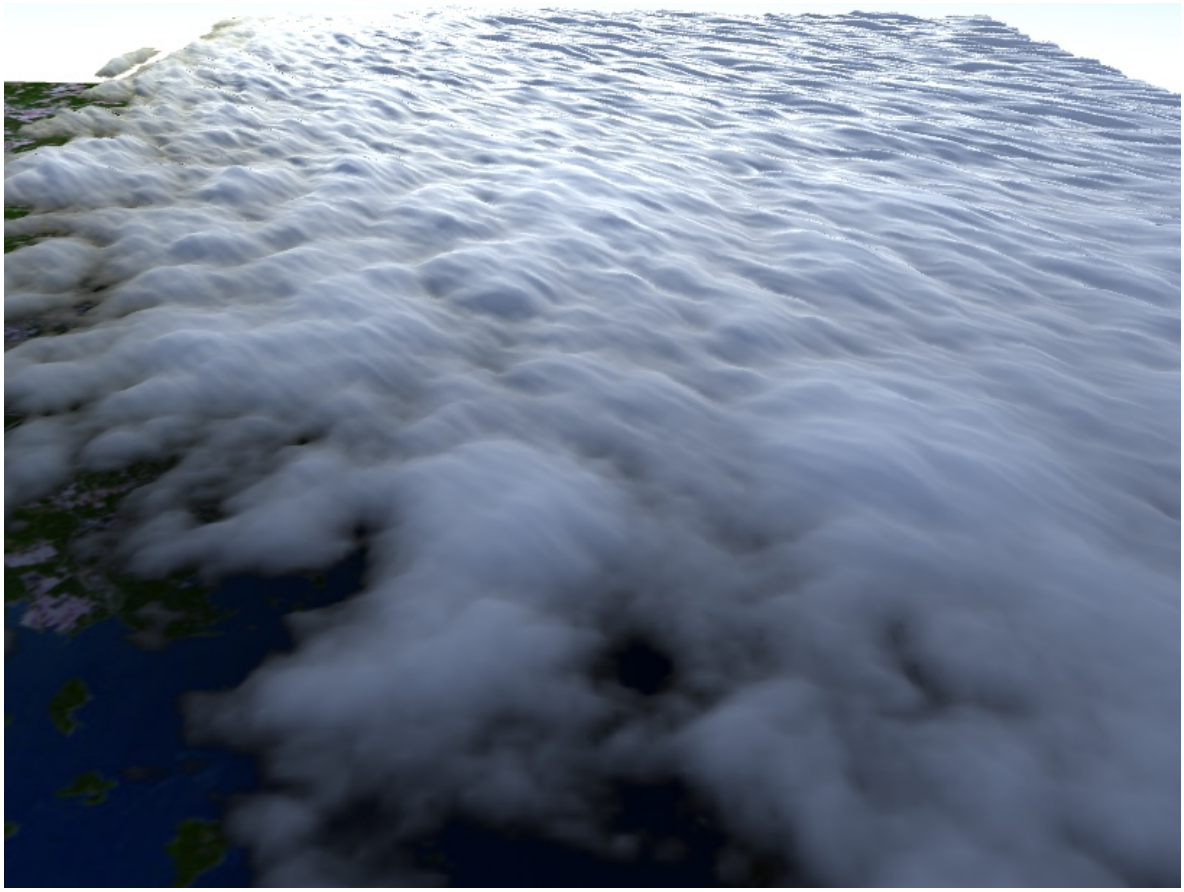


Figure 6.4: *Clouds above the desert. Note the strong forward scattering (edges are brighter in the sun direction), the diffusive thick parts, the blue tints (sky contribution) in the bottom left part of the image, the orange tints (ground contribution) in the top right part of the image.*

(a) Clouds in full daylight. Note the high backscattering.



(b) Clouds at sunset. Note the pseudo-specular effect.

Figure 6.5: *Views from above the clouds with two different sun angles.*

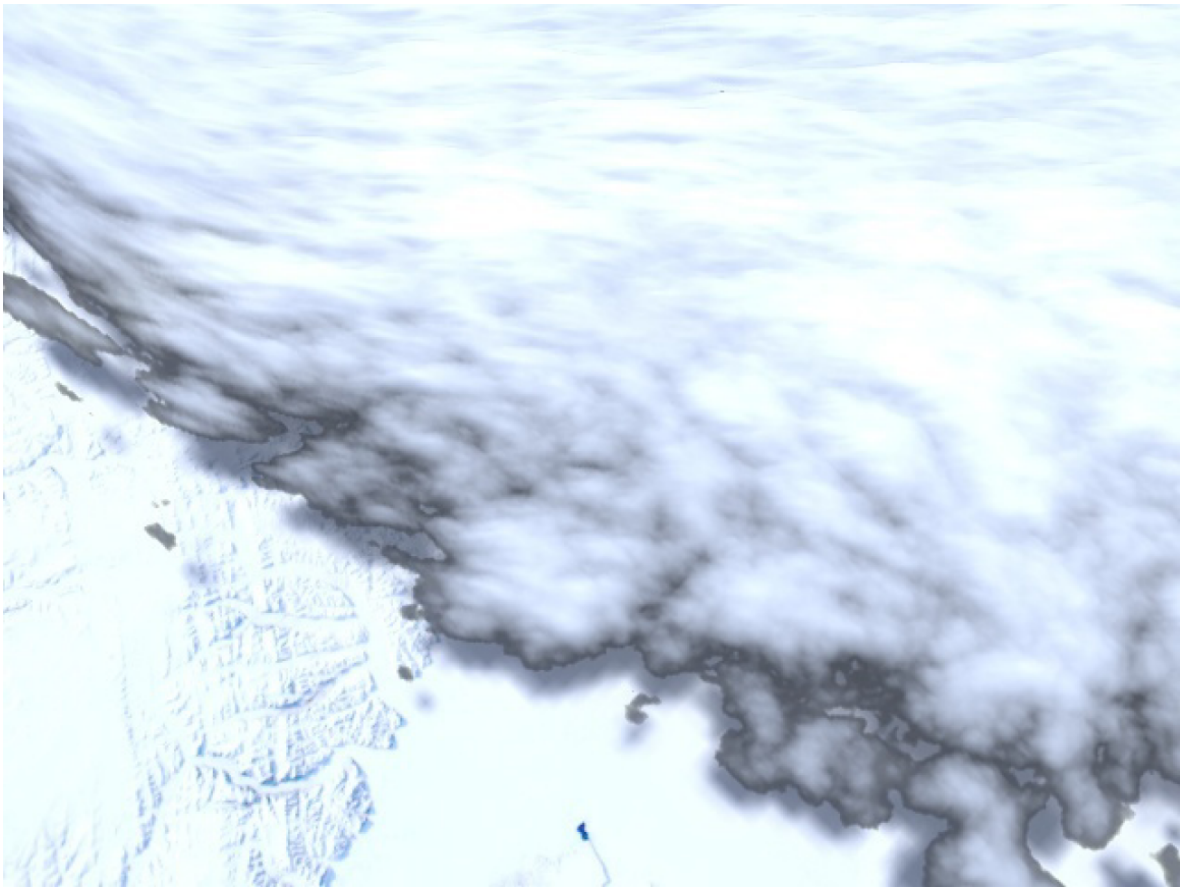
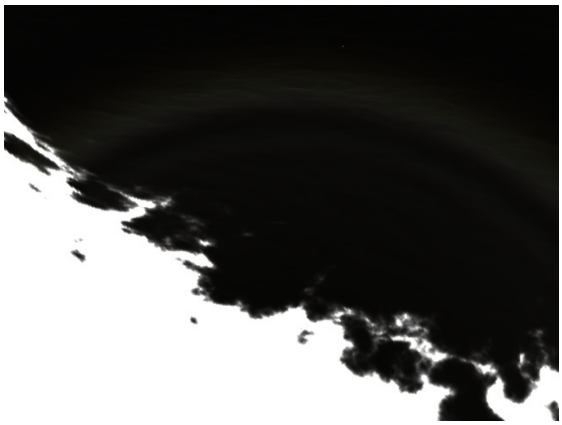
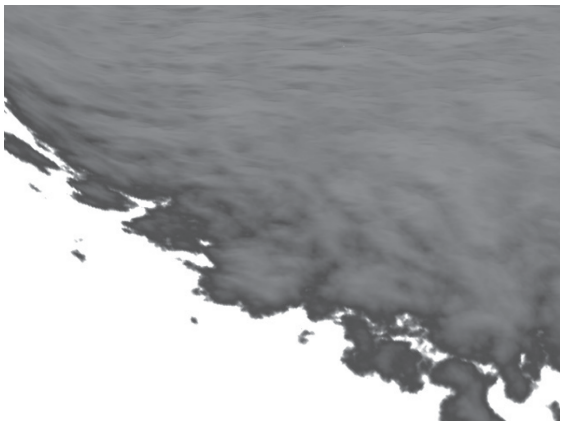


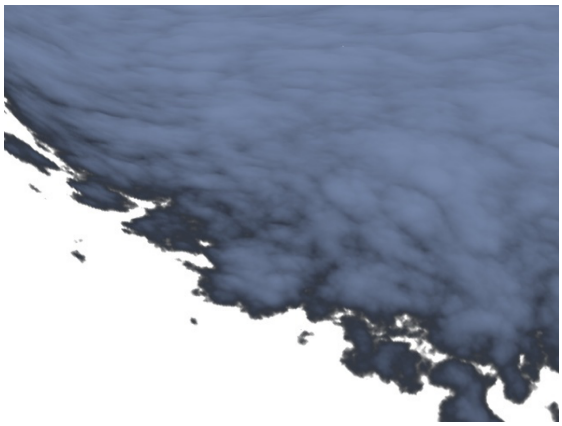
Figure 6.6: *Clouds viewed from above over the ice. The different components contributing to this image are shown on Figure 6.7.*



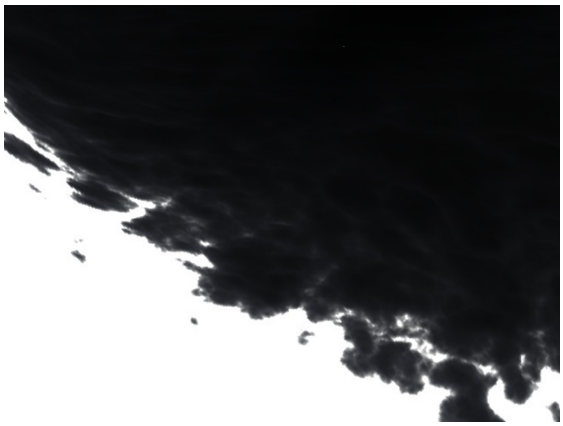
(a) Contribution of the sun due to orders $n = [1, 2]$. Note the high anisotropy bringing and the silver lining.



(b) Contribution of the sun due to orders $n = [3, \infty]$. Note the diffusive transmittance.



(c) Contribution of the sky.

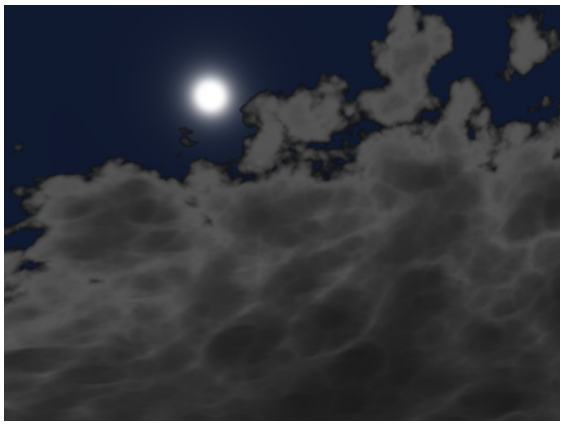


(d) Contribution of the ground.

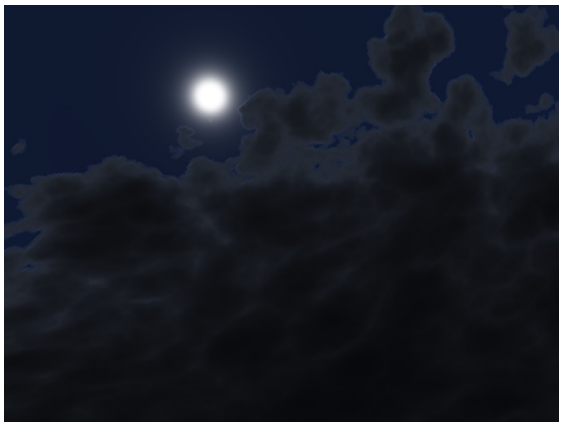
Figure 6.7: *The different components contributing to the image of Figure 6.6.*

Figure 6.8: *Clouds viewed from below. The different components contributing to this image are shown on Figure 6.9.*

(a) Contribution of the sun due to orders $n = [1, 2]$. Note the very low reflectance and the fogbow.



(b) Contribution of the sun due to orders $n = [3, \infty]$. Note the diffuse reflectance and the dark edges.



(c) Contribution of the sky. Note importance of this contribution.

(d) Contribution of the ground.

Figure 6.9: *The different components contributing to the image of Figure 6.8.*

Figure 6.10: *Close-up on the silver lining.*

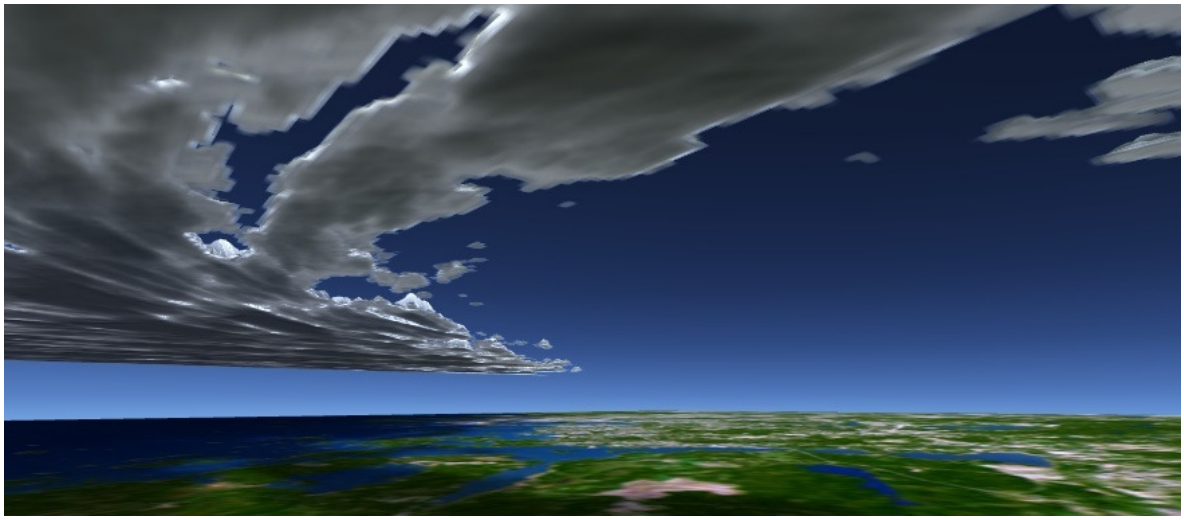
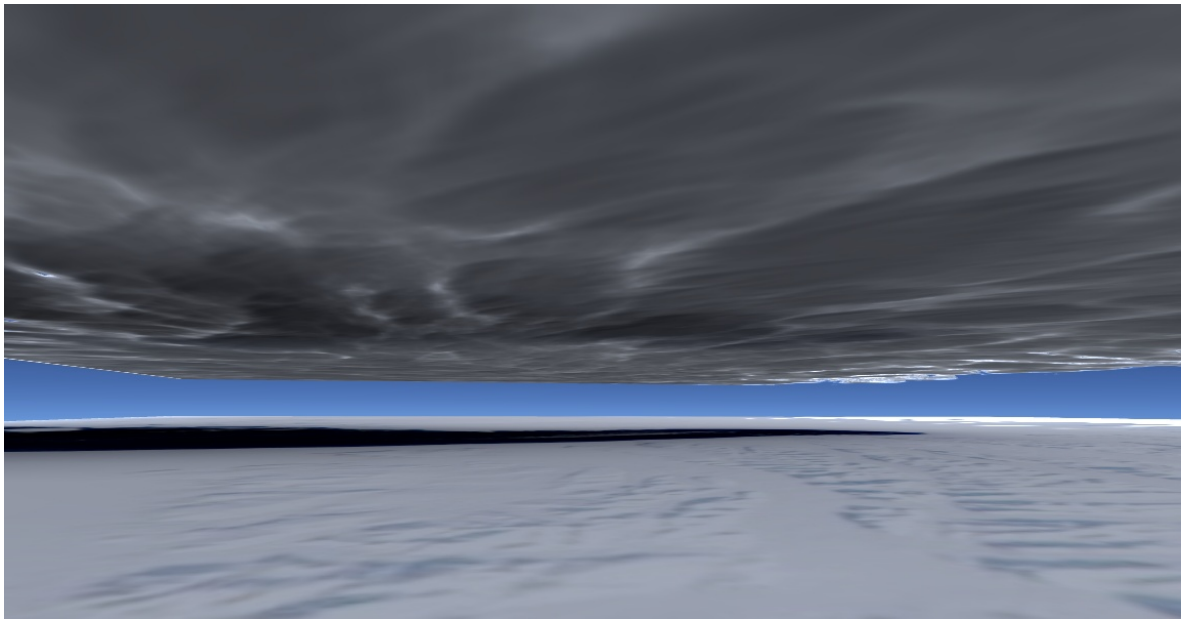
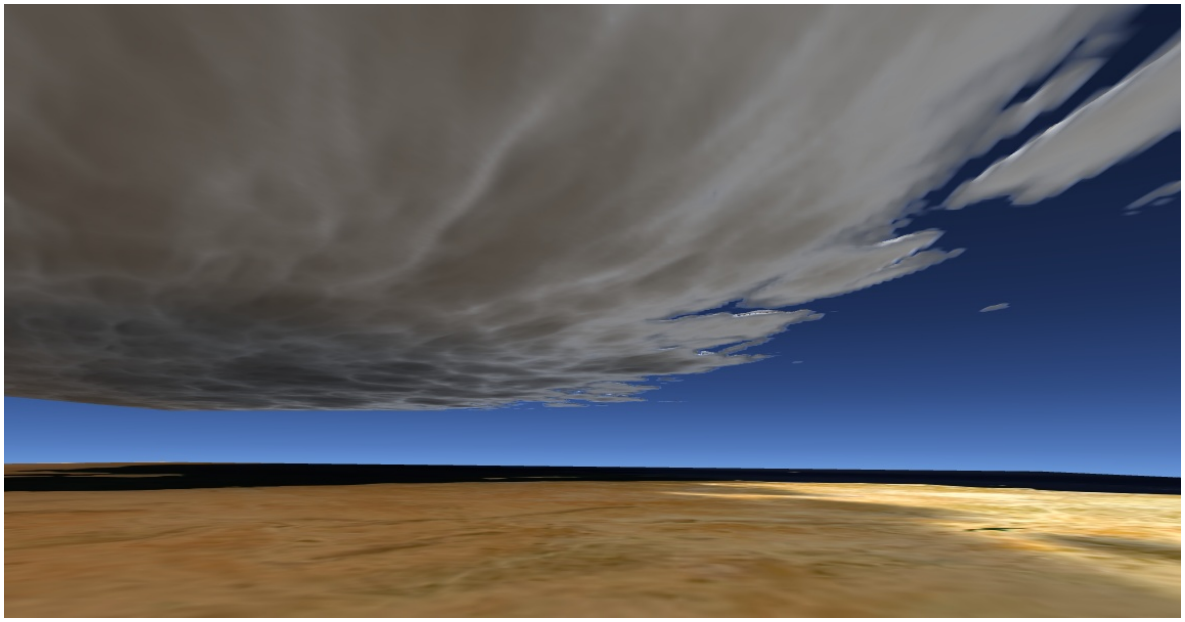


Figure 6.11: *View from under the clouds in daytime.*



(a)



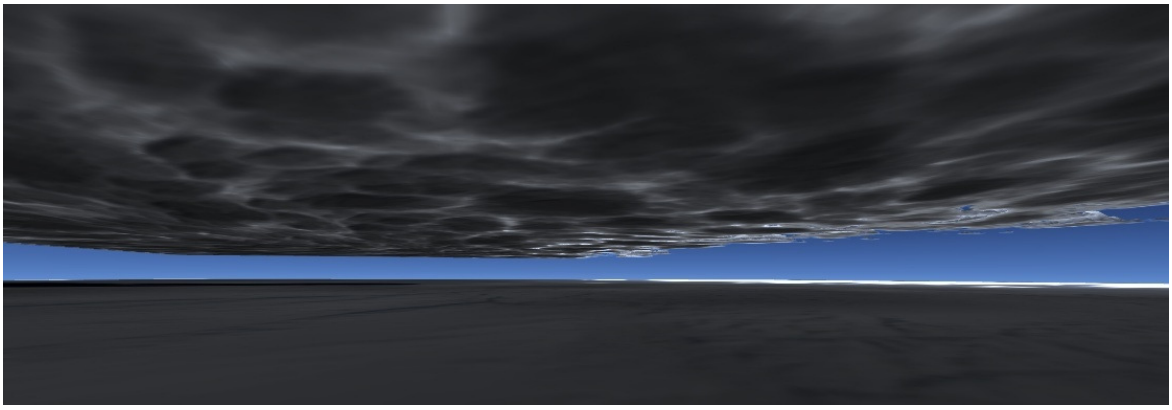
(b)

Figure 6.12: *Effect of the ground-clouds inter-reflections. Note how the clouds are darker above the sea.*

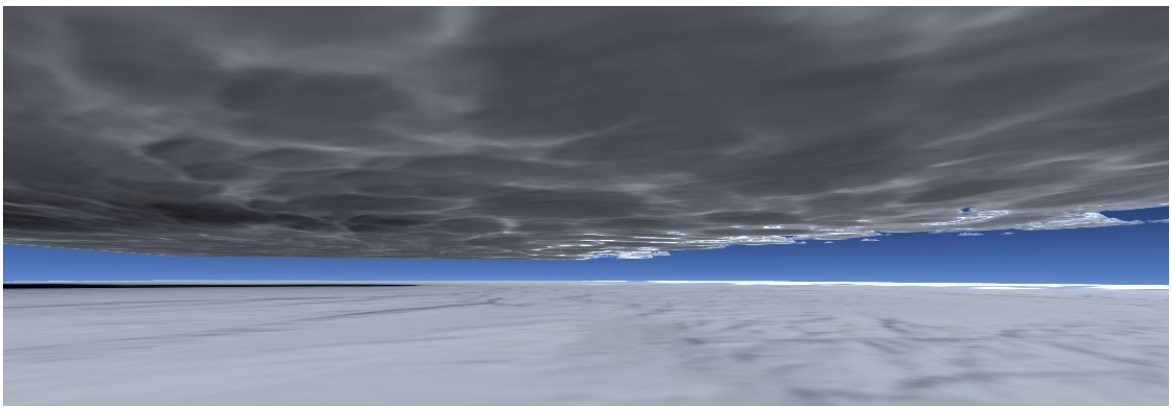
(a) Fogbow.

(b) Glory.

Figure 6.13:



(a) Without inter-reflections.



(b) With inter-reflections.

Figure 6.14: *Cloudy sky over the ice with and without accounting for inter-reflections between the clouds and the ground.*

6.6 Discussion

We have presented a method capable of rendering realistic stratiform clouds in realtime. Relying on geometry for clouds is very efficient as compared to the massive overdraw caused by methods with sliced volumes. As shown on Figure 6.3, the BRDF and BTDF models we use are approximating the reference models well. The errors in our model are mainly due to our approximation of double scattering and our assumption that light paths of order $n \geq 3$ are diffusive (whereas it is the case only for $n \geq 16$, as we have seen in our study). Still, this method is capable of reproducing all the visual features of clouds we described in §2.2 (which previous methods were not capable of), including ground-clouds inter-reflections, at speeds at least as fast as previous works.

The main limitation of this method is that it is limited to clouds having stratiform shapes. Moreover, it does not handle points of view that are on the side of clouds or inside them. Finally, ignoring spatial spreading, along with considering their shape as totally homogeneous with a sharp boundary, gives them a too solid appearance. We do not treat self-shadowing (*i.e.,* the shadow of clouds on themselves). Also, the interaction with a non-flat ground has yet to be treated. We did not consider aerial perspective.

Nevertheless, it has a few potential applications. In numerous video games (*e.g.,* first person shooters, role-playing games, racing games, etc.) the point of view is located on the ground in an outdoor environment. Although the sky is not of primary importance in their graphics, these games do need realistic skies. The traditional approach of using a texture ("*skybox*" or "*skydome*"), which prevents any animation, is nowadays abandoned because it limits the possibilities of immersion of the player.

Using this method would allow for the complete animation of the sky environment, including realistic clouds. Its computing cost is sufficiently low to be used in a game where resources are also needed for other tasks (*e.g.,* gameplay, AI, networking, sound, etc.), especially with the increase of performance of today's GPUs. This method is currently being implemented in *HeroEngine*, the graphics engine of a commercial MMORPG, "*Hero's Journey*", by Simutronics.

6.7 Summary of Chapter 6

We have proposed a simple and efficient model for the realtime rendering of stratiform clouds. This model has been published in [BNL06]. It relies on the representation of the shape of the clouds through a height field, which is particularly efficient (§6.2). Our rendering model accounts for the anisotropic light transport of clouds by treating separately the first two orders of scattering (§6.3.2.1 – §6.3.2.2). It accounts for the diffusive light transport by modeling paths of higher orders through a doubling-adding scheme (§6.3.2.4). This scheme is adapted to the specifics of clouds by adding a correction term inferred from our study of light transport in slabs (§5.2). Moreover, we account for the sky and ground illumination (§6.3.3) as well as the inter-reflections between the ground and the clouds (§6.3.4). We propose a GPU implementation of this model yielding realtime speeds (§6.4).

This model is able to reproduce most of the visual features of clouds: detailed shape, glory, fogbow, view-dependant lighting effects (silver lining, dark edges, pseudo-specular effect), high contrast, sky and ground illumination.

The main limitation of this model is that it is limited to stratiform clouds. In the next chapter, we develop a model that is more complex but suited to any type of clouds, and in particular to cumulus-type clouds.

Note that the method presented in this chapter is actually very close to the approach called "independent pixel approximation" by atmospheric physicists, which is used in the remote sensing of clouds [CRW+94, MOD+99]. It has also been shown that its validity is somewhat limited [CWE97]. Indeed, by only taking into account vertical radiative transfer, one ignores the horizontal spreading of light. In the case of vertical-development clouds (*i.e.,* cumulus-type clouds), this error is significant, since a lot of light escapes from the sides of the clouds. As a result, efficient 3D radiative transfer is necessary for accurate remote sensing of real clouds, and it is currently still a difficult problem in atmospheric physics. The method that we present in the next chapter may give insights to atmospheric physicists on how to achieve efficient 3D radiative transfer.

Nous avons proposé un modèle simple et efficace pour le rendu de nuages stratiformes en temps réel. Ce modèle a été publié dans [BNL06]. Il repose sur la représentation de la forme des nuages par un champ de hauteur, ce qui est particulièrement efficace (§6.2). Notre modèle de rendu tient compte du transport anisotrope de la lumière par les nuages en traitant séparément les deux premiers ordres de diffusion (§6.3.2.1 – §6.3.2.2). Il tient compte du caractère dispersif du transport de la lumière en modélisant les chemins d'ordres supérieurs via une méthode doubling-adding *(§6.3.2.4). Cette méthode est adaptée aux particularités des nuages en ajoutant un terme de correction inféré à partir de notre étude du transport de la lumière dans les dalles (§5.2). De plus, nous tenons compte de l'illumination des nuages par le ciel et le sol (§6.3.3), ainsi que des inter-réflexions entre le sol et les nuages (§6.3.4). Nous présentons une implantation de ce modèle sur GPU aboutissant à des vitesses temps réel (§6.4).*

Ce modèle est capable de reproduire la plupart des éléments visuels des nuages: une forme détaillée, la gloire, l'arc-en-nuage, les effets dépendant du point de vue (silhouette lumineuse ou sombre, effet pseudo-spéculaire), un contraste fort, l'illumination par le ciel et le sol.

La limite principale de ce modèle est qu'il est restreint aux nuages stratiformes. Dans le chapitre suivant, nous présentons un modèle plus complexe mais adapté à tout type de nuage, et en particulier aux nuages cumuliformes.

*Il est à noter que la méthode présentée dans ce chapitre est en réalité très proche de l'approche appelée "*independent pixel approximation*" par les physiciens atmosphériques, qui est notamment utilisée dans la reconstruction de nuages à partir d'images satellite [CRW+94, MOD+99]. Il a été montré que la validité de cette approche est limitée [CWE97]. En effet, lorsque seulement le transfert radiatif vertical est pris en compte, l'étalement horizontal de la lumière est ignoré. Dans le cas des nuages à développement vertical (i.e., du type cumulus), cette erreur est significative, puisque beaucoup de lumière s'échappe par les bords des nuages. En conséquence, une méthode efficace de calcul du transfert radiatif en 3D est nécessaire pour la reconstruction précise de nuages réels, et c'est actuellement encore un problème difficile en physique atmosphérique. La méthode que nous présentons dans le chapitre suivant pourrait donner des idées aux physiciens atmosphériques quand à comment parvenir à un tel calcul efficace de transfert radiatif 3D.*

Demonstration

Fair-weather clouds

Gouache is a versatile paint that can be used diluted with water in the same way as watercolor, or thicker like oils. But it has a character very much its own; the paint has a smooth, matt, velvety texture that produces a richness of color quite unlike any other medium. David Carr uses the medium freely, producing atmospheric paintings which express his feelings for a landscape. He feels his way into the painting, starting with a brush drawing combined with washes of color. Once the composition is mapped out, he starts building up the paint, working from dark to light. Like any landscape artist, David Carr tackles common problems on the way: integrating the clouds into the sky; creating depth in the painting; and unifying the sky area with the landscape.

Colors
Ultramarine blue
Cerulean blue
Ultramarine violet
Indigo
Cobalt green
Viridian
Lemon yellow
Cadmium yellow (medium)
Yellow ocher
Cadmium red
Burnt umber
Titanium white

Brushes
White-bristle filbert, Nos. 6 and 8; Sable or sable mix, Nos. 3 and 12.

Support
This artist usually paints with gouache on cardboard, but gouache can be used on almost any ground that is free from grease or oil – paper, boards, masonite, or stretched linen.

Easel
This artist uses gouache of a near-dry consistency, and paints standing at an easel. However, many painters using gouache prefer to dilute the paint to a more liquid consistency and work on a flat surface.

1 With a white-bristle, No. 8 filbert brush, the artist intuitively explores the landscape. He finds this a particularly versatile brush, using it flat for laying washes and on its edge for drawing with the paint.

1

2 Carr considers this the end of the first stage. He has established the main areas of the picture, both the clouds and the landscape. Already conscious of unifying one with the other, he has the clouds echoing the rocks and the bay.

2

3

3 Now Carr starts to block in areas with slightly thicker paint. Colored highlights – white touched with violet and yellow ocher – are added freely. The strident violet mix will form the background of a rich build-up of delicate colors.

4

4 The bold brushwork is already giving life to this scene, but will be tempered later by overpainting.

5 Carr wants to bring these clouds forward as if they were moving toward him. The yellow ocher in the highlights will help to do this, but here he is creating bolder shadows.

5

San Francisco Bay, *David Carr, gouache.*
Having checked that "the bay and the sky all unify in some sort of sense of light and space," David Carr decides the painting is complete. Knowing when to stop painting is not always easy. Try checking your work in a mirror for an "objective" view.

8 Carr feels the scumbling is too extreme. Loading a No. 12 softhair brush with a lot of clean water, he blends and softens the strokes.

8

7 Dry gray paint is scumbled into the clouds to give volume and definition.

7

6 The clouds are certainly beginning to come together. Working the sky into the clouds helps to integrate them and create a natural, atmospheric effect. Taking the sky color down to the water in the bay also helps to unify sky and landscape.

6

9 Now, returning to the bristle brush, the artist is bringing things together. Here he is touching up a highlight, but he also softens contours in the background so that they recede.

9

Chapter 7

A rendering method for arbitrary clouds

Contents

7.1 Introduction

The model we have presented in Chapter 6 shows that it is possible to compute a realtime, realistic approximation of light transport in clouds. However, this model is limited in several ways. It is restricted to stratiform clouds. It does not provide accurate renderings of the clouds when viewed from their sides. It does not allow the camera to enter the clouds. It does not account for the spatial spreading of light (§2.5.3.3) which, together with the shape model and homogeneous assumption used, give the clouds a very "solid" look.

This chapter presents a second model which makes full use of the results of our study and overcomes all the issues of the previous model. It is designed for any cloud shape, and in particular the dense, fluffy cumulus that are the Achilles's heel of previous methods. It accounts for both spatial and angular spreading of light, as well as all other light transport behaviors supported by our previous method: high-order diffusive backscattering, low-order detailed backscattering, multiple strong forward scattering, diffusive high-order forward scattering, anisotropic scattering. It allows the camera to view the clouds from any point of view, including from inside the cloud. It reproduces almost all the visual features of clouds which are view-dependent cloud edges, bright and detailed lit side, dark and soft unlit side, glory, fogbow and pseudo-specular reflection. It relies on a shape model that accounts for inhomogeneities at the boundaries of the cloud, and it allows animation.

This model is more computationally expensive than our previous one and our implementation shows only interactive frame rates (2–10 FPS) on current hardware. However, as we will see, its cost should be easily scalable if realtime speed is needed. Like the rest of our work, our model uses the subsurface scattering paradigm, *i.e.,* we compute light transport between the lit boundary of the clouds and the visible boundary. It relies heavily on the results of our study in Chapter 4 by carefully and accurately approximating light transport in an arbitrary cloud from light transport in a slab. This model was published in [BNM+08].

This model is composed of various parts. Its main contributions are

- A new way of considering multiple scattering (*i.e.,* orders $n > 1$ of scattering) in clouds by considering the *collecting area* of most important light paths on the *lit surface* of the cloud (§7.4.3.1).
- A new algorithm for quickly finding this collecting area and solving light transport on a cloud of arbitrary shape using the results of our study on stratiform clouds (§7.4.3.5).
- A realtime implementation of this algorithm on GPU (§7.6).
- A new algorithm for the fast computation of single scattering in clouds (§7.4.2), along with its implementation on GPU (§7.6).
- A new model for representing the shape of a cloud, based on Hypertextures [PH89], which allows us to represent finely the visible inhomogeneities of the cloud density.

We begin by describing our shape representation in §7.2. Our rendering method is presented in §7.3 – §7.5. Its implementation is outlined in §7.6. Results are shown in §7.7 and we discuss our model in §7.8.

7.2 Cloud shape and optical parameters

We begin by presenting the shape representation on which we perform our rendering method. This rendering method is presented in the next sections.

7.2.1 General shape representation

Our previous model assumed that the cloud was totally homogeneous inside boundaries delimited by a heightfield, which gives it a "solid" look[1]. This may be sufficient to model sharp edges, but it does not reproduce the puffy features and high frequencies of cumulus clouds. It also does not allow us to reproduce the wisps that are a striking visual features of clouds. These wisps are the result of high-frequency density variations of the droplet concentration.

One solution could be to increase the resolution of our representation but it is limited by the fact that it is a heightfield, hence increasing the resolution does not make us more capable of modeling wisps of puffs.

Another solution would be to fake these details using procedural noise at rendering time such as in [Gar85]. It this method, the noise is not used to add detail to the underlying shape representation, but directly to the rendering result. We have tried using such an approach [Ney00] but the issue of this phenomenological method is that it is very difficult to control, and that it assumes convex shapes, thus limiting the type of clouds we could render.

As a consequence, to account for these details we choose to encode the fine volumetric density variations of clouds in a 3D grid. However, since this solution has posed many problems in previous approaches (see §2.6.3.4) we propose a new way of encoding these inhomogeneities. Indeed, we can see on pictures of real clouds (§2.2) that these density variations are really visible only on the cloud-air interface. Even if there are density variations inside the core of the clouds they are not directly visible to the eye. Our study has shown that light transport in thick clouds spread light paths, blurring these inhomogeneities.

Thus, we use an hybrid approach between the classical representation using volumetric grids and our previous representation using only surfaces. This representation assumes that the core of the cloud is homogeneous and stores inhomogeneities only near the boundary of the cloud. The boundary of the cloud is encoded by a surface (see Figure 7.1). This surface $\mathcal{S}$ represents the limit outside which the concentration ρ is 0. Inside this boundary, ρ is assumed to be inhomogeneous within a distance h from $\mathcal{S}$. The rest of the cloud is considered to have a given homogeneous density ρ_0. This approach has similarities with a method commonly used in computer graphics to add 3D details to a surface called *Hypertexture* [PH89].

We encode the inhomogeneities into a 3D GPU-friendly volumetric representation that stores data only for this hypertexture [CN07], similarly to an octree. Thanks to this approach, we do not need a full 3D volumetric grid to store the droplet concentration throughout the whole cloud, which would require a huge amount of memory. We only store the density variations in a thin layer within the cloud boundary, and the density is defined implicitly to be constant under this layer, and null above it.

[1] As we will see in our results, this solid look mainly came from the ignoring of spatial spreading. Indeed, realistic images can be obtained on homogeneous clouds with the rendering method described in this chapter (*e.g.,* see Figure 7.16).

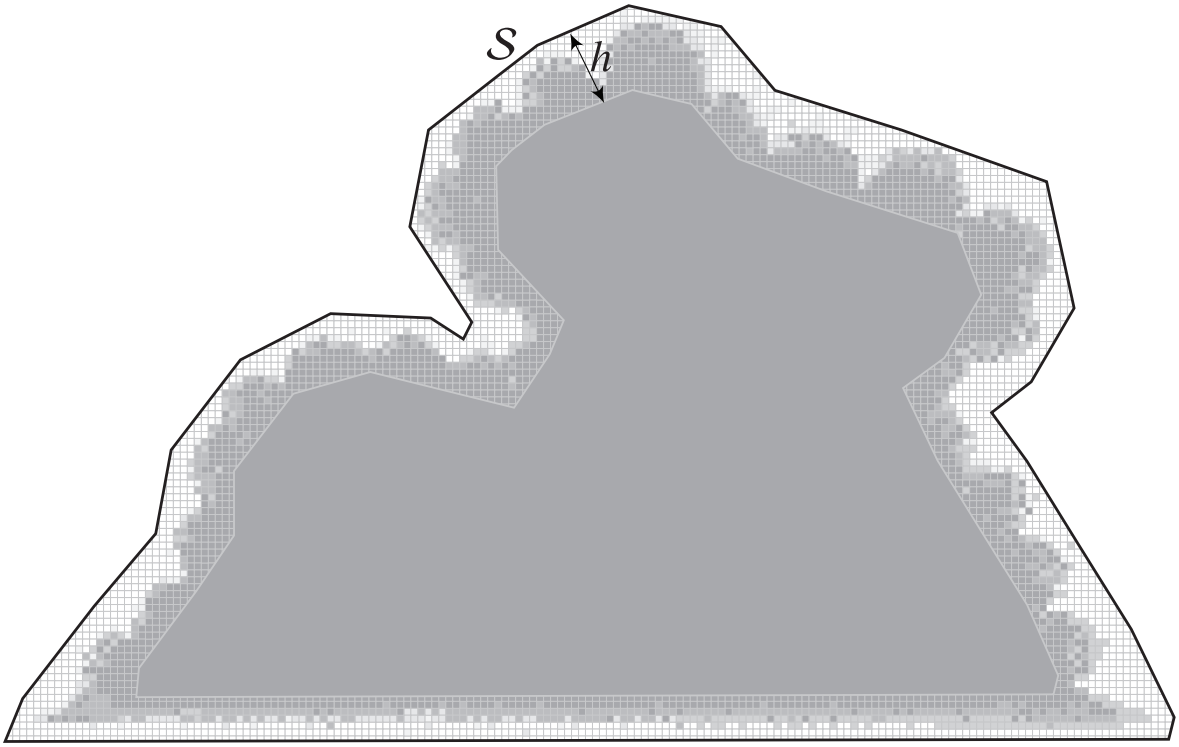


Figure 7.1: *Our cloud representation. The surface $\mathcal{S}$ represents the boundary outside which the droplet density ρ is set to 0. This density is considered inhomogeneous within a distance h inside $\mathcal{S}$. Beyond this distance, the core is set to have a homogeneous density ρ_0.*

7.2.2 Procedural density variations

For a given point $\boldsymbol{p}$, we call $d(\boldsymbol{p})$ the signed closest distance from this point to the boundary $\mathcal{S}$, with $d(\boldsymbol{p}) < 0$ for points outside the boundary and $d(\boldsymbol{p}) > 0$ for points inside the boundary. As we have defined it in the previous section, we know that

$$\rho(\boldsymbol{p}) = 0, \quad d(\boldsymbol{p}) < 0 \tag{7.1}$$

$$\rho(\boldsymbol{p}) = \rho_0, \quad d(\boldsymbol{p}) \geq h. \tag{7.2}$$

What is left for us to do is to define $\rho(\boldsymbol{p})$ for all the points $\boldsymbol{p}$ having a distance $0 < d(\boldsymbol{p}) < h$.

For this we rely on a hypertexture approach [PH89]. The concentration is defined by

$$\rho(\boldsymbol{p}) = \rho_0 s\left(\frac{d(\boldsymbol{p})}{h} + p(\boldsymbol{p})\right), \tag{7.3}$$

where $s(x)$ is a sigmoid function and $p(\boldsymbol{p})$ is a 3D scalar Perlin noise [Per85]. This way, the concentration, on average, varies from 0 at the outer boundary to ρ_0 at the inner boundary.

7.2.3 Optical parameters

Regarding the DSD and the other optical parameters, we use the same values than in our study, that is, the modified gamma distribution and the chopped-peak model with

- $\gamma = 2$,
- $r_e = 6\,\mu m$,
- $\rho_0 = 400\,cm^{-3}$.

7.3 Overview of our light transport model

In this section we show that we can use the subsurface scattering approach to describe light transport in a cloud.

7.3.1 The subsurface scattering approach

A BSSRDF (see §1.5.10) is classically written as

$$S(\boldsymbol{p}, \vec{\boldsymbol{s}}, \boldsymbol{p}', \vec{\boldsymbol{s}}'). \tag{7.4}$$

It encodes light transport between two elements of surface $d\sigma$ and $d\sigma'$ located at $\boldsymbol{p}$ and $\boldsymbol{p}'$, respectively (see Figure 1.11). Traditionally, they belong to the surface of an object within which radiative transfer is arising (*e.g.,* the surface of a marble statue, of human skin or the boundary of a volume of milk). Let us call $\mathcal{V}$ the surface that is viewed by the camera (*viewed surface*) and $\mathcal{I}$ the *illuminated surface*.

viewed surface

illuminated surface

The rendering problem is to determine the luminance $L(\boldsymbol{p}, \vec{\boldsymbol{s}})$ of all points $\boldsymbol{p}$ of the viewed surface $\mathcal{V}$ in the direction $\vec{\boldsymbol{s}}$ of the camera, knowing the illuminance $dE(\boldsymbol{p}', \vec{\boldsymbol{s}}')$ received by all points $\boldsymbol{p}'$ of the illuminated surface $\mathcal{I}$ from all directions $\vec{\boldsymbol{s}}'$. A BSSRDF to solve this problem encodes light transport from the illuminated surface $\mathcal{I}$ to the viewed surface $\mathcal{V}$. If one is provided a model of such a BSSRDF, then one can solve the rendering problem by

$$L(\boldsymbol{p}, \vec{\boldsymbol{s}}) = \int_{\boldsymbol{p}' \in \mathcal{I}} \int_{\vec{\boldsymbol{s}}'} S(\boldsymbol{p}, \vec{\boldsymbol{s}}, \boldsymbol{p}', \vec{\boldsymbol{s}}') dE(\boldsymbol{p}', \vec{\boldsymbol{s}}') d\boldsymbol{p}'. \tag{7.5}$$

It is nowhere stated in the definition of a BSSRDF that $\mathcal{V}$ and $\mathcal{I}$ should be real surfaces. They can be totally virtual and even discontinuous. As written in [NRH+77], *no assumption is made about the mechanism involved other than that there is some form of interaction between radiation and matter by which some of the flux incident at $\boldsymbol{p}$ emerges as reflected flux from $\boldsymbol{p}'$*. That is, a BSSRDF simply encodes radiation transfer between two surfaces in space, no matter how this transfer happens.

In consequence, we can (and will) use a BSSRDF to describe light transport in clouds. In this chapter, we propose to define the surfaces $\mathcal{V}$ and $\mathcal{I}$ as follows. We first define a surface $\mathcal{S}'$ as the boundary enclosing the volume of clouds for which ρ is above a given threshold ρ_1. That is, $\mathcal{S}'$ is the isosurface of value ρ_1 (see Figure 7.2). Note that it is *not* necessarily $\mathcal{S}$ (which is simply an outside bound for the cloud densities). Given this boundary surface $\mathcal{S}'$, we now define the lit surface $\mathcal{I}$ as the lit area of $\mathcal{S}'$ and the viewed surface $\mathcal{V}$ as the visible area of $\mathcal{S}'$ (see Figure 7.2). Given these surfaces, rendering a cloud consists in finding a BSSRDF modeling light transport from $\mathcal{I}$ to $\mathcal{V}$ and use it to integrate Equation 7.5.

We do not use this approach simply for the sake of it. By defining $\mathcal{S}'$ as described above, we implicitly assume that light transport is taking place within the volume bounded by $\mathcal{S}'$. That is, $\mathcal{S}'$ (and thus

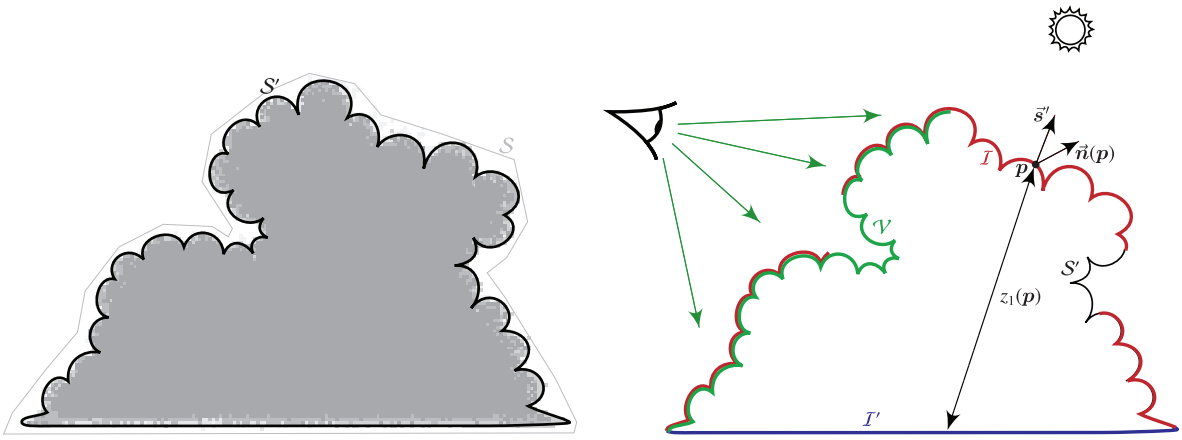


Figure 7.2: *We define $\mathcal{S}'$ as the surface enclosing the volume of clouds for which $\rho > \rho_1$. The surfaces $\mathcal{I}$ and $\mathcal{V}$ between which to compute light transfer are the lit and viewed areas of $\mathcal{S}'$, respectively. We call $\mathcal{I}'$ the unlit area of $\mathcal{S}'$, and $z_1(\boldsymbol{p})$ the distance between $\mathcal{I}$ and $\mathcal{I}'$ in direction $\vec{\boldsymbol{s}}'$ at $\boldsymbol{p}$. We call $\vec{\boldsymbol{n}}(\boldsymbol{p})$ the normal of $\mathcal{S}'$ in $\boldsymbol{p}$.*

$\mathcal{V}$ and $\mathcal{I}$) does not simply serve as a support for describing the problem we have to solve, it also carries information about where light transport is arising. This information will be very useful in our rendering model.

Additionally, we define a few other notions that will be of some use for us. We call $\mathcal{I}'$ the unlit area of $\mathcal{S}'$, *i.e.,* the set of points $\boldsymbol{p}$ of $\mathcal{S}'$ that are the furthest away from the sun in direction $\vec{\boldsymbol{s}}'$ (see Figure 7.2). We call $z_1(\boldsymbol{p})$ the *local thickness* at a given point $\boldsymbol{p}$, defined by the distance between $\mathcal{I}$ and $\mathcal{I}'$ along $\vec{\boldsymbol{s}}'$ at $\boldsymbol{p}$ (see Figure 7.2). Finally, we call $\vec{\boldsymbol{n}}(\boldsymbol{p})$ the normal of $\mathcal{S}'$ in $\boldsymbol{p}$ (see Figure 7.2).

7.3.2 Description of the problem to solve

We can now pose the problem of rendering a cloud as follows. We need to determine the radiance $L(\boldsymbol{p}, \vec{\boldsymbol{s}})$ of the points $\boldsymbol{p}$ of the viewed surface $\mathcal{V}$ in the direction $\vec{\boldsymbol{s}}$ of the camera, knowing the irradiance on the lit surface $\mathcal{I}$. This problem can be solved through a BSSRDF S encoding the radiance transfer between these two surfaces. That is,

$$L(\boldsymbol{p}, \vec{\boldsymbol{s}}) = \int_{\boldsymbol{p}' \in \mathcal{I}} \int_{\vec{\boldsymbol{s}}'} S(\boldsymbol{p}, \vec{\boldsymbol{s}}, \boldsymbol{p}', \vec{\boldsymbol{s}}') dE(\boldsymbol{p}', \vec{\boldsymbol{s}}') d\boldsymbol{p}'. \tag{7.6}$$

Here the illuminance term $dE(\boldsymbol{p}', \vec{\boldsymbol{s}}')$ is defined via sun, sky and ground illumination. What we need to do is find an efficient formulation for the BSSRDF S and an efficient way to compute Equation 7.6.

As in Chapter 6, we tackle this problem by dividing it into several components. We decompose L into three parts: the luminance due to sunlight, skylight and groundlight, *i.e.,*

$$L(\boldsymbol{p}, \vec{\boldsymbol{s}}) = L_{\text{sun}}(\boldsymbol{p}, \vec{\boldsymbol{s}}) + L_{\text{sky}}(\boldsymbol{p}, \vec{\boldsymbol{s}}) + L_{\text{ground}}(\boldsymbol{p}, \vec{\boldsymbol{s}}). \tag{7.7}$$

We solve each of the three components separately. They are described in the following sections.

7.4 Sun illumination

In this section, we propose a model to compute an accurate and efficient approximation of $L_{\text{sun}}(\boldsymbol{p}, \vec{\boldsymbol{s}})$. As all other methods, we consider that the sun illuminates the clouds from a unique incident direction $\vec{\boldsymbol{s}}'$. Thus, Equation 7.6 for L_{sun} comes down to

$$L_{\text{sun}}(\boldsymbol{p}, \vec{\boldsymbol{s}}) = \int_{\boldsymbol{p}' \in \mathcal{I}} S(\boldsymbol{p}, \vec{\boldsymbol{s}}, \boldsymbol{p}', \vec{\boldsymbol{s}}') dE_{\text{sun}}(\boldsymbol{p}', \vec{\boldsymbol{s}}') d\boldsymbol{p}' \tag{7.8}$$

$$= \int_{\boldsymbol{p}' \in \mathcal{I}} S(\boldsymbol{p}, \vec{\boldsymbol{s}}, \boldsymbol{p}', \vec{\boldsymbol{s}}') \vec{\boldsymbol{s}}' \cdot \vec{\boldsymbol{n}}(\boldsymbol{p}') E_{\text{sun}} d\boldsymbol{p}'. \tag{7.9}$$

Since light transport behaves differently with the order of scattering, we decompose $L_{\text{sun}}(\boldsymbol{p}, \vec{\boldsymbol{s}})$ into several components brought by light paths of different order. We use the decomposition that we found in §4.4.8, *i.e.,*

$$\begin{aligned} L_{\text{sun}}(\boldsymbol{p}, \vec{\boldsymbol{s}}) &= L_{\text{sun}}(0, \boldsymbol{p}, \vec{\boldsymbol{s}}) \\ &+ L_{\text{sun}}(1, \boldsymbol{p}, \vec{\boldsymbol{s}}) \\ &+ L_{\text{sun}}(2, \boldsymbol{p}, \vec{\boldsymbol{s}}) \\ &+ L_{\text{sun}}(3, \boldsymbol{p}, \vec{\boldsymbol{s}}) \\ &+ L_{\text{sun}}([4, 5], \boldsymbol{p}, \vec{\boldsymbol{s}}) \\ &+ L_{\text{sun}}([6, 8], \boldsymbol{p}, \vec{\boldsymbol{s}}) \\ &+ L_{\text{sun}}([9, 14], \boldsymbol{p}, \vec{\boldsymbol{s}}) \\ &+ L_{\text{sun}}([15, \infty], \boldsymbol{p}, \vec{\boldsymbol{s}}). \end{aligned} \tag{7.10}$$

We propose different models depending on the order of scattering. For order 0 (transparency), we rely on a ray marching scheme described in §7.4.1. For order 1 (single scattering), we propose an approximate numerical integration described in §7.4.2. For higher orders (multiple scattering), we propose a model relying on the *most probable entry area* paradigm which we describe in §7.4.3.

7.4.1 Model for order 0

To compute $L_{\text{sun}}(0, \boldsymbol{p}, \vec{\boldsymbol{s}})$, we use exactly the same approach as in §6.3.2.1. The only difference is that in the present case the density ρ is not homogeneous. Thus, to compute the optical thickness τ (§2.5.2), we have to integrate it along the view direction, *i.e.,*

$$\tau = \int_0^l \kappa_e(\boldsymbol{p} + x\vec{\boldsymbol{s}}) dx, \tag{7.11}$$

where l is the distance from $\boldsymbol{p}$ to $\mathcal{S}$ in direction $\vec{\boldsymbol{s}}$ (see Figure 7.3).

To compute Equation 7.11, we rely on numerical integration with the standard trapezoidal rule. This equation is transformed into

$$\tau = \sum_{i=1}^{k-1} (x_{i+1} - x_i) \frac{\kappa_e(\boldsymbol{p}_i) + \kappa_e(\boldsymbol{p}_{i+1})}{2}, \tag{7.12}$$

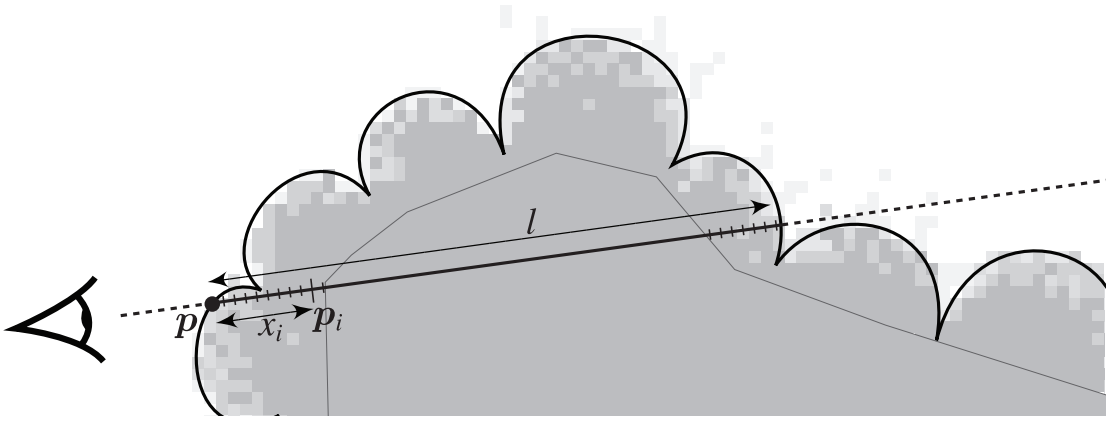


Figure 7.3: *Transparency is computed by marching trough the cloud volume along $\vec{s}$ and integrating the extinction function. We avoid taking small steps in regions we know to be homogeneous.*

with $\boldsymbol{p}_i = \boldsymbol{p} + x_i\vec{s}$. The values x_i are chosen in the range $[0, l]$ and so that the sampling points $\boldsymbol{p}_i$ are located in the regions where $\rho(\boldsymbol{p})$ is inhomogeneous (see Figure 7.3). This is a common method in volume rendering called *ray marching*.

ray marching

The rest is exactly the same as in §6.3.2.1: to recover from the missing peak of the chopped model, we compute the two optical thickness τ_{ref} and τ_{chopped} by computing Equation 7.12 for $\kappa_e = \kappa_{\text{ref}} = \rho(\boldsymbol{p})\sigma_e$ and $\kappa_e = \kappa_{\text{chopped}} = f\rho(\boldsymbol{p})\sigma_e$, respectively. We then compute $L_{\text{sun}}(0, \boldsymbol{p}, \vec{s})$ as a sum of two components: the unscattered radiance and the multiple strongly forward scattered radiance

$$L_{\text{sun}}(0, \boldsymbol{p}, \vec{s}) = \left(\delta(\vec{s} - \vec{s}')e^{-\tau_{\text{ref}}} + P_{\text{peak}}(\Theta)(e^{-\tau_{\text{chopped}}} - e^{-\tau_{\text{ref}}})\right) E_{\text{sun}}. \tag{7.13}$$

7.4.2 Model for order 1

For single scattering, we start from the classical analytical expression using the successive orders of scattering (§2.5.4.3)

$$L_{\text{sun}}(1, \boldsymbol{p}, \vec{s}) = P(\Theta) \int_0^l \beta(\boldsymbol{p}, \boldsymbol{p}_x)\kappa_e(\boldsymbol{p}_x)\beta(\boldsymbol{p}_x, \boldsymbol{p}_{l'})dx, \tag{7.14}$$

with $\boldsymbol{p}_x = \boldsymbol{p} + x\vec{s}$ and $\boldsymbol{p}_{l'}$ the exit point through $\mathcal{I}$ from $\boldsymbol{p}_x$ in direction $\vec{s}'$ (see Figure 7.4). If we write l' the distance from $\boldsymbol{p}_x$ to $\mathcal{S}$ in direction $\vec{s}'$, we have

$$\boldsymbol{p}_{l'} = \boldsymbol{p}_x + l'\vec{s}'.$$

Equation 7.14 expands to

$$L_{\text{sun}}(1, \boldsymbol{p}, \vec{s}) = P(\Theta) \int_0^l \kappa_e(\boldsymbol{p}_x) \exp\left(-\int_0^x \kappa_e(\boldsymbol{p}_{x'})dx'\right) \exp\left(-\int_0^{l'} \kappa_e(\boldsymbol{p}_x + x'\vec{s}')dx'\right) dx, \tag{7.15}$$

The direct numerical integration of Equation 7.15 would be quite expensive due to the inner integrals. We first approximate it by assuming that $\kappa_e(\boldsymbol{p}_x + x'\vec{s}')$ is constant and equal to $\kappa_e(\boldsymbol{p}_x)$. In the following,

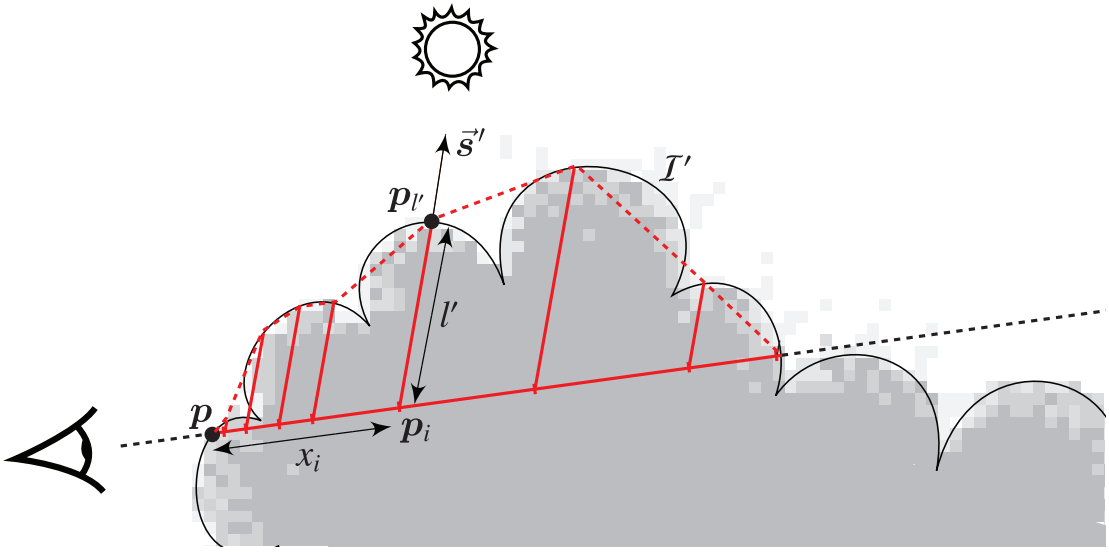


Figure 7.4: *Single scattering is computed by marching trough the cloud volume along $\vec{s}$ and integrating the in-scattered radiance. We use logarithmically-spaced steps. Our numerical integration is equivalent to computing single scattering in the volume enclosed by the dashed red line.*

for simplicity, we write $\kappa(x) = \kappa_e(\boldsymbol{p}_x)$. Using our approximation, Equation 7.15 becomes

$$L_{\text{sun}}(1, \boldsymbol{p}, \vec{\boldsymbol{s}}) = P(\Theta) \int_0^l \kappa(x) \exp\left(- \int_0^x \kappa(x')dx' \right) e^{-l'\kappa(x)} dx. \tag{7.16}$$

The remaining inner integral does not need to be approximated. Indeed, we repeatedly integrate over the same domain. We can optimize this by accumulation as follows.

For simplicity, let us call $F(x)$ the outer integrand and $G(x)$ the inner integral of Equation 7.16, *i.e.,*

$$L_{\text{sun}}(1, \boldsymbol{p}, \vec{\boldsymbol{s}}) = P(\Theta) \int_0^l F(x)dx, \tag{7.17}$$

$$F(x) = \kappa(x) e^{-l'\kappa(x)} e^{-G(x)}, \tag{7.18}$$

$$G(x) = \int_0^x \kappa(x')dx'. \tag{7.19}$$

Note the following property

$$G(b) = G(a) + \int_a^b \kappa(x')dx'. \tag{7.20}$$

This is the property we make use of for integrating numerically Equation 7.17. Using the trapezoidal approximation, it gives

$$L_{\text{sun}}(1, \boldsymbol{p}, \vec{\boldsymbol{s}}) = P(\Theta) \sum_{i=1}^{k-1} (x_{i+1} - x_i) \frac{F(x_i) + F(x_{i+1})}{2}, \tag{7.21}$$

$$F(x_i) = \kappa(x_i) e^{-l' \kappa(x_i)} e^{-G(x_i)}, \tag{7.22}$$

$$G(x_i) = \begin{cases} G(x_{i-1}) + (x_i - x_{i-1}) \dfrac{\kappa(x_i) + \kappa(x_{i-1})}{2}, & i > 1, \\ 0, & i = 1. \end{cases} \tag{7.23}$$

This gives us a numerical integration of Equation 7.16. We use k sampling points $x_1, \ldots, x_k$ in the range $[0, l]$. Since $F(x)$ decreases exponentially with x, we space the sampling points logarithmically (see Figure 7.4).

7.4.3 Multiple scattering model

7.4.3.1 Overview

This section describe how we compute radiance transfer by paths of order 2 and more. For each order (or group of orders) $n \in \{2, 3, [4, 5], [6, 8], [9, 14], [15, \infty]\}$, we use the following approach.

This approach uses the knowledge we gained from our study in Chapter 4. It draws on the fact that in a slab the entry points of light paths for a given order n are located according to an exponential-to-gaussian distribution D around a center $\boldsymbol{c}$ with a standard deviation σ.

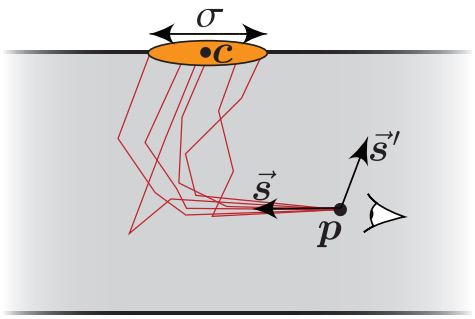
collector area

For a given order n, we call the *collector area* (or *collector*) C the area on the lit slab boundary enclosing the 95% most probable entry points of paths of order n. One of the findings of our study (§4.4.3) is that for a slab, this collector area can be considered as a disk of radius σ (the second moment of D) centered around $\boldsymbol{c}$ (the first moment of D) as pictured on Figure 7.5(a).

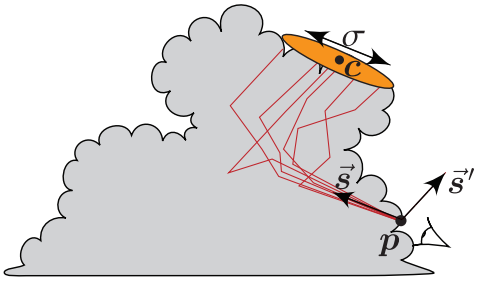
Our first hypothesis is to assume that in a cloud of arbitrary shape, this collector area is also a disk defined by a center $\boldsymbol{c}$ and a radius σ, located somewhere on the lit surface $\mathcal{I}$ (see Figure 7.5(b)). By definition, the role of the cloud surface outside the collector is negligible (Figure 7.5(c)). As a consequence, we locally approximate the cloud shape as a plane-parallel slab aligned on the collector to simplify the computation of light transport (Figure 7.5(d) – 7.5(e)). Given this slab, we can use our procedural BSDF model of §5.4 to compute light transport for light paths of order n in the cloud (Figure 7.5(f)).

This approach can also be seen from the point of view of the most probable paths. The approach of [PAS03] is to find the most probable path arriving at $\boldsymbol{p}$ in direction $\vec{\boldsymbol{s}}$, then compute the radiance transport along this path. Our approach is to find the most probable *entry area* of light paths arriving at $\boldsymbol{p}$ in direction $\vec{\boldsymbol{s}}$ *for each order* (or group of orders) of scattering. This allows us to account for all types of significant light paths (from the short, anisotropic ones due to low orders to the long, diffusive ones due to high orders). Moreover, instead of walking the volume to find these most probable paths, we only have to walk the lit surface $\mathcal{I}$ to find the most probable entry area, which is much more efficient.

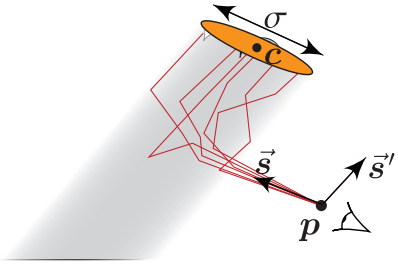
As a result, our approach is summarized as follows. For each order $n \in \{2, 3, [4, 5], [6, 8], [9, 14], [15, \infty]\}$, we



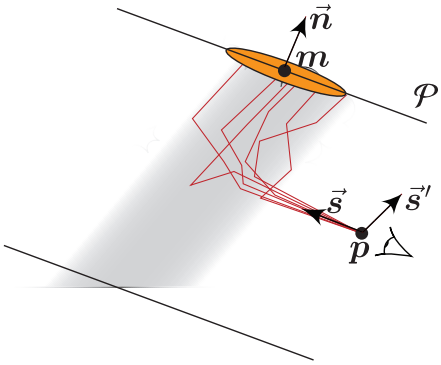
(a) In a slab, for a given order n, the collector is a disk of center $\boldsymbol{c}$ and radius σ



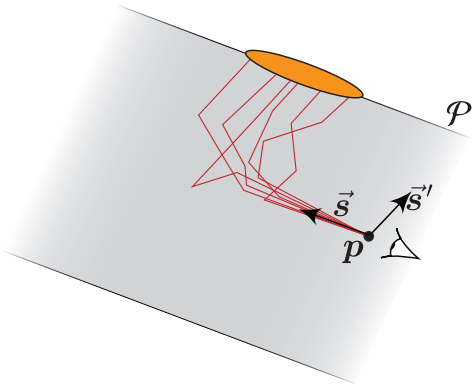
(b) We assume this collector C also has a disk shape in an arbitrary cloud.



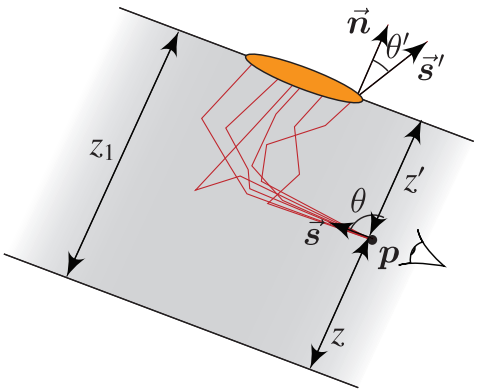
(c) By considering only the part of the lit surface $\mathcal{I}$ and the thickness that is inside the collector, we obtain the most significant top surface and the associated thickness.



(d) We fit a slab on this most significant part of the cloud.



(e) We can now represent the problem as light transport in a slab.



(f) Thanks to our light transport study, we can compute the radiative transfer in a slab.

Figure 7.5: *The main principle of our multiple scattering model.*

1. Search for the center $\boldsymbol{c}$ and size σ of the collector C on $\mathcal{I}$.
2. Find the slab that fits the cloud at the collector location.
3. Deduce the viewing coordinates z, θ, ϕ and lighting coordinates θ', ϕ' in the reference frame of this equivalent slab

4. Obtain the radiance $L(n, \boldsymbol{p}, \vec{\boldsymbol{s}})$ by using the BSDF $S(n, z_1, z, \theta, \phi, \theta', \phi')$ from our procedural model of §5.4 and compute Equation 7.6 as

$$L_{\text{sun}}(\boldsymbol{p}, \vec{\boldsymbol{s}}) \quad = \quad S(n, z_1, z, \theta, \phi, \theta', \phi') \cos\theta' E_{\text{sun}}. \tag{7.24}$$

The rest of this section proposes an algorithm to perform these steps in realtime on GPU. We discuss the validity and limits of this method and of our algorithm in §7.8.

7.4.3.2 Notation

As noted in the overview (§7.3.1 and Figure 7.2), we write $\mathcal{I}$ the lit area of $\mathcal{S}'$, *i.e.,* the set of points that are closest to the light source in the light direction $\vec{\boldsymbol{s}}'$. Similarly, we write $\mathcal{V}$ the viewed area of $\mathcal{S}'$, *i.e.,* the set of points that are closest to the camera in the view direction $\vec{\boldsymbol{s}}$. Moreover, we define the "unlit" area $\mathcal{I}'$ as the set of points that are farthest from the light source in the light direction $\vec{\boldsymbol{s}}'$. We write $\vec{\boldsymbol{n}}(\boldsymbol{p})$ the normal of $\mathcal{S}'$ in $\boldsymbol{p}$. Finally, we define the "local thickness" $z_1(\boldsymbol{p})$ as the distance between $\mathcal{I}$ and $\mathcal{I}'$ in direction $\vec{\boldsymbol{s}}'$ (see Figure 7.2).

The surface $\mathcal{V}$ represents the surface on which we need to compute the radiance for rendering. The surface $\mathcal{I}$ represents the surface on which we look for the collectors (§7.4.3.3). The thickness $z_1(\boldsymbol{p})$ and the normal $\vec{\boldsymbol{n}}(\boldsymbol{p})$ will be used to deduce the thickness of the slab fitting the cloud at the collector location (§7.4.3.4).

In the following sections we explain how we find the collector and how we find the slab fitting the cloud at the collector location. The collector is characterized by its center $\boldsymbol{c}$ and its size σ. We explain how we find this collector in §7.4.3.3. The slab is characterized by its thickness z_1 and its lit plane $\mathcal{P}$. This lit plane is defined by a point $\boldsymbol{m}$ on this plane and a normal $\vec{\boldsymbol{n}}$. We explain how we find the slab given the collector in §7.4.3.4

Given the lit plane, the viewpoint $\boldsymbol{p}$, view direction $\vec{\boldsymbol{s}}$ and incident direction $\vec{\boldsymbol{s}}'$ can be expressed in the reference frame of the slab (Figure 7.5(f)) as

$$z(\boldsymbol{p}, \vec{\boldsymbol{s}}, \boldsymbol{p}', \vec{\boldsymbol{s}}', \boldsymbol{m}, \vec{\boldsymbol{n}}, z_1) \quad = \quad z_1 - (\boldsymbol{m} - \boldsymbol{p}) \cdot \vec{\boldsymbol{n}} \tag{7.25}$$

$$\theta(\boldsymbol{p}, \vec{\boldsymbol{s}}, \boldsymbol{p}', \vec{\boldsymbol{s}}', \boldsymbol{m}, \vec{\boldsymbol{n}}, z_1) \quad = \quad \arccos(\vec{\boldsymbol{s}} \cdot \vec{\boldsymbol{n}}) \tag{7.26}$$

$$\theta'(\boldsymbol{p}, \vec{\boldsymbol{s}}, \boldsymbol{p}', \vec{\boldsymbol{s}}', \boldsymbol{m}, \vec{\boldsymbol{n}}, z_1) \quad = \quad \arccos(\vec{\boldsymbol{s}}' \cdot \vec{\boldsymbol{n}}) \tag{7.27}$$

$$\psi(\boldsymbol{p}, \vec{\boldsymbol{s}}, \boldsymbol{p}', \vec{\boldsymbol{s}}', \boldsymbol{m}, \vec{\boldsymbol{n}}, z_1) \quad = \quad \arccos\left(\frac{\vec{\boldsymbol{s}} - \vec{\boldsymbol{n}}(\vec{\boldsymbol{s}} \cdot \vec{\boldsymbol{n}})}{\|\vec{\boldsymbol{s}} - \vec{\boldsymbol{n}}(\vec{\boldsymbol{s}} \cdot \vec{\boldsymbol{n}})\|} \cdot \frac{\vec{\boldsymbol{s}}' - \vec{\boldsymbol{n}}(\vec{\boldsymbol{s}}' \cdot \vec{\boldsymbol{n}})}{\|\vec{\boldsymbol{s}}' - \vec{\boldsymbol{n}}(\vec{\boldsymbol{s}}' \cdot \vec{\boldsymbol{n}})\|}\right) \tag{7.28}$$

We also write z' the distance between $\boldsymbol{p}$ and $\mathcal{P}$, *i.e.,* $z' = z_1 - z$. These coordinates have exactly the same meaning as in our study (§4.2).

7.4.3.3 Finding the collector

Overview To find the collector, we start from a simple observation. Our procedural model described in §5.3 provides us with a procedural formulation for the center of the collector $\boldsymbol{c}(n, z_1, z, \theta, \psi, \theta')$ and its size $\sigma(n, z_1, z, \theta, \psi, \theta')$ in a slab. Thus, if we knew the slab fitting the cloud at the collector location, we could deduce the collector location and size.

This is a chicken-and-egg problem. In mathematical terms, we are looking for a fixed point. A fixed point $\hat{x}$ of a function $f(x)$ is a value for which $\hat{x} = f(\hat{x})$. In our case, we are looking for the values $(\hat{\boldsymbol{c}}, \hat{\sigma})$, defined by

$$\hat{\boldsymbol{c}} = \boldsymbol{c}(n, \hat{z_1}, \hat{z}, \hat{\theta}, \hat{\psi}, \hat{\theta'}), \tag{7.29}$$
$$\hat{\sigma} = \sigma(n, \hat{z_1}, \hat{z}, \hat{\theta}, \hat{\psi}, \hat{\theta'}), \tag{7.30}$$

where $\hat{z_1}, \hat{z}, \hat{\theta}, \hat{\psi}, \hat{\theta'}$ are obtained from $\hat{\boldsymbol{c}}, \hat{\sigma}$ via Equations 7.25 – 7.28.

One way to find a fixed point is to use an iterative approach. Starting with a value x_0, one computes $x_1 = f(x_0)$, then $x_2 = f(x_1)$, and so on. If f is well conditioned and if x_0 is "close enough" to $\hat{x}$, this method converges to

$$\lim_{n\to\infty} x_n = \hat{x}. \tag{7.31}$$

This method is very sensitive to the initial value and requires f to behave correctly, or it may diverge. As an example, it will find the fixed point $\hat{x} = 0$ of $f(x) = x/2$ for any x_0, but it will not find the fixed point $\hat{x} = 0$ of $f(x) = 2x$ for any $x_0 \neq 0$. Another issue is that it might come back to the same point after a finite number of iteration, *i.e.,* $x_n = x_{n+k}$. In that case, it does not diverge, but it does not converge either. These points are called *periodic points*.

Another way to find a fixed point is to look for the roots of $f(x) - x$. A very wide range of root-finding methods exist out there. The problem is that either they require more knowledge about the function (*e.g.,* to know its derivatives) or they are too computationally expensive for our realtime goals. Moreover, they may not give better results than the iterative approach[2].

Finally, and maybe more importantly, any given algorithm requires that this solution $(\hat{\boldsymbol{c}}, \hat{\sigma})$ exists and is unique. We discuss this issue in §7.8. For now, we assume it is existing and unique.

Algorithm We aim for simplicity and efficiency by using the iterative algorithm with carefully chosen initial points. That is,

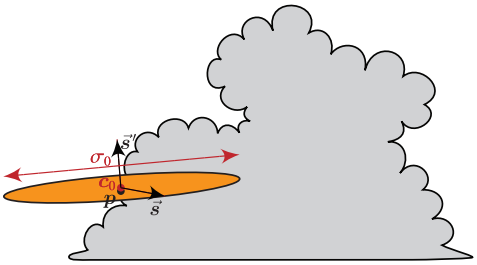
1. We start at $i = 0$ with initial values $\boldsymbol{c}_0$ and σ_0.
2. From $\boldsymbol{c}_i$ and σ_i we compute $\boldsymbol{m}$, $\vec{\boldsymbol{n}}$ and z_1 by fitting a slab onto the cloud at the collector location as described in §7.4.3.4
3. From this slab we deduce z, θ, ψ, θ' as described in §7.4.3.2
4. With these values we compute the new collector using our model of §5.3

$$\boldsymbol{c}_{i+1} = \boldsymbol{c}(n, z_1, z, \theta, \psi, \theta'), \tag{7.32}$$
$$\sigma_{i+1} = \sigma(n, z_1, z, \theta, \psi, \theta'). \tag{7.33}$$

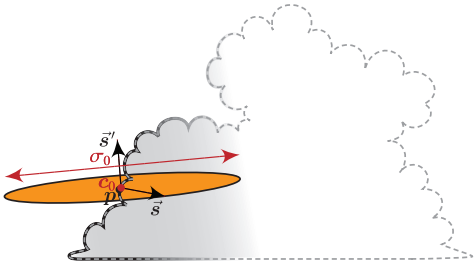
5. We then loop over to step 2 with $i = i + 1$.

This loop is repeated until $\boldsymbol{c}_{i+1} \approx \boldsymbol{c}_i$. In that case, we take $\hat{\boldsymbol{c}} = \boldsymbol{c}_i$ and $\hat{\sigma} = \sigma_i$. This algorithm is summarized on Figures 7.6 – 7.7 pages 216 – 217.. We discuss the choice of $\boldsymbol{c}_0$ and σ_0 and the convergence of this algorithm in §7.8.

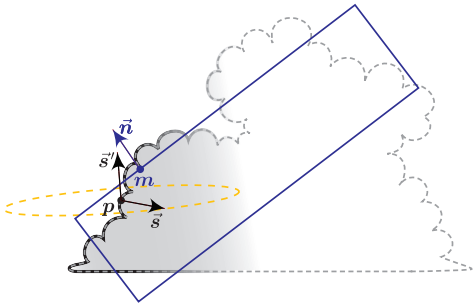
[2] Indeed, as we will see in §7.8, the main error we do is to assume that the collector is a unique disk.



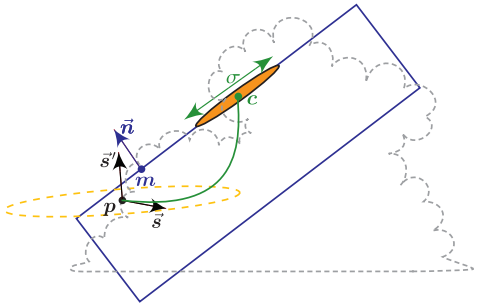
(a) We start with an initial collector (orange disk) of center $\boldsymbol{c}_0$ and size σ_0.



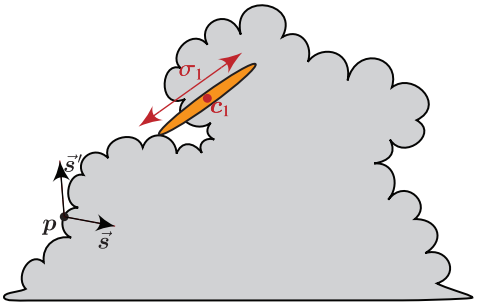
(b) We consider only the significant cloud surface and thickness (grey area) under the collector.



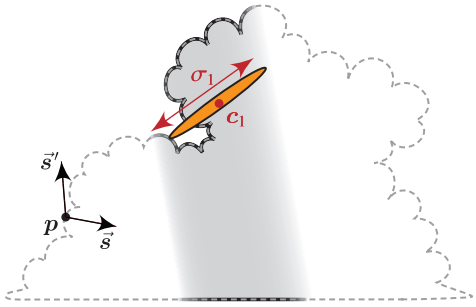
(c) We search for a slab (blue rectangle) fitting this significant cloud part.



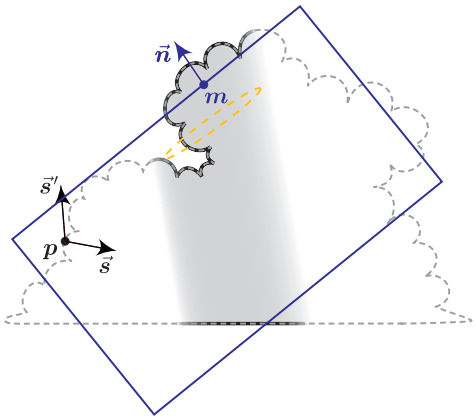
(d) Our study gives us a new collector (orange disk) of center $\boldsymbol{c}$ and size σ for this slab.



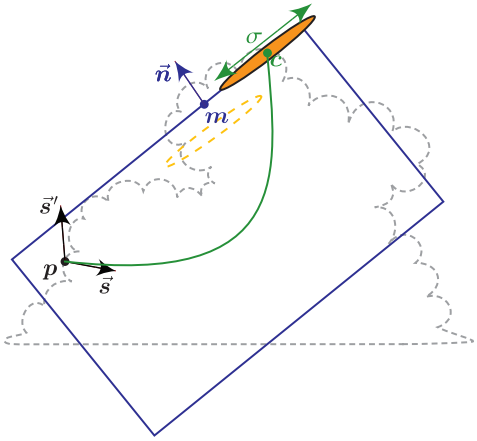
(e) We repeat the process with $\boldsymbol{c}_1 = \boldsymbol{c}$ and $\sigma_1 = \sigma$...



(f)

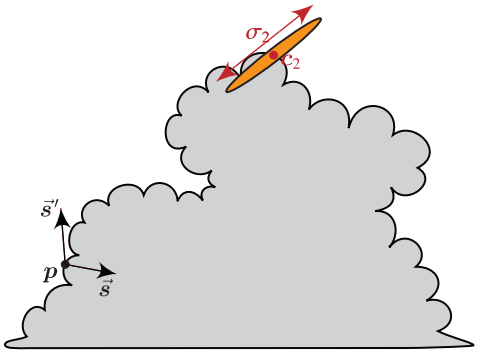


(g)

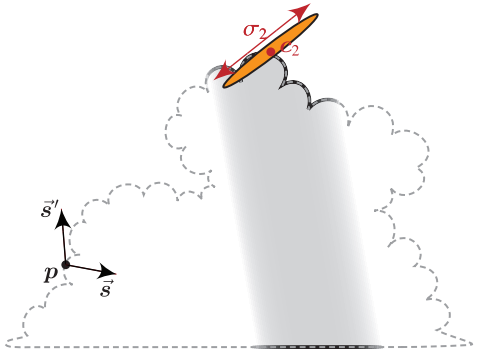


(h)

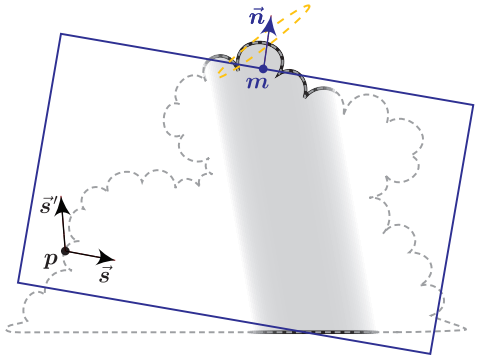
Figure 7.6: *The first steps of our algorithm.*



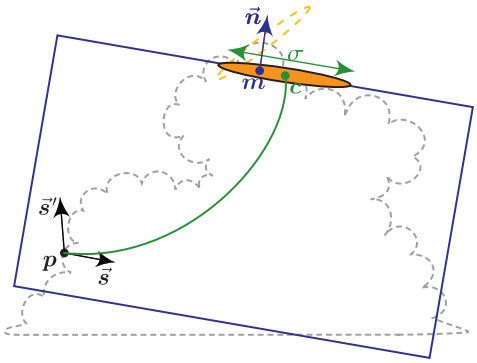
(a) ...and again with c_2 and σ_2...



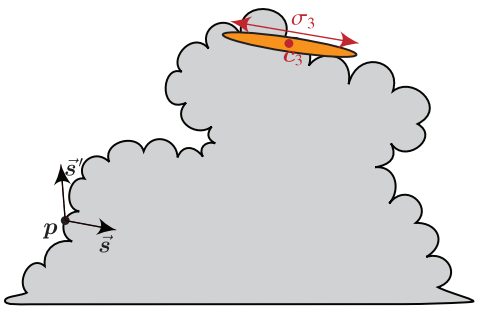
(b)



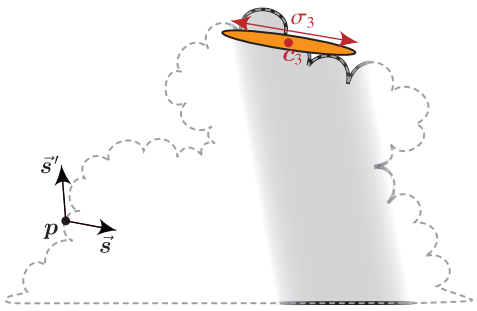
(c)



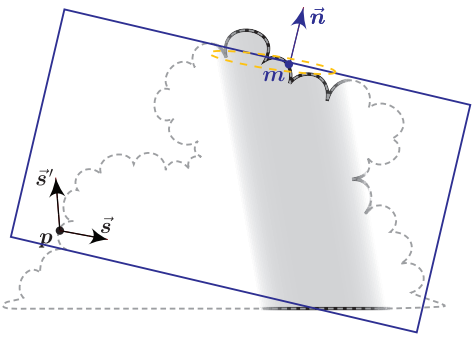
(d)



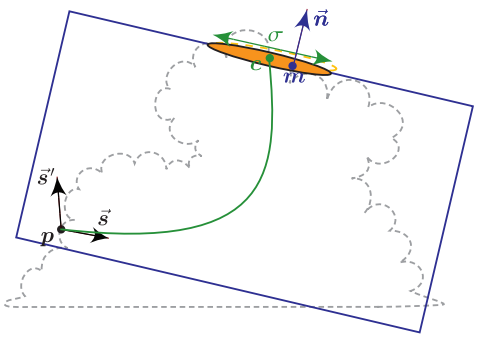
(e)



(f)



(g)



(h) ...Until we converge, *i.e.*, $\boldsymbol{c} \approx \boldsymbol{c}_3$.

Figure 7.7: *The last steps of our algorithm.*

7.4.3.4 Finding the fitting slab

We write $\mathcal{P}$ the plane approximating $\mathcal{I}$ at the collector location. This plane can be described by a point $\boldsymbol{m}$ and a normal $\vec{\boldsymbol{n}}$. According to our study (§4.4.3), the entry points of light are distributed with a somewhat exponential behavior around the center $\boldsymbol{c}$ of the collector. Thus, to approximate $\mathcal{I}$ by a plane, we convolve the lit surface $\mathcal{I}$ by an exponential distribution D whose mean is the center $\boldsymbol{c}$ of the collector and whose standard deviation is the size σ of the collector, *i.e.,*

$$\boldsymbol{m} = \int_{\boldsymbol{p}' \in C} \boldsymbol{p}' D(\boldsymbol{c}, \sigma, \boldsymbol{p}') d\boldsymbol{p}', \tag{7.34}$$

$$\vec{\boldsymbol{n}} = \int_{\boldsymbol{p}' \in C} \vec{\boldsymbol{n}}(\boldsymbol{p}') D(\boldsymbol{c}, \sigma, \boldsymbol{p}') d\boldsymbol{p}', \tag{7.35}$$

with D the distribution defined by

$$D(\boldsymbol{c}, \sigma, \boldsymbol{p}') = \frac{1}{\sigma} e^{-\frac{1}{\sigma}\|\boldsymbol{c}-\boldsymbol{p}'\|} \tag{7.36}$$

Note that $\boldsymbol{m}$ is not necessarily on $\mathcal{I}$. Similarly, we find the thickness z_1 of the equivalent slab by convolving $z_1(\boldsymbol{p})$ in the same way, *i.e.,*

$$z_1 = \int_{\boldsymbol{p}' \in C} z_1(\boldsymbol{p}') D(\boldsymbol{c}, \sigma, \boldsymbol{p}') d\boldsymbol{p}'. \tag{7.37}$$

We call this operation the *filtering* stage. It allows us to account for the spatial spreading of light through the cloud volume.

7.4.3.5 Summary

To summarize, our multiple scattering model works as follows. For each order $n \in \{2, 3, [4, 5], [6, 8], [9, 14], [15, \infty]\}$, we

1. Search for the center $\boldsymbol{c}$ and size σ of the collector C on $\mathcal{I}$ by using our iterative algorithm described in §7.4.3.3. This algorithm
   (a) Starts from an initial center $\boldsymbol{c}_0$ and size σ_0 (§7.4.3.3).
   (b) Fits a slab fitting the cloud at the collector location through filtering the cloud surface and thickness (§7.4.3.4).
   (c) Deduces the viewing coordinates z, θ, ϕ and lighting coordinates θ', ϕ' in the reference frame of this equivalent slab (§7.4.3.2)
   (d) Computes a new collector center $\boldsymbol{c}_i$ and size σ_i from our model described in §5.3
   (e) Loops until it has converged to a fixed collector center $\hat{\boldsymbol{c}}$ and size $\hat{\sigma}$ (§7.4.3.3)
2. Uses the slab thickness $\hat{z_1}$, viewing coordinates $\hat{z}$, $\hat{\theta}$, $\hat{\phi}$ and lighting coordinates $\hat{\theta'}$, $\hat{\phi'}$ (§7.4.3.2) associated with the solution $(\hat{\boldsymbol{c}}, \hat{\sigma})$ to obtain the radiance $L(n, \boldsymbol{p}, \vec{\boldsymbol{s}})$ by computing the BSDF $S(n, \hat{z_1}, \hat{z}, \hat{\theta}, \hat{\psi}, \hat{\theta'})$ from our procedural model of §5.4.
3. Use this BSDF to compute the luminance via $L(n, \boldsymbol{p}, \vec{\boldsymbol{s}}) = S(n, \hat{z_1}, \hat{z}, \hat{\theta}, \hat{\psi}, \hat{\theta'}) \cos\theta' E_{\text{sun}}$ (Equation 7.24).

In the end, this method provides us with the luminance $L(n, \boldsymbol{p}, \vec{\boldsymbol{s}})$ due to all orders n of multiple scattering. This luminance can be used to render the cloud.

7.5 Sky and ground illumination

The luminance due to the sky and the ground are computed essentially in the same way than for the sun contribution. The difference is that sky and ground are diffuse sources while our light transport model is designed for directional sources.

The ideal way to handle diffuse sources would be to have a BSDF model designed for diffuse sources, but we did not have the time to find such a model (we believe it is possible to find a model very similar to the directional one from our study). Another solution would be to use the directional model to compute luminance for numerous incident directions, but this would be too costly.

As a result, we consider the sky and ground as directional sources. The sky is a vertical source located above the clouds while the ground is a vertical source illuminating the cloud from below. If we were to directly apply our light transport model of §7.3 for these sources, it would result in strongly visible view-dependent effects (*e.g.,* blue silver lining when looking at the clouds from below) which are not desirable since the sky and the ground bring soft, flat tints.

As a result, we only account for transparency (order $n = 0$) and light paths of high order ($n \geq 16$) and we ignore the contribution of light paths of lower orders for these sources. This clearly underestimates the energy due to these sources and a better model should be found. However, it brings the brownish and blueish tints to the clouds in the expected locations.

7.6 Implementation of our model on GPU

7.6.1 Cloud shape representation

The method we use to represent the shape described in §7.2.1 on GPU is explained as follows. The boundary surface $\mathcal{S}$ is stored as a mesh, which can be quite coarse. In the examples shown in §7.7, we use a mesh composed of 5,000 triangles. The underlying Hypertexture, as explained in §7.2.1, is defined through a *distance field* $d(\boldsymbol{p})$ and a noise function $p(\boldsymbol{s})$. We store the distance field $d(\boldsymbol{p})$ into an optimized structure [CN07] (roughly speaking, an *octree* suited for GPU) storing $d(\boldsymbol{p})$ only in the necessary locations (see Figure 7.1). If the cloud shape is animated, this distance field can quickly be regenerated using a technique described in [CN07]. We choose not to store the noise function $p(\boldsymbol{s})$ in memory and instead to evaluate it on the fly on the GPU when necessary. For this we use a GPU implementation [Gus05] of Perlin noise [Per85]

7.6.2 Surface representations

Before rendering each frame, we compute the necessary surfaces on which our BSSRDF is defined (§7.3.1) into GPU-friendly representations. These surfaces ($\mathcal{I}$, $\mathcal{I}'$, $\mathcal{V}$) are computed using the GPU and stored as depth maps (see Figure 7.2). This is done by rendering the bounding mesh $\mathcal{S}$ with a shader performing ray marching on the GPU. This shader marches through the volume, starting from $\mathcal{S}$ until it reaches the isosurface $\mathcal{S}'$ of value ρ_1, *i.e.,* until hits a voxel of density $\rho(\boldsymbol{p}) > \rho_1$. Doing this from the cameras's point of view allow us to compute the viewed surface $\mathcal{V}$, and doing this from the sun, sky and ground point of views allow us to compute $\mathcal{I}$ and $\mathcal{I}'$ for each of these light sources. The local thickness $z_1(\boldsymbol{p})$ is found by computing the difference between the two depth maps $\mathcal{I}$ and $\mathcal{I}'$. We call it a "*thickness map*". $\vec{\boldsymbol{n}}(\boldsymbol{p})$ is a normal map and is computed from $\mathcal{I}$ using finite differences.

thickness map

7.6.3 Rendering

We then render the visible surface $\mathcal{V}$ with a shader computing the whole light transport model described in §7.3, including transparency, single scattering and multiple scattering. This shader takes as input the maps described above and proceeds as follows.

7.6.3.1 Order 0 (transparency)

The implementation of our model for order 0 (§7.4.1) on GPU is straightforward. For each pixel, the shader marches through the volume and evaluates the density $\rho(\boldsymbol{p})$ at each step using the distance field $d(\boldsymbol{p})$ and the procedural noise function $p(\boldsymbol{p})$ in order to compute Equation 7.13. The optimized structure of [CN07] also allows the shader to know whether it is in the homogeneous core of the cloud, hence to take larger steps.

7.6.3.2 Order 1 (single scattering)

The implementation of our model for order 1 (§7.4.2) is very similar to that of order 0. In the case of single scattering, we take 10 logarithmically-spaced steps. Equation 7.22 requires to know the distance l' between the marching location $\boldsymbol{p}_{x_i}$ and the lit surface $\mathcal{I}$. This is easily done by reading into the depth map representing $\mathcal{I}$ at $\boldsymbol{p}_{x_i}$, like for a shadow map.

7.6.3.3 Multiple scattering

Our multiple scattering model (§7.4.3.5) requires three steps. We must be able to

- Fit a slab against the cloud at the collector location (§7.4.3.4).
- Compute a collector center and size from our model described in §5.3.
- Compute the BSDF from our model described in §5.4.

Step 7.6.3.3 require us to convolve the lit surface $\mathcal{I}$ and the thickness $z_1(\boldsymbol{p})$ with an exponential kernel D. We approximate this kernel D by a sum of 3 box kernels. Convolving the depth map representing $\mathcal{I}$ and the thickness map representing $z_1(\boldsymbol{p})$ with a box kernel comes down to reading into these maps at the right MIP-map level. As a result, this filtering stage is performed efficiently using three texture reads.

Steps 7.6.3.3 and 7.6.3.3 require us to implement our models described in §5.3 and §5.4. This implementation is straightforward. The necessary parameters for these models are passed as textures. These textures take a total space of 2MB on GPU, which is fairly light.

The multiple scattering algorithm itself as described in §7.4.3.5 is also implemented on GPU. It loops over each of the 7 groups of order of scattering. For each of those, it iteratively finds the collector and computes the associated light transport.

7.6.3.4 Result of rendering

Luminance for orders $n = 0$ (§7.6.3.1), $n = 1$ (§7.6.3.2) and $n \geq 2$ (§7.6.3.3) for the sun contribution is computed and summed up. The contribution from the ground and sky is also computed for orders

$n \geq 16$ (§7.5) using the same shader as for the sun contribution. The result is then displayed on screen, blended with the background using the transparency computed in §7.6.3.1.

7.6.4 Aerial perspective and HDR rendering

We use the realtime version [HP03] of the model of Preetham *et al.* [PSS99] to render the sky and account for aerial perspective on the cloud. To handle HDR we use a simplified, unblurred version of [GWWH03].

7.6.5 Rendering cost

We only consider the cost of texture reads and not instruction execution since texture access is the obvious bottleneck here.

We write C_p the cost of evaluating the Perlin noise. Our noise is made of 5 octaves. Evaluating each octave requires 12 texture reads. This yields $C_p = 60$.

The cost of computing the surfaces (§7.6.2) and order 0 (§7.6.3.1) is proportional to the number of steps taken in our ray marcher. Let us write k the average number of steps. Since we perform 7 raymarching passes, the raymarching cost is $7 \times k \times C_p$. Our ray-marching implementation is very unoptimized, taking very small numerous steps, resulting in $k \approx 20$. This yields about $8,000$ texture reads per pixels.

Computing order 1 (§7.6.3.1) takes 10 iteration, thus its cost is $O(10C_p) = 600$.

Computing multiple scattering for one order (or group of orders) takes up to 10 iterations for finding the collector. Each of these iterations involves reading 3 times in the depth map and 3 times in the normal map for the filtering stage. The cost of evaluating the phenomenological models is also a few texture reads. Thus one order of multiple scattering costs about 100 texture reads.

Since we compute these multiple scattering computations for 7 sets of order of scattering for the sun, plus one for the sky and one for the ground, the computation of multiple scattering is roughly 1,000 texture reads.

As we can see, we perform approximately 10,000 texture reads at each frame to render our clouds. Per pixel. This is a huge number and it is actually surprising to us that we reach interactive speeds. Most of the memory accesses are in the ray-marcher, which is clearly unoptimized and is not at all the contribution of this work. The choice of evaluating the Hypertexture noise on the fly is clearly wrong and would divide by almost 60 the number of texture reads.

We believe that the texture reads concerning the multiple scattering computations are very coherent (taking advantage of the GPU memory cache), which explains that we are able of reaching 10FPS on homogeneous clouds (where ray marching is not necessary).

7.7 Results

Our implementation was done on a Pentium 4 at 1.86 GHz with a nVidia 8800 GTS graphics board. All benchmarks were done at resolution 800×600. Figures 7.8 – 7.20 show the results of our method.

Figure 7.8: *A cumulus cloud at 2 FPS. Note the strong contrast between the crisp, bright lit parts and the soft, diffusive, unlit parts. Note also the blue and brown tints on top and bottom and the sharp puffy edges due to an appropriate shape model.*

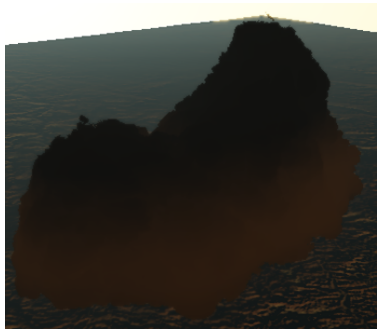
We tested our method on various cloud shapes. We first used a cloud slab to validate the results. We used the animated stratocumulus layers of Chapter 6 which allowed us to verify that our rendering model supports animation and other cloud shapes. Finally we used a cumulus cloud as described in §7.2, with and without the Hypertexture. Note that for homogeneous models (*i.e.,* all models except the one with Hypertexture), the ray marching steps are not necessary. Since our implementation of ray marching on GPU is highly unoptimized, this drastically affects the framerate.

The slab (Figure 7.18) allows us to test our algorithm and validate directly its results against the canonical transport function. It also allows us to see features such as anisotropic reflectance, anisotropic transmittance in thin slabs and isotropic transmittance in thick slabs. The stratocumulus layer (Figures 7.14, 7.16, 7.17) allows us to test our method against a shape close to the slab and to validate the handling of animations. It is composed of 130K triangles, and the framerate is 10 FPS. We can see on this example all the desired cloud features: anisotropic scattering, diffusion, backscattering, glory, fogbow. The cumulus cloud model (Figures 7.8, 7.10, 7.9, 7.12, 7.13) allows us to test our algorithm on an arbitrary shape. It is composed of 5K triangles with a 512^3 distance field. We obtain a framerate of 2 FPS on this model. Note that most of the time is spent in evaluating the procedural noise on the fly in the Hypertexture during ray marching. Without the Hypertexture, the framerate rises up to 10 FPS. This model also displays all the desired cloud features. 10 iterations of our collector-finding algorithm are sufficient in all cases to reach convergence. Figures 7.19 – 7.20 show the result of our

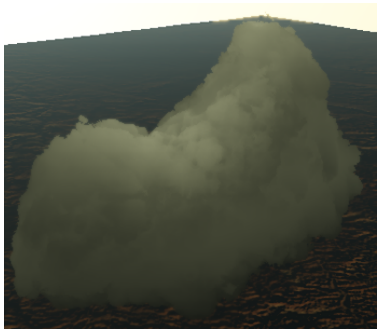
Figure 7.9: *A cumulus cloud at 2 FPS. Again, note the strong contrast between the lit and unlit parts.*

method on an arbitrary mesh with using Hypertextures. The mesh vertices have been jittered to add detail on the silhouette.

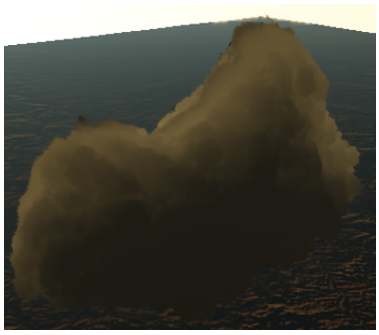
Figure 7.10: *A cumulus cloud at 2 FPS. Note the strong contrast between the silver lining and the soft, diffusive, unlit parts. Note also the grey and brown tints on top and bottom and the sharp puffy edges due to an appropriate shape model. Figure 7.11 shows the contributions of each of the sources and orders of scattering resulting in this image.*



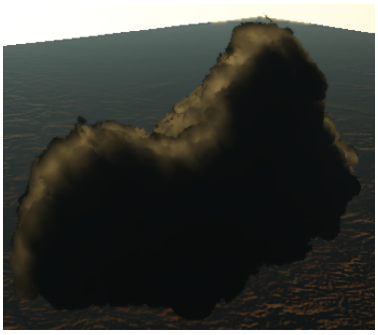
(a) Ground illumination. Note the subtle red tints at the bottom.



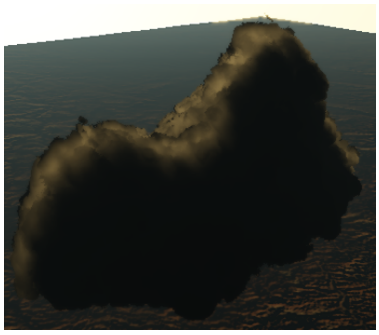
(b) Sky illumination, bringing soft blue-grey tints.



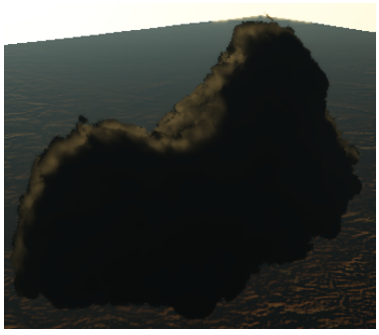
(c) Sun illumination, orders $n = [16, \infty]$. Note the yellow-orange soft tints.



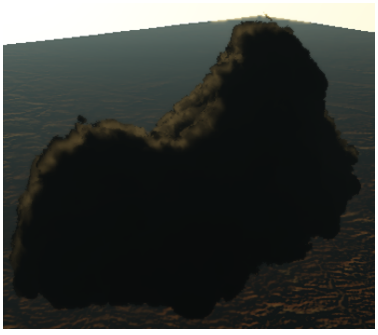
(d) Sun illumination, orders $n = [10, 15]$.



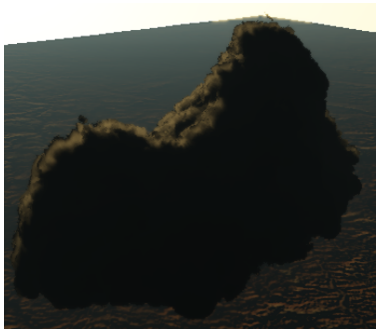
(e) Sun illumination, orders $n = [9, 14]$.



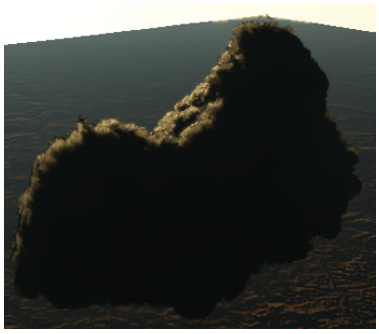
(f) Sun illumination, orders $n = [6, 8]$.



(g) Sun illumination, orders $n = [4, 5]$.



(h) Sun illumination, order $n = 3$.



(i) Sun illumination, order $n = 2$.

(j) Sun illumination, order $n = 1$.

Figure 7.11: *Contributions of each of the sources and orders of scattering resulting in the image of Figure 7.10. Note how the contrast increases for low orders.*

Figure 7.12: *A cumulus cloud at 2 FPS. Note the strong backscattering.*

Figure 7.13: *Close-up on a cumulus cloud at 2 FPS. The Hypertexture approach allows us to model very fine details. Dark pixels on the lit edges of the cloud are due to bugs in our implementation.*

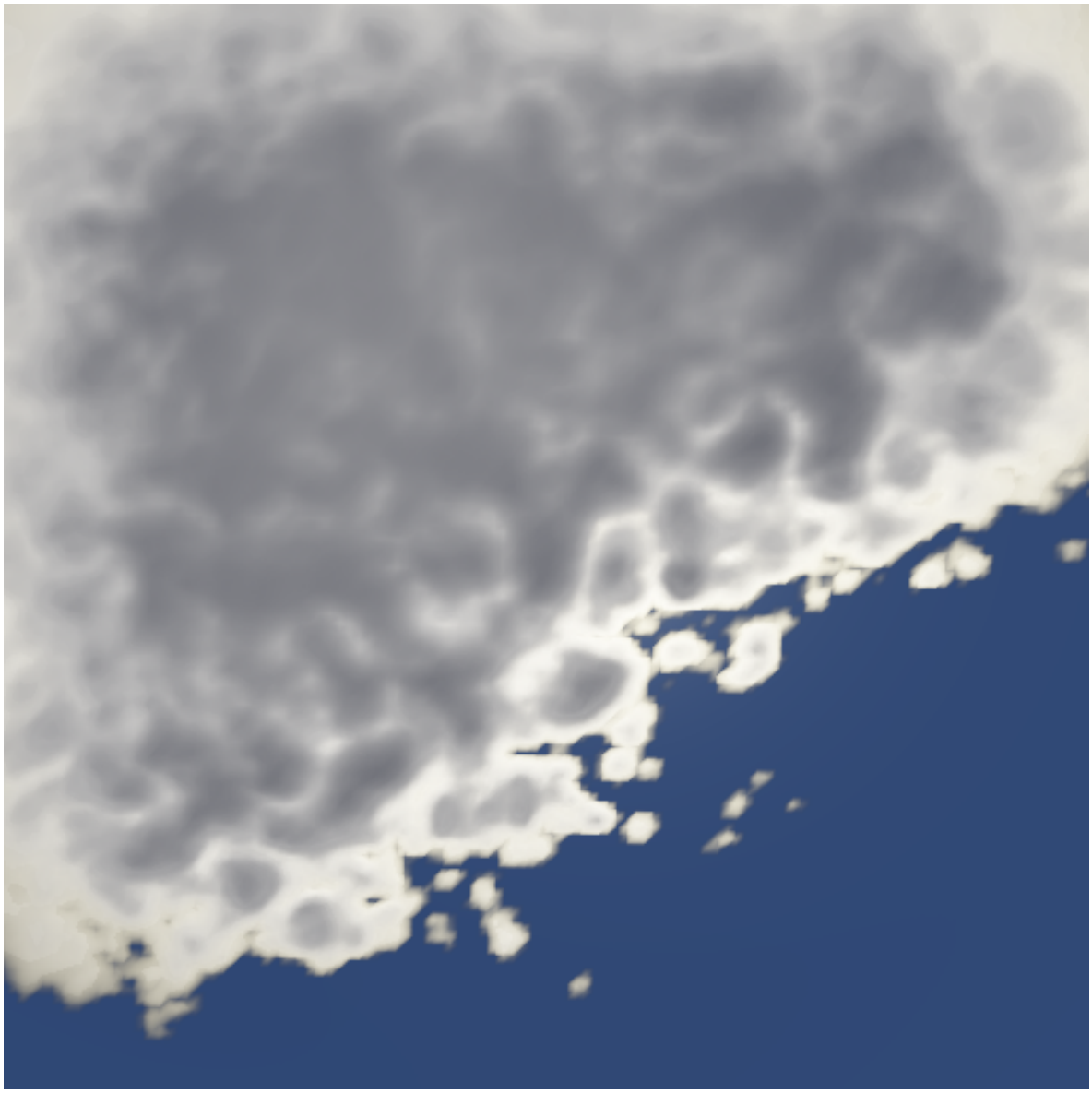
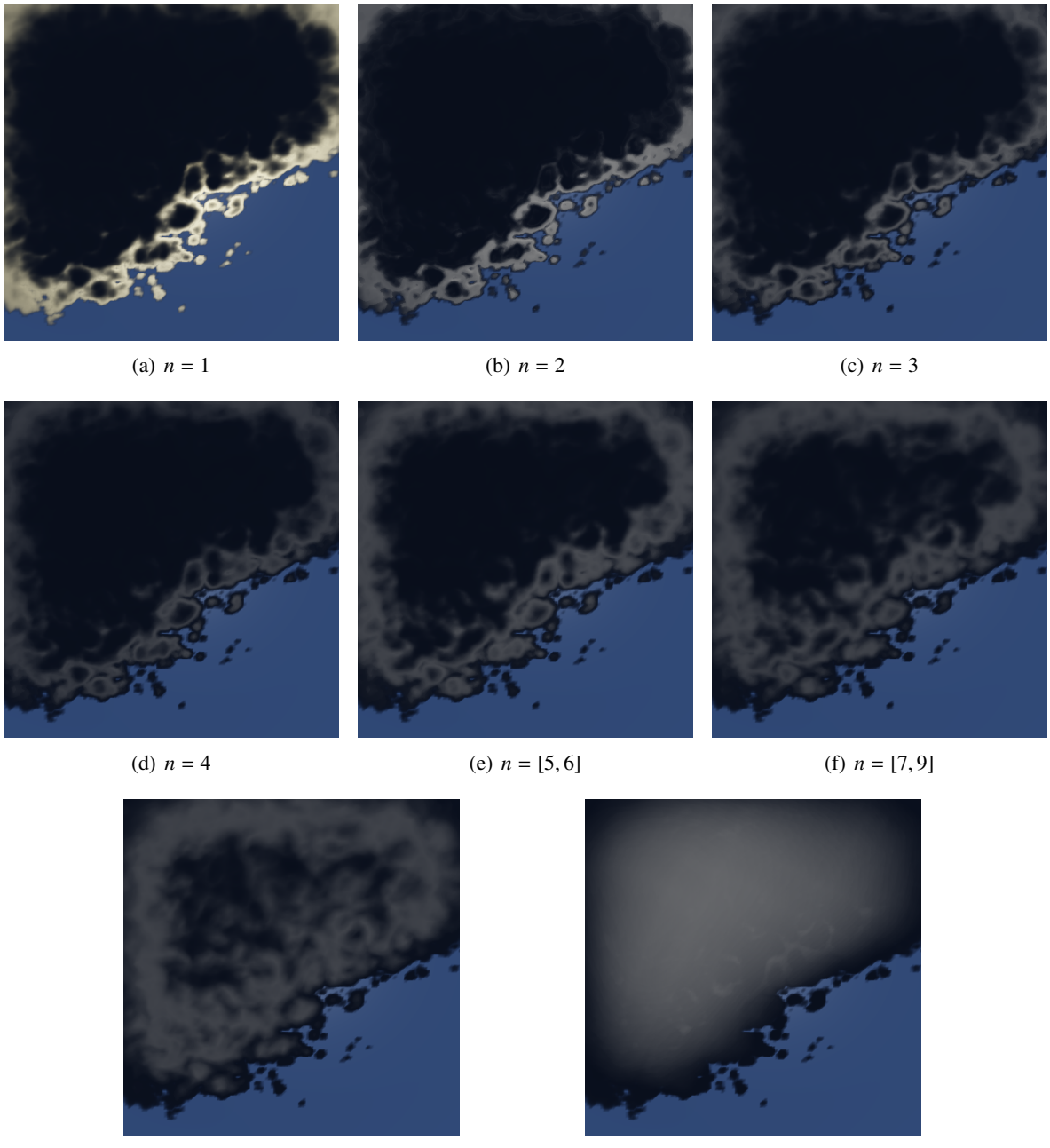


Figure 7.14: *A stratocumulus viewed from below at 10 FPS. Note contrast between forward scattering thin parts and diffusive thick parts. The shape is model similar to that of Chapter 6: no Hypertexture is used, it is totally homogeneous within the boundary $\mathcal{S}$ (compare with Figure 6.6). Figure 7.15 shows each the contributions of each of the orders of scattering resulting in this image.*



(a) $n = 1$ (b) $n = 2$ (c) $n = 3$

(d) $n = 4$ (e) $n = [5, 6]$ (f) $n = [7, 9]$

(g) $n = [10, 15]$ (h) $n = [16, \infty]$

Figure 7.15: *Contributions of each of the groups of orders of scattering for the sun illumination resulting in the image shown on Figure 7.14. Note how the low orders bring bright, contrasted, anisotropic illumination while high order bring soft, diffuse light. Also note that low orders bring light in thin parts while higher orders are stronger in thicker parts.*

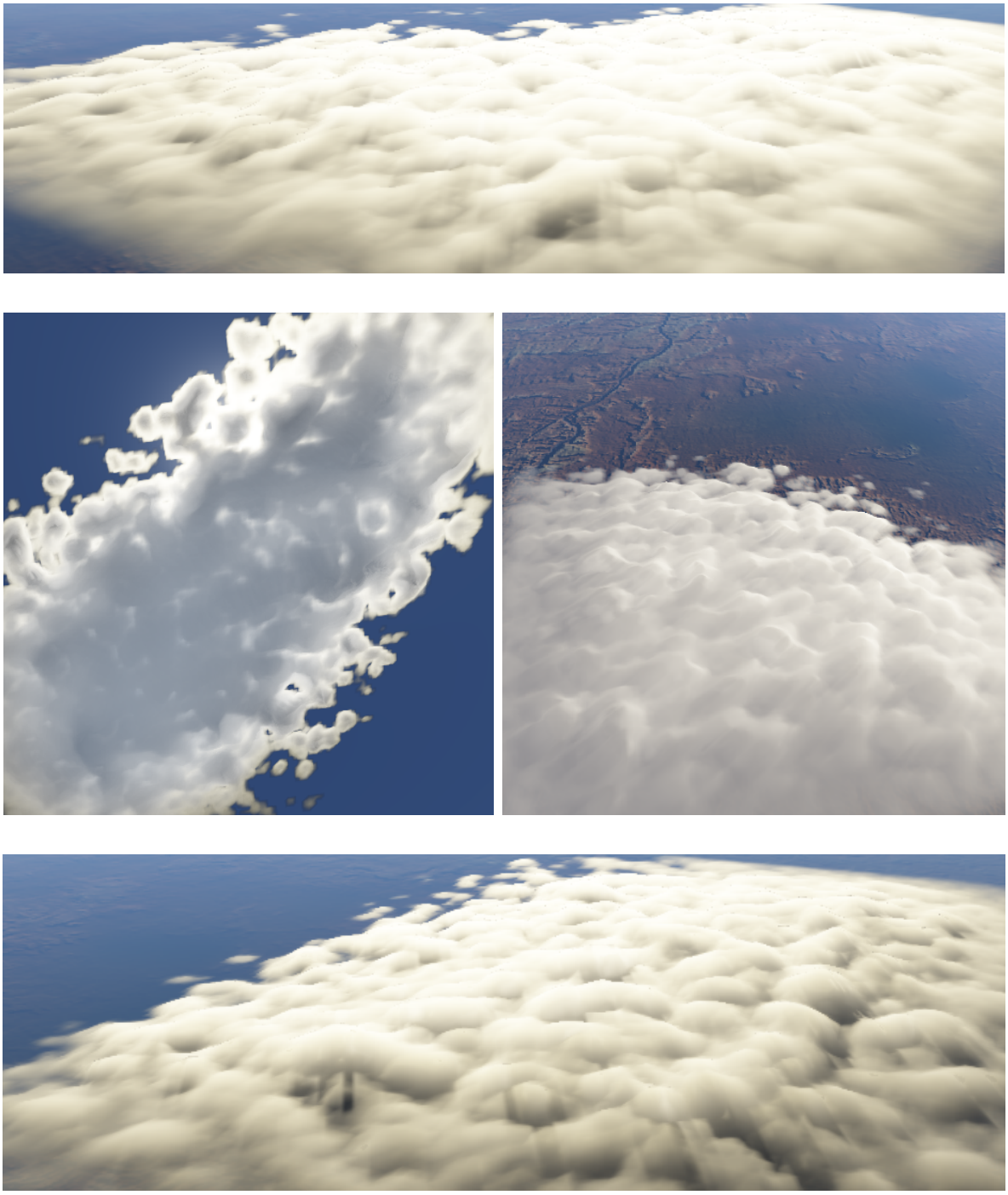
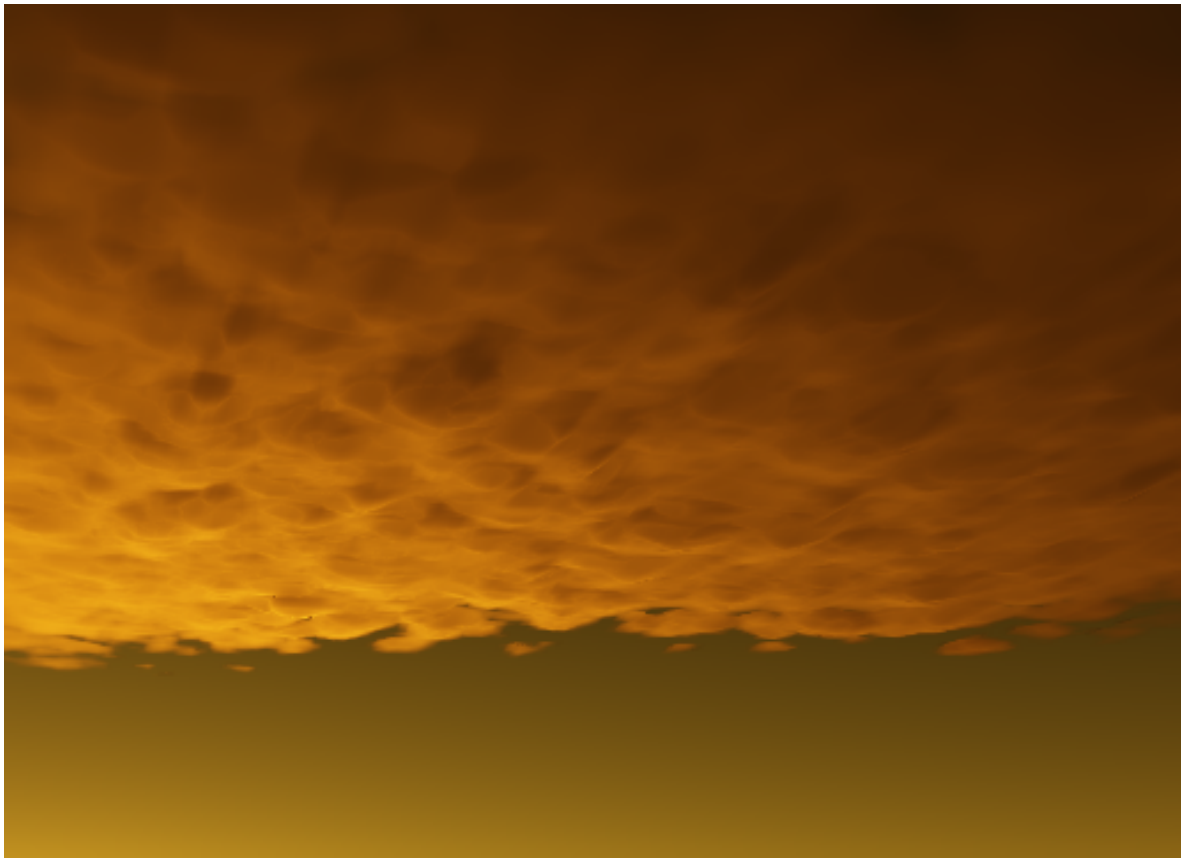
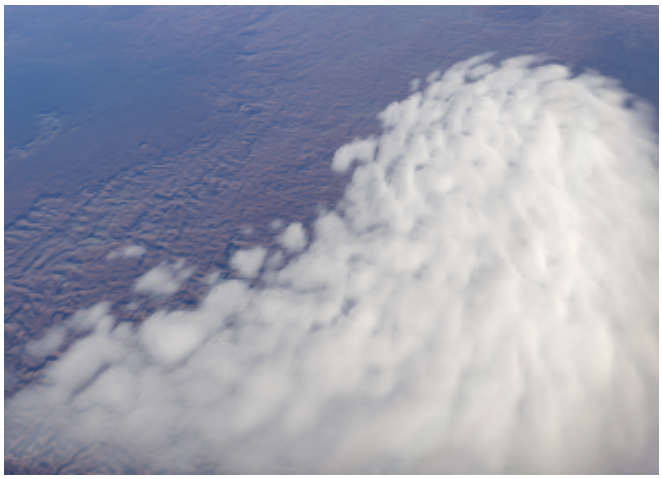


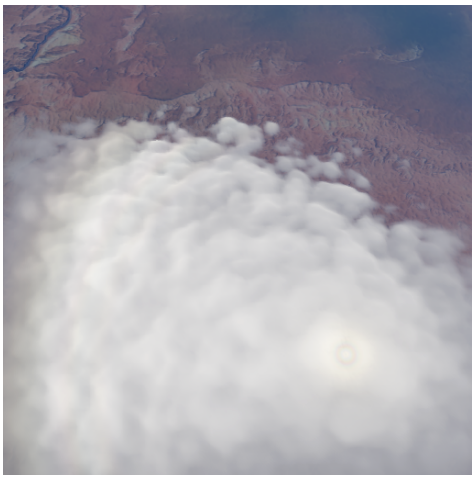
Figure 7.16: *Stratiform clouds rendered at 10 FPS. Note the pseudo-specular effect for grazing angles. The shape is model similar to that of Chapter 6: no Hypertexture is used, it is totally homogeneous within the boundary $\mathcal{S}$. Note how taking into account the spatial spreading of light avoids the "hard" look of our previous model (see Figure 6.5(b)).*



(a) At sunset.



(b)



(c) Note the glory (bottom right of the image) and fogbow (left side of the image).

Figure 7.17: *Stratiform clouds at 10FPS. The shape is model similar to that of Chapter 6: no Hypertexture is used, it is totally homogeneous within the boundary $\mathcal{S}$.*

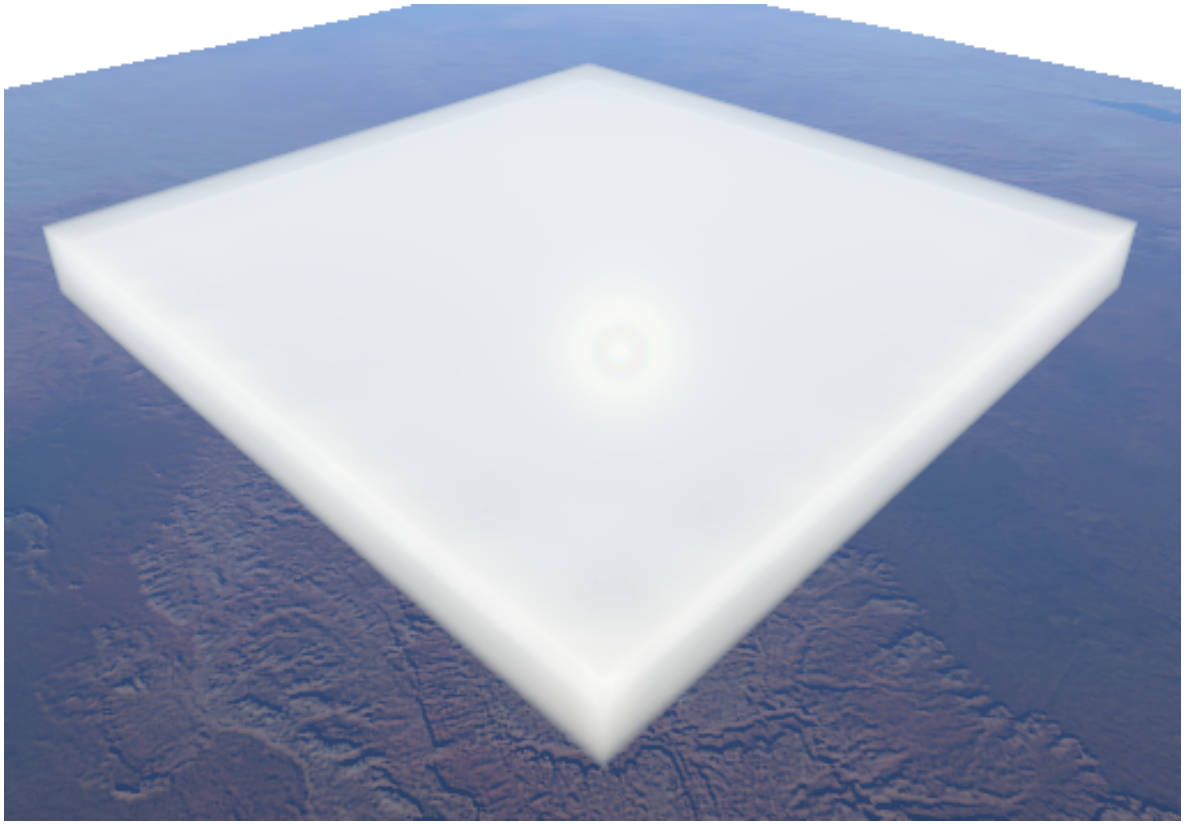


Figure 7.18: *A slab of cloud. The glory is visible. Amusingly, even such a sharp representation has a soft look, thanks to the spatial spreading of light.*

Figure 7.19: *Bunny cloud at 10FPS. In this model no Hypertexture is used. The cloud is considered homogeneous within the boundary mesh, which we jitter to add procedural noise to the shape. Note the brighter thinner areas such as the neck or the ears of the bunny.*

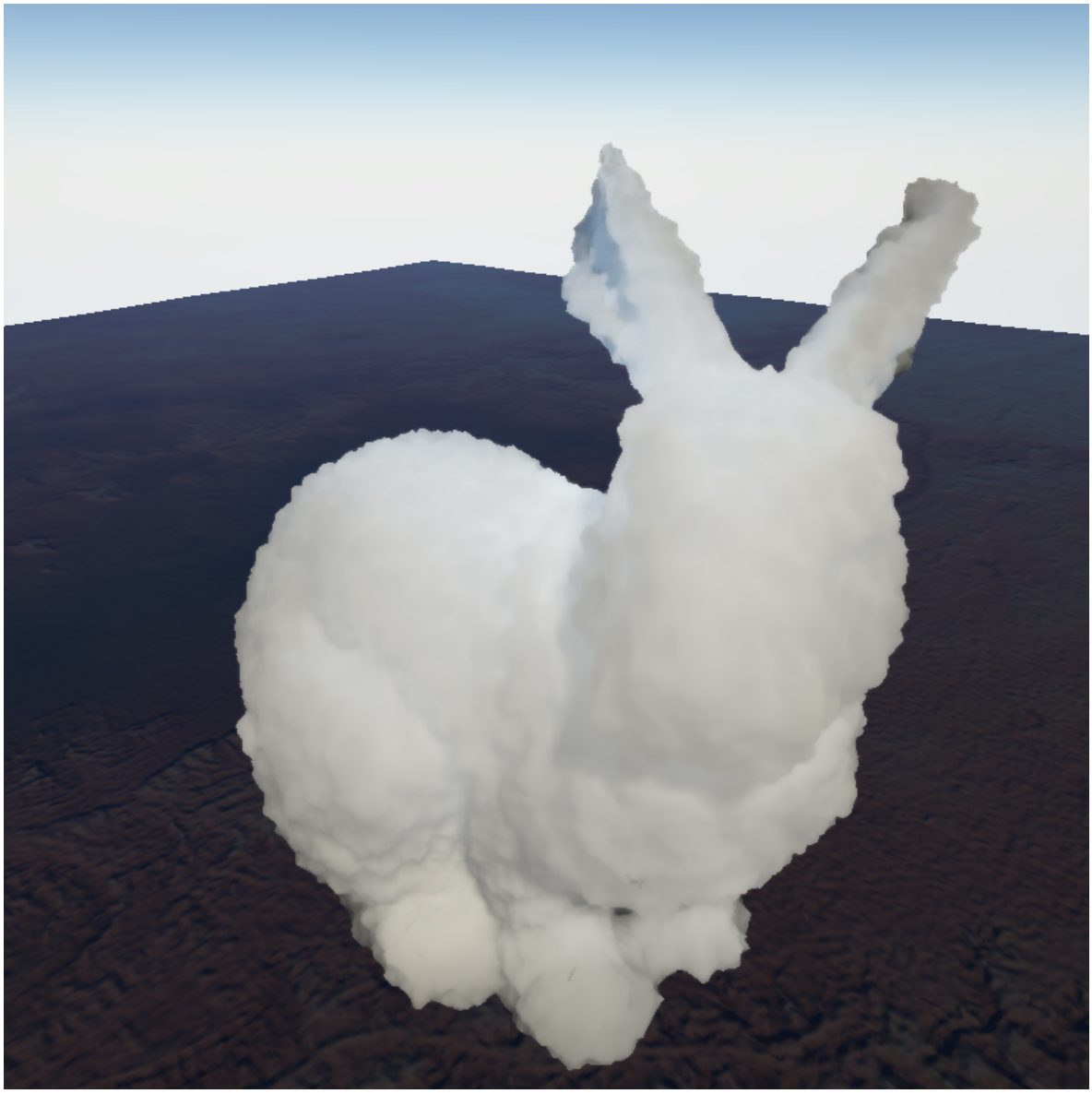


Figure 7.20: *Bunny cloud at 10FPS. In this model no Hypertexture is used. The cloud is considered homogeneous within the boundary mesh, which we jitter to add procedural noise to the shape. Note the strong backscattering.*

7.8 Discussion

7.8.1 Validity and convergence of our collector-finding algorithm

Our whole light transport model relies on our collector-finding algorithm (§7.4.3.3). It assumes that the collector exists, that it is unique, and that a fixed-point method is sufficient to find it. We discuss here how robust these assumptions are.

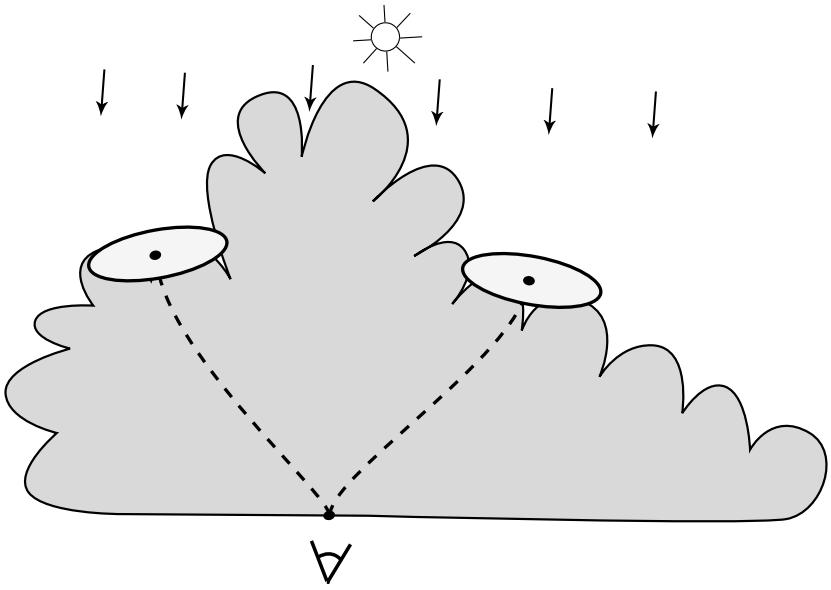


Figure 7.21: *Example of the issue of looking for only one collector. In this case, the collector is made of two disconnected areas. Our algorithm will only find one of them.*

7.8.1.1 Validity

Regarding the existence and unicity of the collector C, we can easily prove them. Indeed, by definition, all light paths have an entry point on the lit surface $\mathcal{I}$. And since these light paths exist, it is possible to define the unique area $C \in \mathcal{I}$ that encloses the entry point of the 95% most significant light paths. Thus, C exists and it is unique.

The approximation we make is that we assume that on an arbitrary cloud C is a simple disk (§7.4.3.1). Although it is certainly reasonable for a slab, for a real cloud it might be possible that C takes vastly different shapes, including shapes made of disconnected areas. Figure 7.21 shows an example of how such a case might arise. In this 2D example, we can see that the collector would be formed of two disconnected areas. If we imagine the 3D equivalent, chances are that the collector would take the form of a ring around the cloud tower.

In this kind of case, our algorithm naively assumes there is one disk-shaped collector and ends up at a one of the two locations. The solution to this issue is not straightforward. One can imagine searching for several disk-shaped collectors to recover all the portions of the ideal collector, but one would have to make sure that these disks do not overlap. This problem more generally corresponds to solving a function of multiple roots. Although there exist solutions for this type of problems, their application is not straightforward in the present case and would require detailed research. We leave this problem as open.

As a result, the validity of our approach strongly depends on whether real collectors in arbitrary clouds are disk-shaped or not. This could be verified by computing these real collectors using Monte-Carlo path tracing in the given cloud. However, time lacked us to perform such a validation.

As a consequence, we can only conclude by observing the realism of our results that this assumption does not seem this far away from reality.

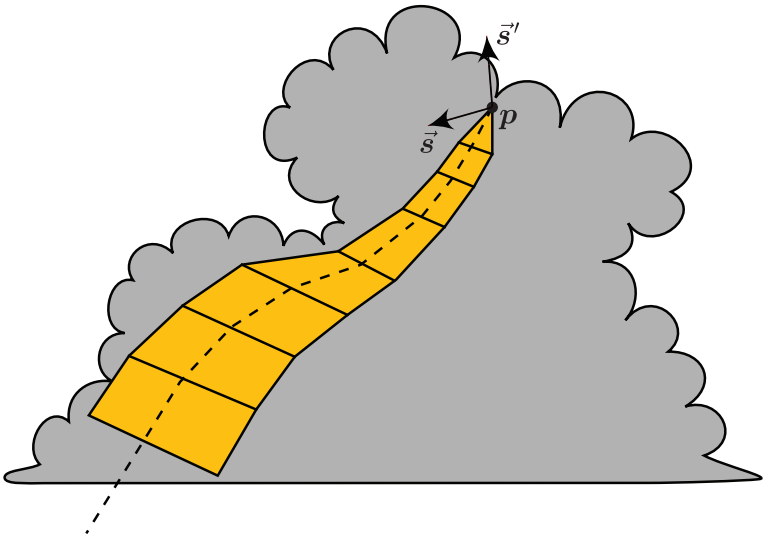


Figure 7.22: *Principle of the strips method. We integrate light transport along a strip located on the lit surface of the slab.*

One of the consequences of the issue shown on Figure 7.21 is that our algorithm may alternate between the two possible solutions for two neighboring viewpoints or two neighboring view directions. This may lead to spatial or temporal discontinuities and is probably the most visible effect of our approximation. Fortunately, we have observed that even when these cases arise, their consequences are visually negligible.

One of the reasons for this is that both potential collectors actually have similar contributions, thus the visual result of choosing one or the other is not much different. Note that this is quite inherent to the issue, since it is because they have similar that they both constitute equal candidates for our algorithm. Another reason for the low visual impact of this issue is that it usually does not happen for more than only one order, having a low influence on the total result.

7.8.1.2 Generalization

We can view our collector-finding approach in a more general way, as follows. It is actually a way to estimate the distribution D of entry points of light paths on the lit surface. In our collector-based approach we assume that this distribution has a strong peak and we look for the location of this peak. However, it is possible to imagine various different ways to estimate this distribution D.

Prior to developing our collector-based approach we have tried another method that we call "strips-based" approach. We sketch it here quickly.

This approach assumes that the distribution is strongest within a strip C (this strip can be seen as a band-shaped collector). This strip is assumed to be located along the intersection of $\mathcal{I}$ with the plane passing through $\boldsymbol{p}$ of normal $\vec{s} \wedge \vec{s}'$ (see Figure 7.22). We sample this band by considering k elements of strips, and we compute the light transport from each element of strip using a model based off of our study that we do not describe here. In terms of computations, this method is no more intensive than doing k iterations of our collector-finding model. Figures 7.23 – 7.24 shows some results of this method.

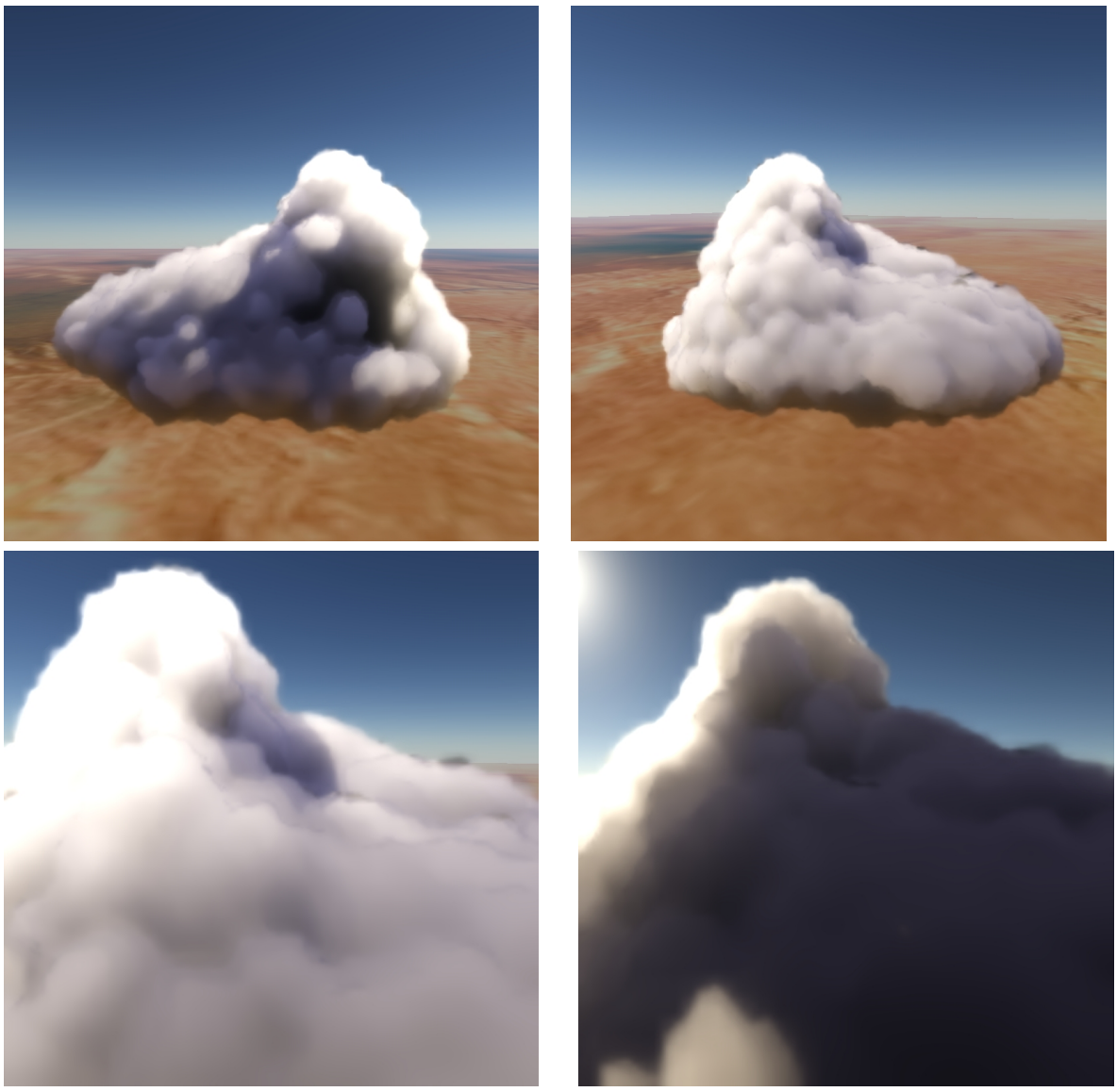


Figure 7.23: *Results using the strips method. The shape model is a homogeneous cloud bounded by a mesh.*

Figure 7.24: *Results using the strips method. The shape model is a homogeneous cloud bounded by a mesh.*

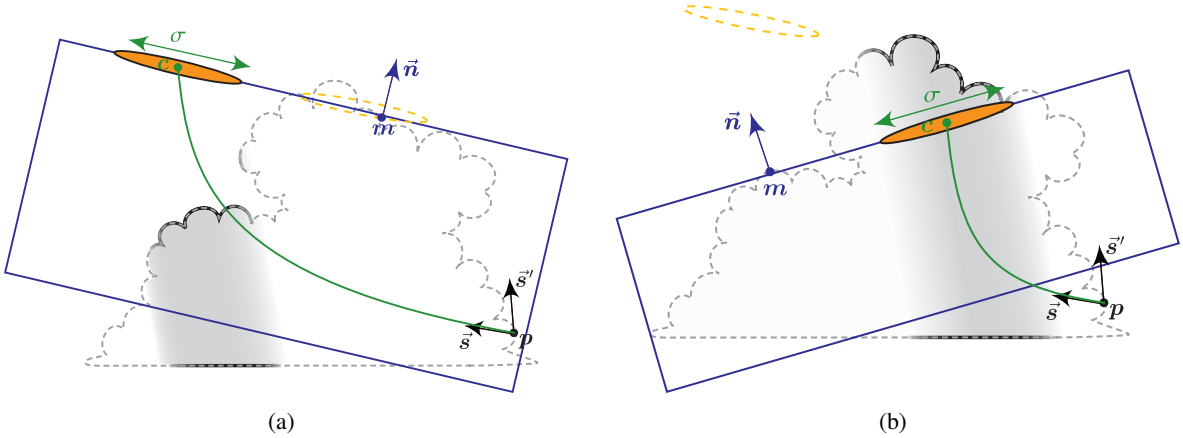


Figure 7.25: *Possible periodic points can arise near discontinuities if we do not constrain the displacement of the collector between each iteration. In this example, the fitted slab in* (a) *gives a collector which results in the fitted slab in* (b)*, which gives the collector in that results in the fitted slab in* (a)*, etc.*

7.8.1.3 Convergence

Assuming that looking for a disk-shaped collector is valid enough, we discuss here the robustness and convergence of our fixed-point algorithm.

The robustness can be defined as follows. If a small local modification of the surface of the cloud makes our algorithm give very different results, it is not robust. Thus, to ensure good robustness, we choose the initial collector size σ_0 as large as possible. Ideally, this means that at the first step the collector spans the whole lit surface. Then in subsequent steps, our algorithm will likely decrease the collector size and thus "refine" the cloud surface at the most probable collector location.

Concerning the initial collector location, we choose to use the viewpoint location, *i.e.,* $\boldsymbol{c}_0 = \boldsymbol{p}$. Indeed, we know that in the asymptotical case of an extremely dense medium, the entry point and the exit point of light paths are at the same location (a solid non-subsurface-scattering surface can be seen as a surface where sub-surface scattering happens at the molecular scale). Moreover, from our study we have seen that the most probable entry point is never very far away from $\boldsymbol{p}$. Thus, by choosing $\boldsymbol{c}_0 = \boldsymbol{p}$ we are ensuring that we start near a possible solution.

According to our experiments, these initial values have shown to be robust choices. Regarding convergence, we have observed that our algorithm is sometimes trapped in periodic points, *i.e.,* it loops between various collector locations without converging. This is often due to discontinuities in $\mathcal{I}$, as shown on Figure 7.25. These discontinuities make our algorithm take steps too large between each iteration. We overcome this problem by applying a constraint on the variation of $\boldsymbol{c}$ and σ between each iteration. Experimentally, we found the following constraints to work well:

$$\|\boldsymbol{c}_{i+1} - \boldsymbol{c}_i\| < \sigma_i/2, \tag{7.38}$$

$$|\sigma_{i+1} - \sigma_i| < \sigma_i/2. \tag{7.39}$$

Note that if the constraints are made too small, it will make the algorithm converge more slowly than without constraints, which is undesirable. We found that the constraints above were a good compromise.

Using the criterion

$$\|\boldsymbol{c}_{i+1} - \boldsymbol{c}_i\| \quad < \quad \sigma_i/10 \tag{7.40}$$

as convergence criterion, we found that our algorithm converges in almost all cases in less than 10 iterations. In cases where it does not converge, the result is close enough to the ideal solution or the error is small enough so as not to have visual effect.

7.8.2 Implementation issues

Some of the limitations and issues of our results are bound on the implementation rather than on the method itself. Relying on depth maps inherently yields discretization issues, like other image-space subsurface scattering techniques [DS03]. Like these methods, it is also biasing the results when the clouds are non-convex (see Figure 7.26). These issues can be avoided by using a different implementation or using additional techniques, such as depth peeling to treat non-convex shapes [DS03] and deep shadow maps [LV00] to leverage discretization issues and give better results for the single scattering.

7.8.3 Speed and quality

The rendering framerate for our multiple scattering method is 10FPS. We believe it can be increased through a better implementation. Moreover, the speed can be further increased at the expense of quality in various manners. For example, one can decrease the maximum number of iterations of our collector-finding algorithm, simplify the computation of sky and ground contributions, or change the grouping of scattering orders to yield less groups. All these ideas provide handles on which a developer can act to adjust the ratio between quality and speed. Another example is to simply not recompute $\mathcal{I}$, $\mathcal{I}'$, $\vec{\boldsymbol{n}}(\boldsymbol{p})$, etc. at every frame if the shape or the light sources are not animated.

Our rendering framerate drops to 2 – 3 FPS when computing transparency through ray marching. However, our ray marcher is highly unoptimized and previous implementations [HKO07] have shown that it is perfectly possible to reach realtime framerates on current hardware. Moreover, we could gain significant speed by simply storing the Hypertexture noise instead of evaluating it on the fly.

Thus, we consider our method to achieve interactive framerates and to be have the potential to yield realtime framerates.

Note that although we have shown that our approach works for one cloud and a limited cloud layer, it has yet to be demonstrated that it can work for a full cloudy sky. It is likely that this task requires management of level of detail. As an example, it is widely known in CG that the depth maps we use to represent the surfaces (§7.6.2) have limitations on the range of distances they can handle because of discretization and quantization issues [SD02].

7.8.4 Potential applications

Our method allows for the realtime rendering of arbitrary clouds for any given viewpoint, view angle and light direction, and allows the animation of clouds. This makes it best suited for applications that have a high demand on the quality of clouds and need the viewer to be able to see them from all angles, such as flight simulators and airplane video games.

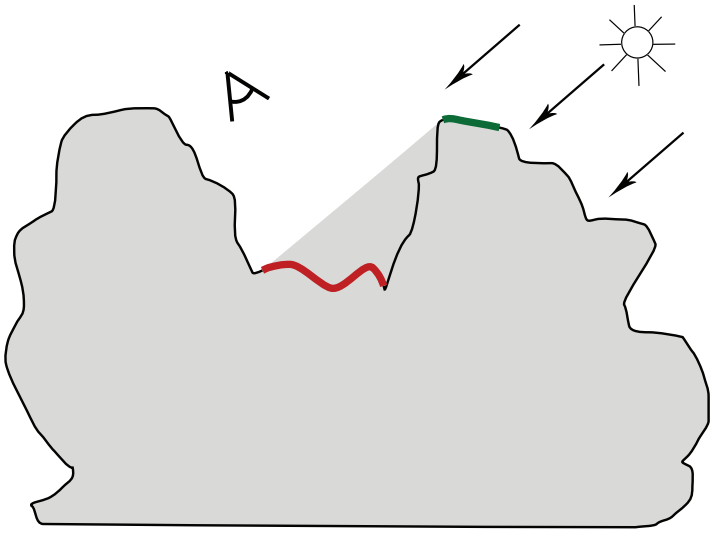


Figure 7.26: *Issue of relying on depth maps to represent the lit surface of the cloud in the case of non-convex clouds. The red surface is assumed to be inside the cloud and its luminance is underestimated. However this error is of low visual importance. Conversely, our method considers there is more medium below the green surface than there actually is. This diminishes the dark edges effect at the green surface.*

However, this is not the only possible application. This approach is a solution for the subsurface scattering of light in a bounded semi-homogeneous object. Thus, it can be applied to any other kind of subsurface scattering problem [JMLH01] such as the rendering of marble, skin, fruits, milk, etc. Moreover, since these objects are generally less anisotropic than cloud droplets, this approach can probably be simplified to obtain greater speeds.

Finally, we believe that more quality-oriented, slower versions can be implemented for the computation of offline visual effects.

7.9 Summary of Chapter 7

In this chapter, we have proposed four new complementary approaches for the realtime realistic rendering of clouds. These approaches are:

- A new way of considering light transport in clouds based on searching for the area containing the entry points of the most probable light paths (the *collector area*) on the lit surface of the cloud for each order of scattering (§7.4.3.1).
- An algorithm to find this collector area, based on our study of light transport (Chapter 4), that assumes this collector is disk-shaped (§7.4.3.5).
- An interactive implementation of this algorithm on GPU (§7.6).
- A new model for the representation of the shape of the clouds that is efficient, high-resolution and GPU-friendly (§7.2.1).

The combination of these four approaches results in the images shown in §7.7.

We have discussed the advantages and limitations of each of these approaches (§7.8). In particular, we believe there may be better algorithm to find the collector area and that the implementation of the

algorithm we propose can be greatly improved. Moreover, we have seen that searching for the collectors is a particular case of estimating the density distribution D of the entry points of light paths on the lit cloud surface. We have shown that there are other possible ways of estimating this distribution.

We have shown that the rendering model presented in this chapter can be applied to shape models other than the one we proposed (§7.2). Since it relies on defining an isosurface on the cloud it can be used on various other models, such as our heightfield representation of Chapter 6 or the models of [DKY+00, ES00, HBSL03, REK+04]. Conversely, our new shape representation (§7.2) may be used with previous rendering approaches.

Finally, we believe that our approach is not restricted to clouds and can be applied on other objects displaying scattering behavior in a bounded volume, such as all the types of objects rendered by classical subsurface scattering approaches.

As a consequence, we believe that the contributions of this chapter open an interesting range of research possibilities and improvements.

Dans ce chapitre, nous avons présenté quatre nouvelles approches complémentaires pour le rendu réaliste de nuages en temps réel. Ces approches sont:

- *Une nouvelle façon de considérer le transport de la lumière dans les nuages, consistant à rechercher l'aire contenant les points d'entrée des chemins lumineux les plus probable (l'*aire collectrice*) sur la surface illuminée du nuage pour chaque ordre de diffusion (§7.4.3.1).*
- *Un algorithme permettant de trouver cette aire collectrice, basé sur notre étude du transport lumineux (chapitre 4), qui assume que ce collecteur a la forme d'un disque (§7.4.3.5).*
- *Une implantation de cet algorithme sur GPU ayant une vitesse interactive (§7.6).*
- *Un nouveau modèle de représentation de la forme des nuages à la fois efficace, haute-résolution et adapté au GPU (§7.2.1).*

Le résultat de la combinaison de ces quatre approche est visible sur les images en §7.7.

Nous avons présenté les avantages et limites de chacune de ces approches (§7.8). En particulier, nous pensons qu'il existe de meilleurs algorithmes de recherche du collecteur et que l'implantation de notre algorithme peut être grandement améliorée. De plus, nous avons montré que l'opération de recherche du collecteur peut être généralisée comme étant un moyen d'évaluer la distribution de densité D des points d'entrées des chemins lumineux sur la surface illuminée du nuage. Nous avons montré qu'il y a d'autre façons possibles d'évaluer cette distribution.

Nous avons montré que le modèle de rendu présenté dans ce chapitre peut être appliqué à des représentations de forme différentes de la notre (§7.2). Étant donné qu'il repose sur la définition d'une isosurface sur le nuage, il peut être utilisé sur une variété d'autres modèles, tels que notre représentation à base de champ de hauteur (chapitre 6) ou les modèles de [DKY+00, ES00, HBSL03, REK+04]. Inversement, notre nouvelle représentation de forme (§7.2) pourrait être utilisée avec d'autres approches de rendu.

*Pour terminer, nous pensons que notre approche n'est pas restreinte aux nuages uniquement et peut être appliquée à d'autres objets présentant des propriétés diffusives dans un volume restreint, tels que ceux habituellement rendu via des techniques de diffusion sous-surfacique (*subsurface scattering*) classiques.*

En conséquence, nous pensons que les contributions de ce chapitre ouvrent un éventail de possibilité de recherches et d'amélioration intéressant.

Part IV

Conclusions

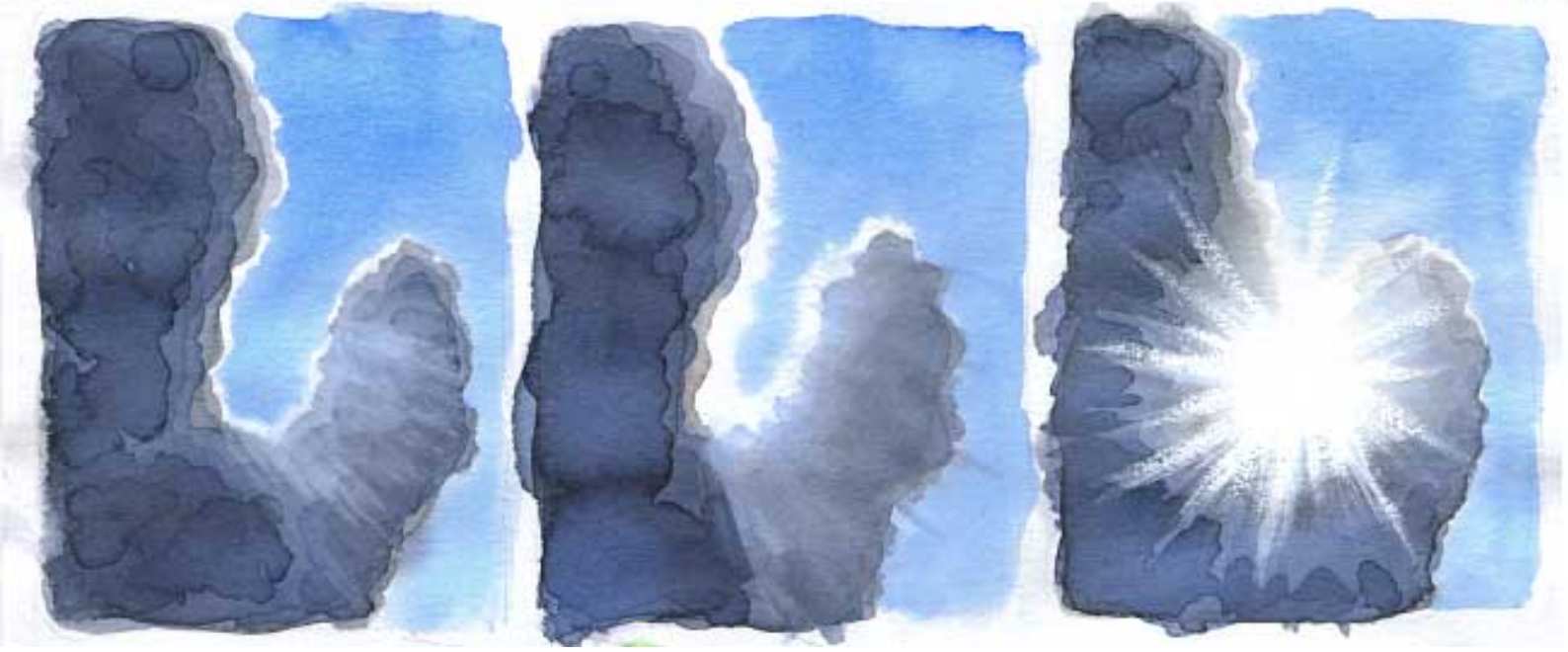

Féchié...
Ce qui m'énerve, c'est qu'à quelques kilomètres, il fait plein soleil et pas ici !
En fait, la moitié du ciel est dégagée et moi chuis sous la mauvaise moitié...
Mais avec un peu de bol, c'est juste un gros nuage qui va se barrer.

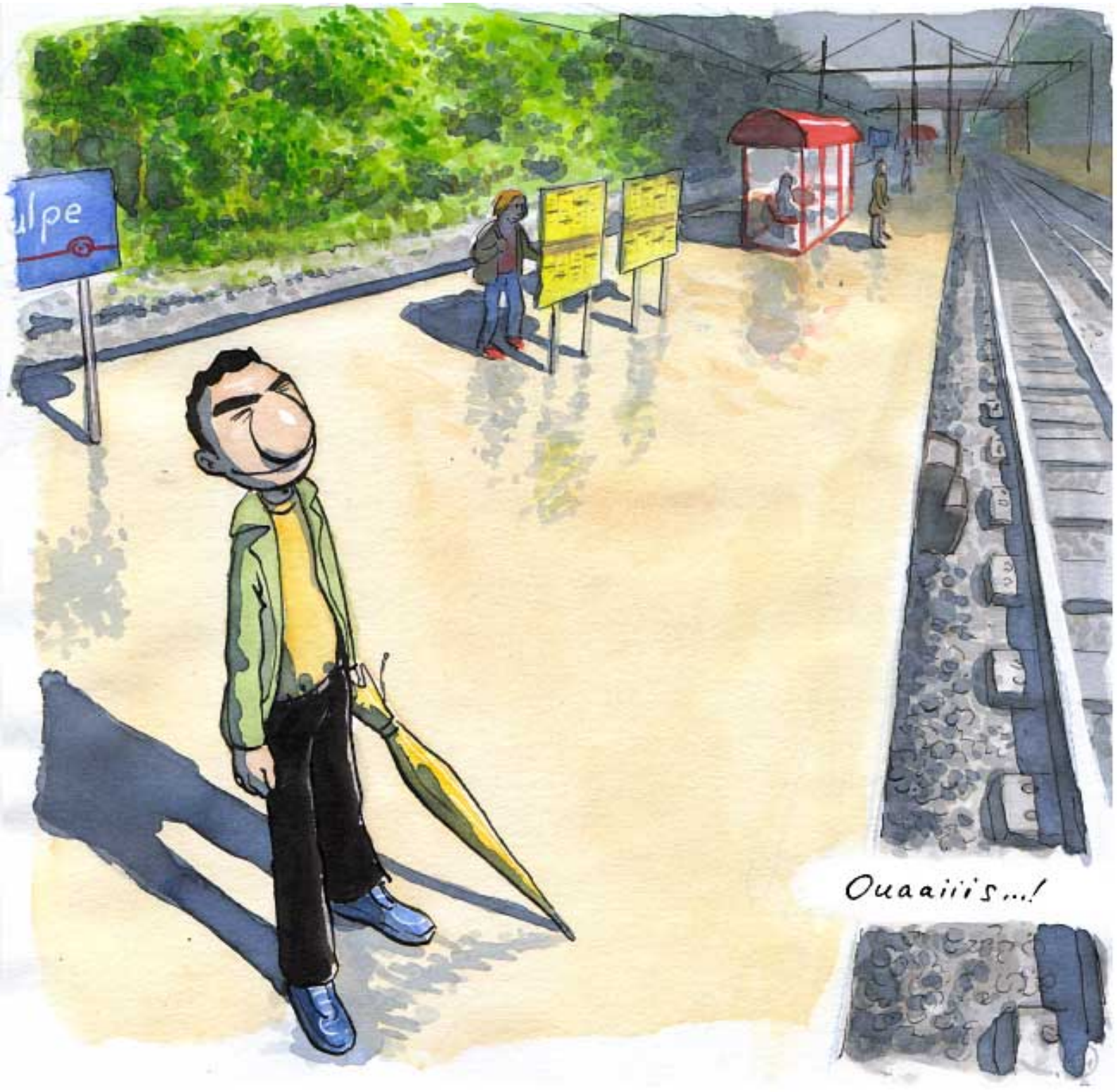

Ah ben ouais.
La Hulpe

C'est pas juste une couche de gris tartinée sur le ciel...

C'est un blob monstrueux, une immense guimauve cendrée.

Et elle tourne lentement...




lpe
Ouaaiii's...!

Chapter 8

Conclusions

Contents

8.1 Summary of results

In this thesis we have proposed two new models (Part III) for the realtime, realistic rendering of clouds. These models are based on a study of light transport in a slab of clouds (Part II). This study has helped us understanding the behavior of light in clouds (Chapter 4) and find efficient phenomenological models of light transport in clouds (Chapter 5). We have obtained realistic results by observing real clouds and seeking the visual features that characterize them (Chapter 2) in order to reproduce these features in our models.

These visual features are a detailed shape, puffy or wispy edges, glory, fogbow, view-dependant lighting effects (silver lining, dark edges, pseudo-specular effect), bright creases, high contrast, sky and ground illumination, etc.

Our first model (Chapter 6) is simple, efficient and designed for stratiform clouds. It relies on the representation of the shape of the clouds through a height field, which is particularly efficient. It accounts for the anisotropic light transport of light clouds by treating the first two orders of scattering and the higher orders separately. It accounts for the sky and ground illumination as well as the inter-reflections between the ground and the clouds. We have proposed a GPU implementation of this model yielding realtime speeds and reproducing most of the visual features of clouds. The main limitation of this model is that it is limited to stratiform clouds.

Our second model (Chapter 7) is designed to handle any cloud shape, and more specifically cumulus clouds. It relies on the subsurface scattering paradigm. We propose a new approach to find significant light paths in clouds via searching for their *collector area* which comprises their entry point on the lit surface of the cloud. In addition to this new formulation we have proposed a new, GPU-based algorithm to find these collectors and compute their contributions in interactive time. Similarly to our previous model, we account for the varying anisotropy of light transport by treating separately light paths of different orders. We have applied this method on a new representation of the shape of clouds that allows us to reproduce the fine details of clouds such as the wisps, sharp edges and small puffs. This method also reproduces all the visual features of real clouds.

Both of our rendering model support the animation of the cloud shape, view directions and light directions.

Dans cette thèse, nous avons proposé deux nouveaux modèles (partie III) de rendu réaliste de nuages en temps réel. Ces modèles se basent sur une étude du transport lumineux dans des dalles de nuages (partie II). Cette étude nous a permis de comprendre le comportement de la lumière dans les nuages (chapitre 4) et de trouver des modèles du transport lumineux dans les nuages (chapter 5). Nous avons obtenu des résultats réalistes en observant les nuages réels et en identifiant leurs caractéristiques visuelles (chapitre 2) afin de reproduire ces caractéristiques dans nos modèles.

*Ces caractéristiques consistent en une forme détaillée, des silhouettes gonflées ou torturées, la gloire, l'arc-en-nuage, des effets dépendant du point de vue (*silver lining*, bords sombres, effet pseudo-spéculaire), des creux lumineux, un contraste fort, un éclairement du ciel et du sol non négligeable, etc.*

Notre premier modèle (chapitre 6) est simple, efficace et conçu pour les nuages stratiformes. Il repose sur une représentation de la forme des nuages via un champ de hauteur, ce qui est particulièrement optimisé. Il reproduit le transport anisotrope de la lumière par les nuages en traitant séparément les deux premiers ordres de diffusion et les ordres supérieurs. Il reproduit l'éclairement du ciel et du sol

ainsi que les inter-réflexions entre les nuages et le sol. Nous avons proposé une implantation de ce modèle sur GPU aboutissant à des vitesses de rendu temps-réel et reproduisant la plupart des caractéristiques visuelles des nuages. La principale limite de ce modèle est qu'il est restreint aux nuages stratiformes.

Notre second modèle (chapitre 7) est conçu pour fonctionner sur toutes les formes de nuages, et plus spécifiquement du type cumuliforme. Il repose sur le principe de la diffusion sous-surfacique. Nous proposons une nouvelle façon de trouver les chemins lumineux significatifs dans les nuages en recherchant leur aire collectrice *qui comprend leurs points d'entrée sur la surface éclairée du nuage. En plus de cette nouvelle formulation, nous avons proposé un nouvel algorithme permettant de trouver ces aires collectrices et de calculer leurs contributions en temps interactif sur GPU. De même qu'avec notre modèle précédent, nous tenons compte de l'anisotropie du transport lumineux en traitant séparément les chemins lumineux de différents ordres. Nous avons appliqué cette méthode sur une nouvelle représentation de la forme des nuages qui nous permet de reproduire les détails fins des nuages tels que les barbules, les bords nets et les volutes. Cette méthode est capable de reproduire toutes les caractéristiques visuelles des vrais nuages.*

Nos deux modèles de rendu sont compatibles avec l'animation des nuages, du point de vue et de l'éclairage.

8.2 Future research

The method that we have proposed to render cumulus clouds has the potential of being applied on other materials having scattering properties, in particular it can apply to all other objects addressed by subsurface scattering techniques [JMLH01].

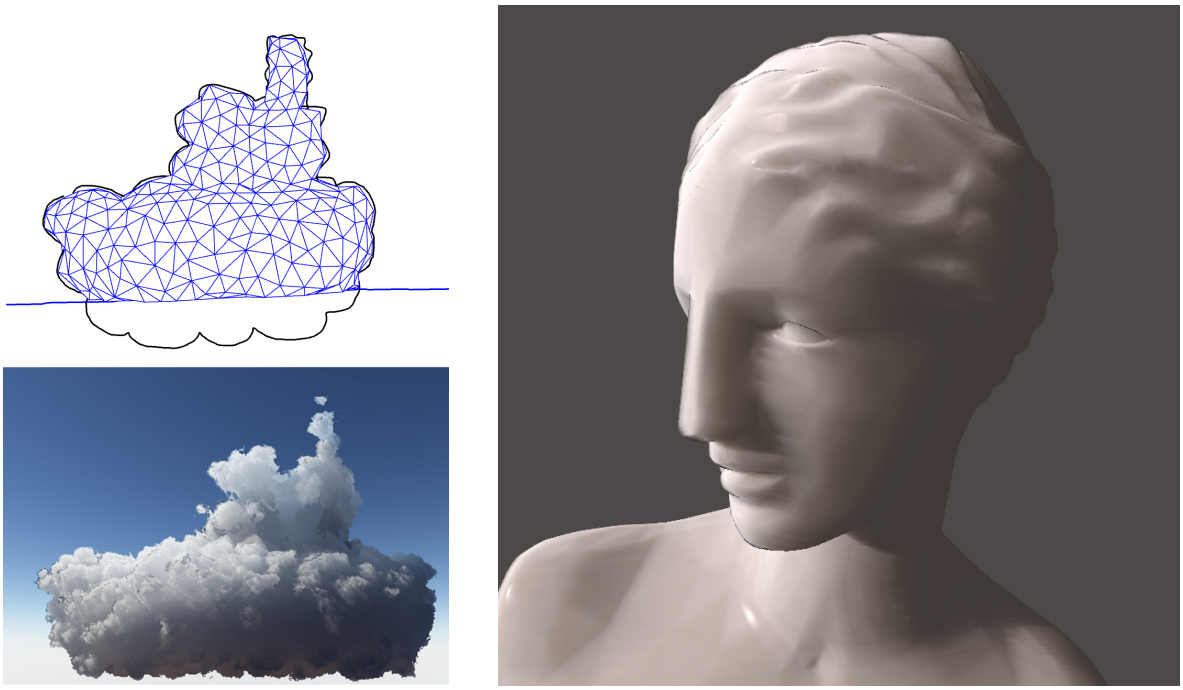
Other methods can be imagined to find the collector in a faster and more robust way. Other models for the computation of the exitance, collector position and size, and BSDF in a slab may be derived from the results of our study of light transport. In particular, it would be interesting to compare our BSDF model with existing BRDF models [HTSG91] and with other radiative transfer approaches [Cha60].

There is still the need for better shape and animation models of clouds. Our rendering method can be applied on other shape models than the ones we use in this thesis.

We believe this approach can be used for other applications, such as the ultra-realistic offline rendering of clouds for visual effects, or for inverse rendering (*i.e.,* retrieving the shape of a cloud from its appearance).

Still, some of the visual features of clouds are not fully reproduced by our methods and need further research. We did not address the inter-reflections between clouds, and between different parts of clouds which we believe to be the cause of the bright creases of clouds. The treatment of non-convex shapes yields plausible results in self-shadowed parts of clouds but they are not physically realistic. We did not address the rendering of ice clouds, which are made of non-isotropic particles. We believe it is possible to apply our method on this type of clouds by simply accounting for the particular effects of ice clouds (*e.g.,* halos, parhelia) through another phase function. The realistic rendering of a full sky would require better models for the rendering of the atmosphere and of other atmospheric phenomena such as rain and snow.

Finally, this work needs to be extended to take into account inhomogeneous clouds.



(a) Result of our cloud sketching method.

(b) Result of our realtime subsurface scattering method.

Figure 8.1: *Some side results.*

8.3 Side studies

Along the course of this thesis we have made a few other contributions to the state of research that we do not consider part of this thesis. We list them quickly here.

8.3.1 Subsurface scattering

While studying existing subsurface scattering models, we have proposed a new GPU implementation [BBNM07] of Jensen's subsurface scattering approach using the dipole method [JMLH01]. Our approach is very similar to that of Dachsbacher and Stamminger [DS03], which we didn't know of at the time of writing. The main difference between our method and all other realtime subsurface scattering methods is that we *do* take into account single scattering (using the approach of §7.4.2). Our approach is unpublished for the moment (see Figure 8.1(b)).

8.3.2 Cloud shape models

Although we have focused on rendering and not shape modeling in this thesis, we had to find decent ways of representing the shapes of clouds since we were not satisfied by existing approaches. As a result, we have made a few contributions on the modeling and representation of the shape of clouds.

Prior to starting this thesis we have made an attempt at creating cloud shapes procedurally [BN04]. However this method lacks user control and relies on the expensive triangulation of an implicit surface, which is not practical.

For our stratiform clouds model we chose to represent the clouds via height fields (§6.2), which gave us the ability of generating high resolution clouds with a very low memory footprint.

During our first experiments on cumuliform clouds, we asked a CG artist to create a cloud shape with a traditional modeling tool using spherical primitives as in [BN04]. After hours of work this resulted in a highly tesselated cloud mesh. To add details to this mesh we displaced its vertices using procedural noise. It is this shape that we used to test our strips model (§7.8.1.2). It can be seen on Figures 7.23. It is also a coarsened version of this mesh that we use as the boundary $\mathcal{S}$ of our improved shape representation in §7.2.1. The artist also attempted to produce an animated version of this model, but the results are currently not satisfying.

The time spent on the modeling of this one single cloud clearly show the lack of a practical, user-friendly modeling tools for clouds, not to mention animation. In the end of this thesis we have proposed a tool for the rapid sketching of a cloud shape [WBC08]. The result of this tool is a mesh that can be directly used by our rendering method of Chapter 7 and is clearly more practical that a conventional modeling package. Figures 8.1(a) show some examples of these results.

Finally, the new shape representation of §7.2.1 results from the work of a Master student [CN07], now starting a PhD in our team, whose research goal is to perform the realtime rendering of volumes at extremely high resolution. This representation came very handy at a point where a simple mesh or a coarse volumetric grid could not do justice to our rendering approach.

8.3.3 Triangulation of implicit surfaces

Since our procedural cloud modeling method [BN04] was relying on the triangulation of an implicit surface, we did some side research on this subject with a fellow PhD student, which has lead to a publication on the subject [BN07] and a couple of reports [NB06, RBNH07].

8.4 Reflections

This sections lists a few of the unexpected things that happened during the course of this thesis.

The goal of this thesis was of course at start much broader. As an example, the study done on a slab in Chapter 4 was only the first one planned among others. Obviously it took more time than expected, but more importantly it resulted in much more things to observe than we could think. It is quite interesting to see how much information such a simple model can give for the understanding of light transport, and finally for the creation of a rendering method designed for arbitrary cloud shapes.

However, the part that probably took most of the time of this thesis was to find the phenomenological models of §5.3 and §5.4. A BSSSDF is such a high-dimensional function that it was very difficult to find a way to simplify it and it took us months to finally use the behavior of C against z' as a core model. This was actually not very intuitive.

On the other hand, we have sometimes spent unnecessary time reinventing the wheel. As an example, we did not know of the doubling-adding method before using it, and neither did the reviewers of [BNL06]. We discovered its existence at the time we presented our paper to the Eurographics Workshop on Natural Phenomena. Similarly, we did not know of the existence of the chopped-peak model and found it randomly much later while looking for other potentially useful methods we might had overlooked [Len85].

This made us think that there could actually be existing model for the order-dependent BSDF of a slab of cloud that we would not know of. Unfortunately, we did not find any, resulting in the lengthy process of finding our phenomenological model of §5.4. We think there are probably more accurate models existing (or to be found) than the empirical ones we proposed. In particular we believe it could be possible to inspire more from the X and Y formulation (§2.5.5.5) to find a BSDF.

We also would have liked to have a better validation of some of our approaches, such as the assumption of a disk-shaped collector. Unfortunately, although there exist now a few open rendering packages such as *pbrt*(§5.2), a model of the moments of the spatial distribution of entry points of light paths on the lit surface of a slab (§5.3) and a model of the BSDF of a slab (§5.4)., we did not find any that could handle the complex rendering of a cloud.

Nevertheless, this thesis has been a very interesting and fruitful experience.

Appendix A

The chopped-peak model

We consider the narrow peak to be within $\Theta_p = 8°$. The chopped-peak approximation that is described in [Len85] is to replace the peak by a constant value, *i.e.,*

$$\tilde{P}(\Theta) = \begin{cases} P_{\text{ref}}(\Theta_p) & , \quad \Theta < \Theta_p, \\ P_{\text{ref}}(\Theta) & , \quad \Theta >= \Theta_p, \end{cases}$$

where $\tilde{P}(\Theta)$ represents the chopped phase function.

The removed peak is

$$P_{\text{peak}}(\Theta) = P_{\text{ref}}(\Theta) - \tilde{P}(\Theta), \tag{A.1}$$

and its weight is

$$f = 2 \int_{-\pi}^{\pi} P_{\text{peak}}(\Theta) \sin\Theta d\Theta. \tag{A.2}$$

The new phase function is normalized and the extinction coefficient is diminished to compensate the lost of the peak, *i.e.,*

$$P_{\text{chopped}}(\Theta) = \frac{\tilde{P}(\Theta)}{1-f}, \tag{A.3}$$

$$\kappa_{\text{chopped}} = f\rho\sigma_e. \tag{A.4}$$

Replacing the peak by a constant value creates a strong singularity at $\Theta = \Theta_p$ (see Figure A.1), which is visible to the eye and gives a very artificial look to the cloud at rendering time (indeed, the human eye notices easily any discontinuities in an image).

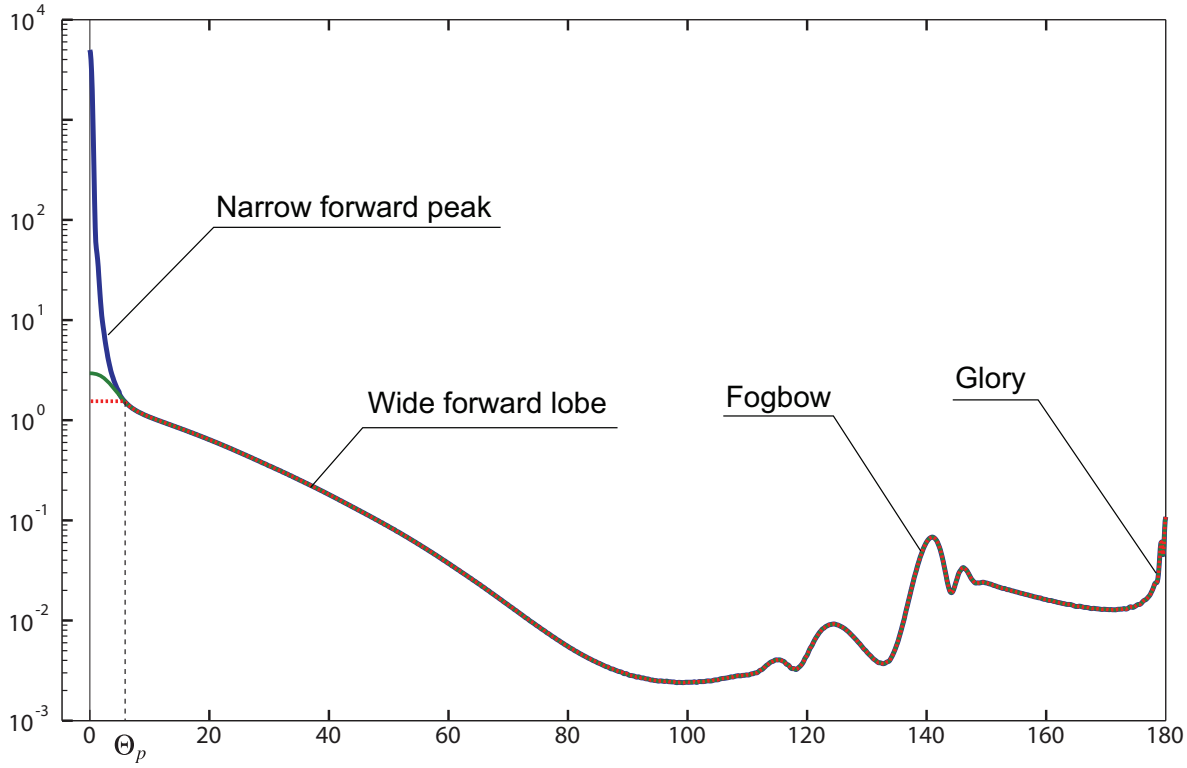


Figure A.1: Blue: *Reference Mie phase function.* Red: *Literally chopping the peak will result in a strong saliency in rendering.* Green: *Or solution is to replace the peak by a small lobe that extrapolates the wide forward lobe in a continuous manner.*

To overcome this problem, instead of replacing the peak by a constant value we replace it by a piece of Gaussian function so that $\tilde{P}(\Theta)$ is C^1-continuous (see Figure A.1), that is

$$\tilde{P}(\Theta) = \begin{cases} ae^{-\frac{(1-\Theta)^2}{2c^2}} & , \quad \Theta < \Theta_p, \\ P_{\text{ref}}(\Theta) & , \quad \Theta >= \Theta_p, \end{cases} \tag{A.5}$$

where a and c are chosen so that $\tilde{P}(\Theta_p) = P_{\text{ref}}(\Theta_p)$ and $\frac{d\tilde{P}(\Theta_p)}{d\Theta} = \frac{dP_{\text{ref}}(\Theta_p)}{d\Theta}$.

We measured that the weight f of the narrow-forwarding peak (Equation A.2) is about 50% depending on the DSD used. This means that about half of the scattering events are strongly forward scattering. In our study, for $\gamma = 2$ and $r_e = 3\,\mu\text{m}$ we have $f = 0.43$. This yields a mean free path $l_0 = 46\,\text{m}$ and an asymmetry factor (Equation 2.4) $g = 0.74$.

Notation index

Notation	Description	Page List
$1-\alpha$	confidence level	97, 260
$C(n, z_1, z, \theta, \psi, \theta')$	bidirectional scattering distribution function	117, 260
$D(n, z_1, z, x', y', \theta, \psi, \theta')$	surface density distribution	117, 260
$E(\boldsymbol{p})$	illuminance, irradiance	27, 260
$I(\boldsymbol{p}, \vec{s})$	luminous intensity, radiant intensity	26, 260
$L(\boldsymbol{p},\theta,\phi)$	luminance, radiance	26, 260
$L_0(\boldsymbol{p})$	liquid water content	58, 260
$M(\boldsymbol{p})$	exitance	28, 260
$N(r)$	droplet size distribution	58, 260
$P(\Theta)$	phase function	61, 260
$R(\boldsymbol{p},\theta,\phi,\theta',\phi')$	bidirectional reflectance distribution function	28, 260
$R(\mu,\phi,\mu',\phi')$	diffuse reflection function	74, 260
$S(\boldsymbol{p},\vec{s},\boldsymbol{p}',\vec{s}')$	bidirectional subsurface reflectance distribution function	31, 260
$S(n, z_1, z, \theta, x', y', \psi, \theta')$	bidirectional sub-surface scattering distribution function	117, 260
$T(\boldsymbol{p},\theta,\phi,\theta',\phi')$	bidirectional transmittance distribution function	74, 260
$T(\mu,\phi,\mu',\phi')$	diffuse transmission function, transmittance	74, 260
Ω	projected solid angle	22, 260
Φ	luminous flux, luminous power, radiant flux, radiant power	25, 260
Θ	phase angle, scattering angle	61, 260
α	opacity	65, 260
$\beta(\boldsymbol{a},\boldsymbol{b})$	extinction function	64, 65, 260
γ	transmission coefficient	76, 260
κ_a	absorption coefficient	24, 260
κ_e	extinction coefficient	24, 64, 260
κ_s	scattering coefficient	24, 260
λ	wavelength	24, 260
$\nu(\lambda)$	luminous efficiency	24, 260

Notation	Description	Page List
ω	solid angle	22, 260
$\rho(\boldsymbol{p})$	droplet concentration	58, 260
σ_e	extinction cross-section	64, 260
$\tau(\boldsymbol{a},\boldsymbol{b})$	optical thickness, transmissivity, transparency	64, 260
ϖ_0	single scattering albedo	24, 260
c	confidence interval	97, 260
$j(\boldsymbol{p},\vec{\boldsymbol{s}})$	emission coefficient	70, 260
l_0	mean free path	65, 260
$n(r)$	normalized droplet size distribution	58, 260
r_e	effective radius	64, 260
r_m	mean radius	58, 260
r_n	characteristic radius	59, 260

Acronyms

Notation	Description
BRDF	bidirectional reflectance distribution function. 28
BSDF	bidirectional scattering distribution function. 117
BSSRDF	bidirectional subsurface reflectance distribution function. 31
BSSSDF	bidirectional sub-surface scattering distribution function. 117
BTDF	bidirectional transmittance distribution function. 74
CDF	cumulative distribution function. 97, 260
CG	computer graphics. 16, 260
DOF	degrees of freedom. 76
DSD	droplet size distribution. 58
FEM	finite element method. 84, 260
FPS	frames per second. 17
GPU	graphics processing unit. 99, 260
HDR	high dynamic range. 100, 260
LDR	low dynamic range. 100, 260
LWC	liquid water content. 58
MFP	mean free path. 65
nDSD	normalized droplet size distribution. 58
PDF	probability density function. 85, 96, 260
SIMD	single instruction, multiple data. 99, 260

General Index

Bibliography

[BBNM07] Antoine Bouthors, Eric Bruneton, Fabrice Neyret, and Nelson L. Max. Real-time subsurface scattering on the GPU. Technical report, INRIA, 2007. 2.6.3.3, 8.3.1

[BC06] Craig F. Bohren and Eugene E. Clothiaux. *Fundamentals of Atmospheric Radiation: An Introduction with 400 Problems*. Wiley-VCH, 2006. 2.5, 2.5.2.1, 2.5.3.4, 2.5.5.6

[BH98] Craig F. Bohren and Donald R. Huffman. *Absorption and Scattering of Light by Small Particles*. Wiley-Interscience, 1998. ISBN: 0471293407. 2.4.2, 2.5, 2.5.1.2, 2.6.2, 2.6.4

[Bli77] James F. Blinn. Models of light reflection for computer synthesized pictures. In *SIGGRAPH '77: Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 192–198, New York, NY, USA, 1977. ACM. 1.5.9.3

[Bli82] James F. Blinn. Light reflection functions for simulation of clouds and dusty surfaces. In *SIGGRAPH'82*, pages 21–29, 1982. 2.5.4.4, 2.6.3.2

[BMS+02] H. W. Barker, A. Marshak, W. Szyrmer, A. Trishchenko, J.-P. Blanchet, and Z. Li. Inference of cloud optical depth from aircraft-based solar radiometric measurements. *Journal of the Atmospheric Sciences*, 59(13):2093–2111, 2002. 1.1

[BN04] Antoine Bouthors and Fabrice Neyret. Modeling clouds shape. In *Eurographics (short papers)*, august 2004. 2.6.1.3, 8.3.2, 8.3.3

[BN07] Antoine Bouthors and Matthieu Nesme. Twinned meshes for dynamic triangulation of implicit surfaces. In *Graphics Interface*, Montréal, may 2007. 8.3.3

[BNL06] Antoine Bouthors, Fabrice Neyret, and Sylvain Lefebvre. Real-time realistic illumination and shading of stratiform clouds. In *Eurographics Workshop on Natural Phenomena*, sep 2006. 4.4.2.3, 6.1, 6.7, 8.4

[BNM+08] Antoine Bouthors, Fabrice Neyret, Nelson L. Max, Eric Bruneton, and Cyril Crassin. Interactive multiple anisotropic scattering in clouds. In *ACM SIGGRAPH Symposium on Interactive 3D graphics and games (I3D)*, 2008. 7.1

[Boh01] Craig F. Bohren. *Clouds in a Glass of Beer: Simple Experiments in Atmospheric Physics*. Courier Dover Publications, 2001. 2.5.3.4

[Cha60] S. Chandrasekhar. *Radiative transfer*. New York: Dover, 1960. 1.5.11, 2.5, 2.5.4.2, 6, 2.5.5.1, 2.5.5.5, 2.6.3.1, 5.4.2, 8.2

[Cha02] Jean-Pierre Chalon. *Combien pèse un nuage ?* EDP Sciences, 2002. 2.4.1

[Cla74] Terry L. Clark. A Study in Cloud Phase Parameterization Using the Gamma Distribution. *Journal of the Atmospheric Sciences*, 31(1):142–155, 1974. 2.4.2

[CN07] Cyril Crassin and Fabrice Neyret. Représentation et algorithmes pour l'exploration interactive de volumes procéduraux étendus et détaillés. Master's thesis, INPG/UJF M2R - IVR, 2007. 7.2.1, 7.6.1, 7.6.3.1, 8.3.2

[Con00] Jeff Conrad. Getting the right exposure when photographing the moon, 2000. 1.1

[Cor01] Cornell University Program of Computer Graphics. Reflectance data, 2001. http://www.graphics.cornell.edu/online/measurements/reflectance/index.html. 1.5.9.3

[CRB+06] W.D. Collins, P.J. Rasch, B.A. Boville, J.J. Hack, J.R. McCaa, D.L. Williamson, B.P. Briegleb, C.M. Bitz, S.J. Lin, and M. Zhang. The Formulation and Atmospheric Simulation of the Community Atmosphere Model Version 3 (CAM3). *Journal of Climate*, 19(11):2144–2161, 2006. 1.1

[CRW+94] R. F. Cahalan, W. Ridgway, W. J. Wiscombe, S. Gollmer, and Harshvardhan. Independent Pixel and Monte Carlo Estimates of Stratocumulus Albedo. *Journal of Atmospheric Sciences*, 51:3776–3790, December 1994. 6.7

[CT82] R.L. Cook and K.E. Torrance. A Reflection Model for Computer Graphics. *ACM Transactions on Graphics*, 1(1):7–24, 1982. 1.5.9.3

[CWE97] L. H. Chambers, B. A. Wielicki, and K. F. Evans. Accuracy of the independent pixel approximation for satellite estimates of oceanic boundary layer cloud optical depth. *J. Geophys. Res*, 102:1779–1794, 1997. 6.7

[Dal92] S. J. Daly. Visible differences predictor: an algorithm for the assessment of image fidelity. In B. E. Rogowitz, editor, *Proc. SPIE Vol. 1666, p. 2-15, Human Vision, Visual Processing, and Digital Display III, Bernice E. Rogowitz; Ed.*, volume 1666 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 2–15, August 1992. 1.1.4.2

[DCWP02] Kate Devlin, Alan Chalmers, Alexander Wilkie, and Werner Purgathofer. Star: Tone reproduction and physically based spectral rendering. In Dieter Fellner and Roberto Scopignio, editors, *State of the Art Reports, Eurographics 2002*, pages 101–123. The Eurographics Association, September 2002. 3.5

[Dei64] D. Deirmendjian. Scattering and polarization properties of water clouds and hazes in the visible and infrared. *Applied Optics IP*, 3:187–+, February 1964. 4.2.3.2

[Dei69] D. Deirmendjian. *Electromagnetic Scattering on Spherical Polydispersions*. American Elsevier Publishing Company, New York, 1969. ISBN 0444000380. 2.4.2, 2.5

[DEYN07] Yoshinori Dobashi, Yoshihiro Enjyo, Tsuyoshi Yamamoto, and Tomoyuki Nishita. A fast rendering method for clouds illuminated by lightning taking into account multiple scattering. *The Visual Computer*, 23(9-11):697–705, 2007. 2.6.3.2

[DGS82] W. L. Dunn, R. D. M. Garcia, and C. E. Siewert. Concise and accurate solutions for chandrasekhar's X and Y functions. *Astrophysical Journal*, 260:849–854, September 1982. 2.5.5.5, 4.3.5

[DK02] Stanislav Darula and Richard Kittler. Cie general sky standard defining luminance distributions. In *eSim'02 Proceedings*, 2002. 1.12

[DKY+00] Yoshinori Dobashi, Kazufumi Kaneda, Hideo Yamashita, Tsuyoshi Okita, and Tomoyuki Nishita. A simple, efficient method for realistic animation of clouds. In *SIGGRAPH'00*, pages 19–28, July 2000. 2.31(a), 2.31, 2.6.1.2, 2.6.3.2, 2.6.3.2, 2.6.3.2, 2.6.4, 4.2.5.6, 7.9

[DNYO98] Yoshinori Dobashi, Tomoyuki Nishita, Hideo Yamashita, and Tsuyoshi Okita. Modeling of clouds from satellite images using metaballs. *pg*, 00:53, 1998. 2.6.1.3, 2.6.3.2

[Dra88] J. F. Drake. *Retrieving the Distribution of Liquid Water Content in Clouds from Observations of Thermal Microwave Radiation.* PhD thesis, UNIVERSITY OF CALIFORNIA, LOS ANGELES., 1988. 2.4.2

[DS03] Carsten Dachsbacher and Marc Stamminger. Translucent shadow maps. In *Eurographics Workshop on Rendering (EGWR)*, pages 197–201, 2003. 2.6.3.3, 7.8.2, 8.3.1

[Dun04] Jody Duncan. The day after tomorrow. *Cinefex*, 98:75–93, 2004. 2.6.3.3

[DYN01] Yoshinori Dobashi, Tsuyoshi Yamamoto, and Tomoyuki Nishita. Efficient rendering of lightning taking into account scattering effects due to cloud and atmospheric particles. In *PG '01: Proceedings of the 9th Pacific Conference on Computer Graphics and Applications*, page 390, Washington, DC, USA, 2001. IEEE Computer Society. 2.6.3.2

[Ebe97] David S. Ebert. Volumetric modeling with implicit functions (a cloud is born). In *SIGGRAPH'97 Sketches*, page 245, 1997. 2.6.1.3, 2.6.1.3

[ES00] Pantelis Elinas and Wolfgang Stürzlinger. Real-time rendering of 3D clouds. *J. Graph. Tools*, 5(4):33–45, 2000. 2.6.1.3, 7.9

[Eva98] K. F. Evans. The Spherical Harmonics Discrete Ordinate Method for Three-Dimensional Atmospheric Radiative Transfer. *Journal of Atmospheric Sciences*, 55:429–446, February 1998. 2.6.3.1

[EVNT78] G Eason, A R Veitch, R M Nisbet, and F W Turnbull. The theory of the back-scattering of light by blood. *Journal of Physics D: Applied Physics*, 11(10):1463–1479, 1978. 2.6.3.3

[EW04] K. F. Evans and W. J. Wiscombe. An algorithm for generating stochastic cloud fields from radar profile statistics. *Atmospheric Research*, 72:263–289, November 2004. 2.6.1.3

[Fer03] J. Ferwerda. Three varieties of realism in computer graphics. In *SPIE Human Vision and Electronic Imaging*, pages 290–297, 2003. 1.1.4.1

[FPW92] Thomas J. Farrell, Michael S. Patterson, and Brian Wilson. A diffusion theory model of spatially resolved, steady-state diffuse reflectance for the noninvasive determination of tissue optical properties in vivo. *Medical Physics*, 19(4):879–888, 1992. 2.6.3.3

[FSJ01] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. Visual simulation of smoke. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 15–22, August 2001. 2.6.3.4

[FvDFH90] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer graphics: principles and practice (2nd ed.)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990. 2.6.3.3, 6.3.4.1, 6.3.4.1

[Gar85] Geoffrey Y. Gardner. Visual simulation of clouds. In *SIGGRAPH'85*, pages 297–304. ACM Press, 1985. 2.6.1.3, 2.32, 2.6.1.3, 2.6.3.2, 7.2.1

[GGSC96] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, New York, NY, USA, 1996. ACM. 2.6.3.1

[Gor94] Howard R. Gordon. Equivalence of the point and beam spread functions of scattering media: a formal demonstration. *Appl. Opt.*, 33(6):1120, 1994. 2.5.3.3

[GTGB84] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. *SIGGRAPH Comput. Graph.*, 18(3):213–222, 1984. 2.6.3.3

[Gus05] Stefan Gustavson. Simplex noise demystified. Technical report, Linköping University, 2005. 7.6.1

[GWWH03] Nolan Goodnight, Rui Wang, Cliff Woolley, and Greg Humphreys. Interactive time-dependent tone mapping using programmable graphics hardware. In *Eurographics Workshop on Rendering (EGRW)*, pages 26–37, 2003. 3.5, 7.6.4

[Hal93] Charles P. Halsted. Brightness, luminance, and confusion. *Information Display*, March 1993. 1.1

[Han71] James E. Hansen. Multiple scattering of polarized light in planetary atmospheres. part i. the doubling method. *Journal of the Atmospheric Sciences*, 28(1):120–125, 1971. 2.5.5.6

[HAP05] Kyle Hegeman, Michael Ashikhmin, and Simon Premože. A lighting model for general participating media. In *ACM SIGGRAPH Symposium on Interactive 3D graphics and games (I3D)*, pages 117–124, 2005. 2.6.2, 2.6.3.2

[HBSL03] Mark J. Harris, William V. Baxter, Thorsten Scheuermann, and Anselmo Lastra. Simulation of cloud dynamics on graphics hardware. In *HWWS '03: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 92–101, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 2.30, 2.6.1.2, 7.9

[HG40] L. G. Henyey and J. L. Greenstein. Diffuse radiation in the Galaxy. *Annales d'Astrophysique*, 3:117–+, January 1940. 2.6.2

[HK93] Pat Hanrahan and Wolfgang Krüger. Reflection from layered surfaces due to subsurface scattering. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 165–174, New York, NY, USA, 1993. ACM. 1.5.10, 2.6.3.3

[HKO07] Jeonghun Han, Hyunwoo Ki, and Kyoungsu Oh. Gpu ray marching for real-time rendering of participating media. *Computational Science and its Applications, 2007. ICCSA 2007. International Conference on*, pages 499–504, 26-29 Aug. 2007. 7.8.3

[HL01] Mark J. Harris and Anselmo Lastra. Real-time cloud rendering. *Computer Graphics Forum*, 20(3):76–84, 2001. 2.6.2, 2.6.3.2, 2.6.3.2, 2.39, 4.2.5.6

[HP03] Naty Hoffman and Arcot J. Preetham. *in Graphics programming methods*, chapter Real-time light-atmosphere interactions for outdoor scenes, pages 337–352. Charles River Media, Inc., 2003. 2.6.4, 7.6.4

[HTSG91] Xiao D. He, Kenneth E. Torrance, François X. Sillion, and Donald P. Greenberg. A comprehensive physical model for light reflection. *SIGGRAPH Comput. Graph.*, 25(4):175–186, 1991. 1.5.9.3, 8.2

[I3R08] I3RC. Publicly available codes to generate heterogeneous cloud structures, 2008. http://i3rc.gsfc.nasa.gov/Public_codes_clouds.htm. 2.6.1.3

[JAD+03] Henrik Wann Jensen, James Arvo, Phil Dutre, Alexander Keller, Art Owen, Matt Pharr, and Peter Shirley. Monte carlo ray tracing. SIGGRAPH course 44, July 2003. 2.6.3.1

[JC98] Henrik Wann Jensen and Per H. Christensen. Efficient simulation of light transport in scences with participating media using photon maps. In *SIGGRAPH'98*, pages 311–320. ACM Press, 1998. 2.6.3.1

[Jen96] Henrik Wann Jensen. Global illumination using photon maps. In *Proceedings of the eurographics workshop on Rendering techniques '96*, pages 21–30, London, UK, 1996. Springer-Verlag. 2.6.3.1

[Jen01] Henrik Wann Jensen. *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd., 2001. 2.6.3.1, 4.3.1, 4.3.3

[JMLH01] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *SIGGRAPH'01*, pages 511–518, 2001. 1.5.10, 2.6.3.3, 7.8.4, 8.2, 8.3.1

[Ken51] J.F. Kenney. *Mathematics of statistics*. D. Van Nostrand Princeton, NJ, 1951. 3.2.1

[Kor69] G. Kortüm. *Reflectance spectroscopy. Principles, methods, applications*. Springer-Verlag, 1969. 1.5.10

[KPHE02] Joe Kniss, Simon Premoze, Charles Hansen, and David Ebert. Interactive translucent volume rendering and procedural modeling. In *VIS '02: Proceedings of the conference on Visualization '02*, pages 109–116, Washington, DC, USA, 2002. IEEE Computer Society. 2.6.1.3, 2.6.3.2

[Kv84] James T. Kajiya and Brian P. von Herzen. Ray tracing volume densities. In *SIGGRAPH'84*, pages 165–174, 1984. 2.6.1.2, 2.6.3.2

[Lam60] J.H. Lambert. *Photometria sive de mensura et gradibus luminis colorum et umbrae*. Sumptibus Vidae Eberhardi Klett, 1760. 1.5.9.3

[Lav] Philip Laven. Mieplot. http://www.philiplaven.com/mieplot.htm. 2.5.1.3, 2.6.2

[LCH95] Richard F. Lutomirski, Anthony P. Ciervo, and Gainford J. Hall. Moments of multiple scattering. *Appl. Opt.*, 34(30):7125, 1995. 2.5.3.3

[Len85] Jacqueline Lenoble. *Radiative transfer in scattering and absorbing atmospheres: standard computational procedures*. A. Deepak Publishing, 1985. 2.5, 2.5.4.4, 2.5.5.6, 2.6.2, 4.2.3.2, 8.4, A

[Lev58] L. M. Levin. Functions to represent drop size distribution in clouds. the optical density of clouds. *Izv. Akad. Nauk. SSSR, Ser. Geofiz.*, 10:198–702, 1958. 2.4.2

[LF02] Arnauld Lamorlette and Nick Foster. Structural modeling of flames for a production environment. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 729–735, New York, NY, USA, 2002. ACM. 2.6.1.2, 2.6.3.4

[LH96] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, New York, NY, USA, 1996. ACM. 2.6.3.1

[LL95] David K. Lynch and William Livingston. *Color and Light in Nature*. Cambridge University Press, 1995. ISBN: 0521775043. 2.5

[LV00] Tom Lokovic and Eric Veach. Deep shadow maps. In *SIGGRAPH'00*, 2000. 7.8.2

[Mas71] B.J. Mason. *The physics of clouds*. Clarendon Press, 1971. 2.4.2, 2.6.1.1

[Max86] Nelson L. Max. Light diffusion through clouds and haze. *Computer Vision, Graphics and Image Processing*, 33(3):280–292, March 1986. 2.6.1.3, 4.2.5.6

[Max94a] Nelson L. Max. Computer animation of clouds. In *Proceedings of Computer Animation '94*, pages 167–174, 204, 1994. 2.6.1.3

[Max94b] Nelson L. Max. Efficient light propagation for multiple anisotropic volume scattering. In *Eurographics Workshop on Rendering (EGWR)*, pages 87–104, 1994. 2.6.2, 2.6.3.2

[MCH87] J. W. McLean, D. R. Crawford, and C. L. Hindman. Limits of small angle scattering theory. *Appl. Opt.*, 26:2053–2054, June 1987. 2.5.3.3

[MDN04] Ryo Miyazaki, Yoshinori Dobashi, and Tomoyuki Nishita. A fast rendering method of clouds using shadow-view slices. In *Proc. CGIM*, volume 8, pages 93–98, 2004. 2.6.3.2

[MFSR77] L. E. Mertens and Jr. F. S. Replogle. Use of point spread and beam spread functions for analysis of imaging systems in water. *J. Opt. Soc. Am.*, 67(8):1105, 1977. 2.5.3.3

[MFW98] John W. McLean, Jonathan D. Freeman, and Ronald E. Walker. Beam spread function with time dispersion. *Appl. Opt.*, 37(21):4701–4711, 1998. 2.5.3.3

[Mie08] G. Mie. Beiträge zur optik trüber medien, speziell kolloidaler metallösungen. *Ann. Phys.*, 25:377–445, 1908. 2.5, 2.5.1.2

[Min54] M. Minnaert. *The nature of Light & Color in the open air*. Dover Publicatiosn Inc, 1954. 2.5

[Mob94] Curtis D. Mobley. *Light and water: radiative transfer in natural waters*. 1994. 4.5

[MOD+99] A. Marshak, L. Oreopoulos, A. B. Davis, W. J. Wiscombe, and R. F. Cahalan. Horizontal radiative fluxes in clouds and accuracy of the Independent Pixel Approximation at absorbing wavelengths. *Geophysical Research Letters*, 26:1585–1588, June 1999. 6.7

[MSPS04] B. Mayer, M. Schröder, R. Preusker, and L. Schüller. Remote sensing of water cloud droplet size distributions using the backscatter glory: a case study. *Atmospheric Chemistry and Physics*, 4:1255–1263, Aug 2004. 2.6.2

[MVC00] Natasha L. Miles, Johannes Verlinde, and Eugene E. Clothiaux. Cloud Droplet Size Distributions in Low-Level Stratiform Clouds. *Journal of the Atmospheric Sciences*, 57(2):295–311, 2000. 6.2

[MYND01] Ryo Miyazaki, Satoru Yoshida, Tomoyuki Nishita, and Yoshinori Dobashi. A method for modeling clouds based on atmospheric fluid dynamics. In *PG '01: Proceedings of the 9th Pacific Conference on Computer Graphics and Applications*, page 363, Washington, DC, USA, 2001. IEEE Computer Society. 2.31(b), 2.6.1.2, 2.6.3.2

[Nat] National Snow and Ice Data Center. Artic climatology and meteorology — water sky and ice blink. http://nsidc.org/arcticmet/basics/phenomena/water_sky.html. 2.6

[NB06] Matthieu Nesme and Antoine Bouthors. Dynamic triangulation of implicit surfaces: towards the handling of topology changes. Technical Report RR-6128, INRIA, Nov 2006. 8.3.3

[ND99] Tomoyuki Nishita and Yoshinori Dobashi. Modeling and rendering methods of clouds. In *Pacific Conference on Computer Graphics and Applications*, pages 218–, 1999. 2.6.3.2

[Ney97] Fabrice Neyret. Qualitative simulation of connective cloud formation and evolution. In *Eurographics Workshop on Computer Animation and Simulation (SCA)*, 1997. 2.6.1.3

[Ney00] Fabrice Neyret. A phenomenological shader for the rendering of cumulus clouds. Technical Report RR-3947, INRIA, May 2000. 2.6.1.3, 2.6.3.2, 7.2.1

[Ney03] Fabrice Neyret. Advected textures. In *ACM SIGGRAPH/EG Symposium on Computer Animation (SCA)*, july 2003. 2.6.1.3, 6.4

[Nic65] Fred E. Nicodemus. Directional reflectance and emissivity of an opaque surface. *Appl. Opt.*, 4(7):767, 1965. 1.5.9.1

[NMN87] Tomoyuki Nishita, Yasuhiro Miyawaki, and Eihachiro Nakamae. A shading model for atmospheric scattering considering luminous intensity distribution of light sources. In *SIGGRAPH'87*, pages 303–310, New York, NY, USA, 1987. ACM Press. 2.6.4

[NN03] S.G. Narasimhan and S.K. Nayar. Shedding light on the weather. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 1:I–665–I–672 vol.1, 18-20 June 2003. 2.5.3.3

[NND96] Tomoyuki Nishita, Eihachiro Nakamae, and Yoshinori Dobashi. Display of clouds taking into account multiple anisotropic scattering and sky light. In *SIGGRAPH'96*, pages 379–386, August 1996. 2.2.2.1, 2.6.1.3, 2.6.2, 2.6.3.2, 2.6.3.2, 4.2.5.6

[NRH+77] Fred E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. Geometrical considerations and nomenclature for reflectance. NBS Monograph 160, National Bureau of Standards, Washington, D.C., USA, October 1977. 1.5, 1.5.10, 4.2.2, 7.3.1

[O'N05] Sean O'Neil. *GPU Gems 2*, chapter Accurate Atmospheric Scattering, pages 235–268. Addison Wesley, 2005. 6.3.3

[PAS03] Simon Premože, Michael Ashikhmin, and Peter Shirley. Path integration for light transport in volumes. In *Eurographics Symposium on Rendering (EGSR)*, pages 52–63, 2003. 2.5.3.3, 2.6.3.2, 2.38, 4.2.5.4, 7.4.3.1

[PAT+04] Simon Premože, Michael Ashikhmin, Jerry Tessendorf, Ravi Ramamoorthi, and Shree Nayar. Practical rendering of multiple scattering effects in participating media. In *Eurographics Symposium on Rendering (EGSR)*, pages 363–374, June 2004. 2.6.2, 2.6.3.2

[Pea86] Willam A. Pearce. Monte Carlo study of the atmospheric spread function. *Appl. Opt.*, 25:438–447, February 1986. 2.5.3.3

[Per85] Ken Perlin. An image synthesizer. In *SIGGRAPH'85*, pages 287–296, July 1985. 2.6.1.3, 6.4, 7.2.2, 7.6.1

[PH89] Ken Perlin and Eric M. Hoffert. Hypertexture. In *SIGGRAPH'89*, pages 253–262, July 1989. 7.1, 7.2.1, 7.2.2

[Pho75] Bui Tuong Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, 1975. 1.5.9.3

[PK97] Hans R. Pruppacher and James D. Klett. *Microphysics of Clouds and Precipitation*. Kluwer Academic Publishers, 1997. 2.4.2

[PSS99] Arcot J. Preetham, Peter Shirley, and Brian Smits. A practical analytic model for daylight. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 91–100, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 2.6.4, 7.6.4

[QXF+07] Feng Qiu, Fang Xu, Zhe Fan, Neophytou Neophytos, Arie Kaufman, and Klaus Mueller. Lattice-based volumetric global illumination. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1576–1583, 2007. 2.6.3.1

[Ray83] L. Rayleigh. Investigation of the character of the equilibrium of an incompressible heavy fluid of variable density. *Proc. Lond. Math. Soc*, 14:170–177, 1883. 1.1.5.1

[RBNH07] Damien Rumiano, Antoine Bouthors, Matthieu Nesme, and Franck Hetroy. Topological modifications of animated surfaces. ENSIMAG-INPG 2nd year training period report, August 2007. 8.3.3

[REHL03] Kirk Riley, David Ebert, Charles Hansen, and Jason Levit. Visually accurate multi-field weather visualization. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 37, Washington, DC, USA, 2003. IEEE Computer Society. 2.29(a), 2.6.1.1, 2.6.1.3

[REK+04] Kirk Riley, David S. Ebert, Martin Kraus, Jerry Tessendorf, and Charles Hansen. Efficient rendering of atmospheric phenomena. In *Eurographics Symposium on Rendering (EGSR)*, pages 375–386, June 2004. 2.29(b), 2.6.1.1, 2.6.2, 2.6.3.2, 7.9

[RPG99] Mahesh Ramasubramanian, Sumanta N. Pattanaik, and Donald P. Greenberg. A perceptually based physical error metric for realistic image synthesis. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 73–82, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 1.1.4.2

[RT87] Holly E. Rushmeier and Kenneth E. Torrance. The zonal method for calculating light intensities in the presence of a participating medium. In *SIGGRAPH'87*, pages 293–302. ACM Press, 1987. 2.6.3.1

[RVL97] HWJ Russchenberg, VKC Venema, and ACA Lammeren. P. van and Apituley, A., 1997: Radar-lidar observations of clouds during the CLARA-campaigns. In *Proceedings of the Workshop on Synergy of Active Instrumens in the Earth Radiation Mission*, 1997. 2.18

[SBA+00] David E. Stevens, John B. Bell, Ann S. Almgren, Vince E. Beckner, and Charles A. Rendleman. Small-scale processes and entrainment in a stratocumulus marine boundary layer. *Journal of the Atmospheric Sciences*, 57(4):567–581, 2000. 2.6.3.4

[SD02] Marc Stamminger and George Drettakis. Perspective shadow maps. In John Hughes, editor, *Proceedings of ACM SIGGRAPH 2002*. ACM Press/ ACM SIGGRAPH, July 2002. 7.8.3

[Sel92] Patricia Seligman. *Painting the effects of weather*. North Light Books, 1992. (document), 2.14, 2.3, 2.15, 2.16

[SGA+07] Veronica Sundstedt, Diego Gutierrez, Oscar Anson, Francesco Banterle, and Alan Chalmers. Perceptual rendering of participating media. *ACM Trans. Appl. Percept.*, 4(3):15, 2007. 1.1.4.2

[SKS02] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 527–536, New York, NY, USA, 2002. ACM. 2.6.3.1

[SKSU05] László Szirmay-Kalos, Mateu Sbert, and Tamás Umenhoffer. Real-time multiple scattering in participating media with illumination networks. In *Rendering Techniques*, pages 277–282, 2005. 2.6.3.1

[SM03] P. Shirley and R.K. Morley. *Realistic Ray Tracing*. AK Peters, Ltd., 2003. 2.6.3.1

[SP89] François Sillion and Claude Puech. A general two-pass method integrating specular and diffuse reflection. In *Proceedings of SIGGRAPH '89*, pages 335–344. ACM Press, 1989. 6.3.4.1

[SSEH03] Joshua Schpok, Joseph Simons, David S. Ebert, and Charles Hansen. A real-time cloud modeling, rendering, and animation system. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, pages 160–166, 2003. 2.6.1.3, 2.6.1.3, 2.33, 2.6.3.2, 2.38

[Sta95a] Jos Stam. *Multi-scale stochastic modelling of complex natural phenomena*. PhD thesis, University of Toronto, Toronto, Ont., Canada, Canada, 1995. 2.6.3.2

[Sta95b] Jos Stam. Multiple Scattering as a Diffusion Process. In *Eurographics Workshop on Rendering (EGWR)*, pages 41–50, 1995. 2.6.3.2, 2.6.3.3

[Sto78] L. B. Stotts. Closed form expression for optical pulse broadening in multiple-scattering media (t). *Appl. Opt.*, 17(4):504, 1978. 2.5.3.3

[Tan89] Z. Tan. Radiative heat transfer in multidimensional emitting, absorbing, and anisotropic scattering media - Mathematical formulation and numerical method. *ASME Transactions Journal of Heat Transfer*, 111:141–147, February 1989. 2.6.3.1

[TB02] Andrzej Trembilski and Andreas Broßler. Surface-based efficient cloud visualisation for animation applications. In *WSCG*, pages 453–460, 2002. 2.6.1.3, 2.6.3.2, 2.6.3.2

[TS67] Kenneth E. Torrance and E.M. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of the Optical Society of America*, 57(9):1105–1114, 1967. 1.5.9.3

[TW03] Jerry A. Tessendorf and D. Wasson. Impact of multiple scattering on simulated infrared cloud scene images. *Proceedings of SPIE*, 2223:461, 2003. 2.5.3.3, 5.4.3

[Tyl60] J.E. Tyler. *Radiance Distribution as a Function of Depth in an Underwater Environment.* University of California Press, 1960. 4.5

[van80] H. C. van de Hulst. *Multiple Light Scattering*. Academic Press, New York, 1980. 2.5.5.6

[vdH81] H. C. van de Hulst. *Light Scattering by Small Particles*. Dover Publications, New York, 1981. 2.5

[Wan04] Niniane Wang. Realistic and fast cloud rendering. *journal of graphics tools*, 9(3):21–40, 2004. 2.6.1.3

[War92] Gregory J. Ward. Measuring and modeling anisotropic reflection. *SIGGRAPH Comput. Graph.*, 26(2):265–272, 1992. 1.5.9.3

[WBC08] Jamie Wither, Antoine Bouthors, and Marie-Paule Cani. Rapid sketch modeling of clouds. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling (SBIM)*, 2008. 8.3.2

[Wik08] Wikipedia, The Free Encyclopedia. Albedo, 2008. [Online; accessed 29-February-2008]. 1.1

[Wor87] World Meteorological Organization. *International Cloud Atlas*. World Meteorological Organization, 1987. 2.2.1.2