THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 3 - Paul Sabatier

Présentée et soutenue par

Pierre MÉZIÈRES

Le 29 novembre 2022

Harmoniques sphériques réelles pour la simulation de l'éclairage et le rendu temps-réel

---

École doctorale : **EDMITT - École Doctorale Mathématiques, Informatique et Télécommunications de Toulouse**

Spécialité : **Informatique et Télécommunications**

Unité de recherche :
IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par
Mathias PAULIN

Jury

M. George DRETTAKIS, Rapporteur
Directeur de recherche, Inria Sophia-Antipolis

M. Cyril SOLER, Rapporteur
Chargé de recherche, Inria Grenoble Alpes

Mme Julie DIGNE, Examinatrice
Chargée de recherche, Université Claude Bernard Lyon 1

M. Laurent BELCOUR, Examinateur
Docteur, Unity Technologies

M. Denis KOUAMÉ, Examinateur
Professeur, Université Toulouse 3 - Paul Sabatier

M. Mathias PAULIN, Directeur de thèse
Professeur, Université Toulouse 3 - Paul Sabatier

Remerciements

« *Miladiou, me pren pas per un roumégaïre, soi just un doctorant !* »

C'est toute une aventure qui se termine et je tiens à prendre quelques lignes afin de remercier les personnes qui ont contribué à leur manière à cette thèse.

En premier lieu, je remercie mon encadrant, Mathias, dont l'expérience a été une boussole précieuse tout au long de ces trois ans, ainsi que pour sa vision de la recherche qui a laissé une trace durable dans mon esprit et qui aiguillera, sans nul doute, ma carrière professionnelle. Merci à David pour son humanité incroyable, ainsi qu'à Nicolas et Loïc pour leurs remarques scientifiques avisées et leur aide précieuse. C'était un réel plaisir d'avoir pu réaliser cette thèse à leurs côtés et d'avoir fait partie de l'équipe STORM dont le talent scientifique est très inspirant. Je remercie également tous ceux qui connaissent la vie de doctorant et qui m'ont fait énormément grandir en si peu de temps, aussi bien professionnellement que personnellement : Amélie, Chems, Florian, les deux François, Hugo, Nicolas, Olivier, Thibault, Vassili, ainsi que Gauthier. Je salue aussi ceux que j'aurais aimé côtoyer un peu plus longtemps, mais je me sens tout de même chanceux d'avoir croisé leur route : Arthur, Mégane et Jules. Je ne peux souhaiter que le meilleur pour la suite à tout ce joli petit monde.

Je souhaite remercier toute ma famille, qui me fournit l'énergie nécessaire pour aller toujours de l'avant. Mes parents, Marie-Laure et Jean-François, pour leur patience et leur amour indéfectible, ainsi que pour la place au chaud qu'ils m'ont toujours gardée et qui m'a permis de me ressourcer et de m'aérer très souvent l'esprit, un aspect particulièrement important de la vie de doctorant. Mon frère Nicolas, dont je suis extrêmement fier, sans qui je n'aurais probablement jamais continué dans la voie des mathématiques et qui, malgré les péripéties que nous avons pu avoir, constitue aujourd'hui l'un des piliers les plus solides sur lesquels je peux m'appuyer pour continuer à me construire. Sans oublier, sa compagne Laura qui a apporté des pierres essentielles à la solidité de notre famille. Mes oncles et tantes : Joël, Martine, Nadine et Serge, pour leur présence et leur soutien inconditionnel, et mes cousins : Benjamin, Chloé, Julie, Théo et Samuel, avec lesquels je ne compte plus les bons moments et j'espère que cela pourra continuer très longtemps. On peut difficilement rêver d'une famille plus inspirante, de par la ténacité pour certains, la malice pour d'autres ou encore la bienveillance pour la plupart. Un merci infini aussi à ceux qui m'inspirent tant et qui, j'aime à croire, auraient été particulièrement fiers : Roger, Odette et Louis.

C'est aussi la fin d'un long parcours d'études, et je tiens à exprimer ma reconnaissance envers toutes les personnes intéressantes et motivantes que j'ai pu rencontrer tout au long de mes études, avec une pensée particulière pour Suzanne. Un grand merci aussi à ceux qui sont là depuis le lycée et avec qui c'est toujours un plaisir de se retrouver : Cyril, David, Jason et Nicolas. Merci également au TCS et plus spécialement à Julien, dont les séances de tennis sont un défouloir indispensable depuis tant d'années. Enfin, un dernier merci, mais pas des moindres, à ceux dont j'envierai toujours la philosophie, mais dont les ronflements n'ont pas non plus beaucoup aidé : Flash, Herma, Clochette et tous ceux avec qui j'ai eu le plaisir de partager une sieste, ou du moins la leur.

Résumé

La simulation efficace et de haute qualité de l'éclairage d'environnements 3D a de nombreuses applications : le rendu en temps réel permet des interactions transparentes dans les jeux vidéo et les applications de réalité virtuelle, tandis que le rendu interactif est utilisé par les créateurs de contenu pour évaluer l'aspect d'une scène en quelques secondes, et l'affiner progressivement avant de passer au rendu hors ligne. Ces applications modélisent une géométrie 3D très détaillée où chaque point de la géométrie reçoit de la lumière de toutes les directions. Une partie de l'énergie lumineuse est absorbée par le matériau et une autre partie est réémise potentiellement dans toutes les directions. En conséquence, le champ de radiance reçu ou émis en chaque point de la scène est une fonction sphérique. Ainsi, la simulation de l'éclairage introduit un besoin de caractériser efficacement ces fonctions sphériques.

Les harmoniques sphériques répondent à ce besoin en réduisant une fonction sphérique quelconque à un vecteur de coefficients, où chaque coefficient est rattaché à une fonction de base des harmoniques sphériques. Les harmoniques sphériques sont analogues aux séries de Fourier, décomposition en sinus et cosinus, mais définies sur la sphère. Ainsi les harmoniques sphériques décomposent le signal en fréquence et jouent le rôle d'un filtre passe-bas dont la fréquence de coupure dépend du nombre d'harmoniques sphériques impliquées. Par construction, les harmoniques sphériques sont définies sur le corps des complexes. Cependant, cette thèse se concentre sur une utilisation efficace de leurs homologues réelles.

En informatique graphique, les harmoniques sphériques sont historiquement utilisées pour stocker efficacement la radiance reçue en tout point de la scène. L'orthogonalité des fonctions de base permet notamment un calcul efficace de la radiance sortante. Néanmoins, le calcul des coefficients d'harmoniques sphériques représentant une radiance quelconque est coûteux et ne peut pas être réalisé en chaque point de la scène et pour chaque image pour le temps réel. Cela limite l'utilisation des harmoniques sphériques à des scènes statiques, dont la géométrie de la scène est généralement fixe et à des sources de lumières statiques elles-aussi, ou à des approximations grossières des coefficients d'harmoniques sphériques représentant le champ de radiance. Ainsi, la première contribution de cette thèse est un calcul efficace des coefficients d'harmoniques sphériques représentant le champ de radiance produit par des lumières sphériques, une dizaine de lumières peuvent-être utilisées en temps réel. La méthode est étendue afin de calculer le gradient spatial des coefficients d'harmoniques sphériques. Le gradient permet de ne calculer les coefficients que sur un ensemble parcimonieux de points tout en ayant une très bonne qualité d'interpolation des coefficients en tout point de la scène. Cela permet d'utiliser des centaines de lumières sphériques en temps réel. Le calcul du gradient des coefficients repose notamment sur une méthode efficace de calcul du gradient des fonctions de base des harmoniques sphériques proposée aussi dans cette thèse. Néanmoins, calculer efficacement les coefficients des fonctions de bases représentant le champ de radiance entrant n'est pas suffisant pour supporter des scènes à la géométrie dynamique. À cette fin, nous proposons un nouveau cadre de travail permettant de calculer efficacement la radiance sortante de n'importe quel point de la scène, en identifiant et en factorisant, notamment, certaines redondances liées aux harmoniques sphériques qui apparaissent dans le calcul de l'éclairage. Enfin, de nombreuses méthodes reposent sur des lumières virtuelles distribuées dans la scène selon l'éclairage primaire pour générer l'éclairage global. Nous proposons un nouveau modèle de lumières virtuelles basé sur le travail effectué pour les lumières sphériques.

Abstract

Efficient and high-quality simulation of lighting in 3D environments has many applications: real-time rendering enables seamless interactions in video games and virtual reality applications, while interactive rendering is used by content creators to assess the appearance of a scene in seconds, and gradually refine it before moving to offline rendering. These applications model highly detailed 3D geometry where each point in the geometry receives light from all directions. Some of the light energy is absorbed by the material and some is potentially re-emitted in all directions. Consequently, the radiance field received or emitted at each point of the scene is a spherical function. Thus, lighting simulation introduces a need to efficiently characterize these spherical functions.

Spherical harmonics address this need by reducing any spherical function to a vector of coefficients, where each coefficient is attached to a spherical harmonic basis function. Spherical harmonics are analogous to Fourier series, decomposition into sine and cosine, but defined on the sphere. Thus, the spherical harmonics decompose the signal in frequency and play the role of a low-pass filter whose cut-off frequency depends on the number of spherical harmonics involved. By construction, spherical harmonics are defined in a complex form. However, this thesis focuses on an efficient use of their real counterparts.

In computer graphics, spherical harmonics are historically used to efficiently store the radiance received at any point in the scene. The orthogonality of the basis functions allows an efficient computation of the outgoing radiance. Nevertheless, the computation of the spherical harmonics coefficients representing any radiance is expensive and cannot be performed at each point of the scene and for each image in real time. This limits the use of spherical harmonics to static scenes, with generally fixed scene geometry and static light sources, or to coarse approximations of the spherical harmonic coefficients representing the radiance field. Thus, the first contribution of this thesis is an efficient computation of the spherical harmonic coefficients representing the radiance field produced by spherical lights, a dozen lights can be used in real time. The method is extended to compute the spatial gradient of the spherical harmonic coefficients. The gradient allows computing the coefficients only on a parsimonious set of points while having a very good quality of interpolation of the coefficients in any point of the scene. This allows using hundreds of spherical lights in real time. The computation of the gradient of the coefficients relies in particular on an efficient calculation method of the gradient of the spherical harmonics basis function also proposed in this thesis. Nevertheless, efficiently computing the spherical harmonic coefficients representing the incoming radiance field is not sufficient to support scenes with dynamic geometry. To this end, we propose a new framework to efficiently compute the outgoing radiance of any point in the scene, by identifying and factoring some redundancies related to the spherical harmonics that appear in the final lighting computation. Finally, numerous methods rely on virtual lights distributed in the scene according to the primary lighting to generate the global lighting. We propose a new model of virtual lights based on the work done for spherical lights.

Table of contents

List of Figures

1 Introduction

Efficiency is a requirement for many applications in computer graphics. The efficiency in rendering spans 3 categories, real-time, interactive time and offline. Real-time corresponds to a computation time of the order of a few milliseconds, thus allowing a fast display frequency, as required in video games generally aiming to more than 60 *Frames Per Second* (FPS). The interactive time is of the order of less than a few seconds allowing a finer quality of rendering while the refresh rate is sufficient to interact efficiently, *Computer-Aided Design* (CAD) generally aim this category. Beyond that, it's called offline rendering, allowing maximum quality like the one wanted for the final film images. The process of film creation generally relies on interactive methods to setup the scenes and ultimately produce the final image over several dozen hours with offline rendering. Efficient rendering is synonymous

Figure 1.1: *From the dusty garage in Call of duty: Modern Warfare [Inf19, SSS+20] (top left) to the subtle misty exterior in The Last of Us Part II [Nau20] (top right) or God of war [SIE18] (bottom left), as well as the highly specular environments of Mirror's Edge Catalyst [DIC16] (bottom right), the needs to simulate complex light transports efficiently are very diverse.*

Figure 1.2: *From the large luminous landscapes of Red Dead Redemption II [Roc18] (left) to the small dark caves of Elden Ring [Fro22] (right), the scale of the light transports may be very different.*

with performance, in terms of number of operations but also in terms of organization of these operations to optimize their execution according to the hardware used. However, the notion of efficiency must also be crossed with the notion of quality. Indeed, offline rendering methods are often based on stochastic processes that need iteration to converge towards the final result. Thus, these methods are generally not suitable for rendering in video games since they tend to create noise until convergence is reached.

In addition, the need for efficiency is always growing. On the one hand, because of the use of 3D scenes that are more and more complex, in terms of their geometry but also because of the complexity of the light transports present in the scene. And on the other hand, the needed resolutions of the images also increases, increasing de facto the number of pixels to be computed. Also, some screens are not any more limited to 60 fps, going to 240 fps and even more. Thus, applications, such as competitive video games, require such efficiency. Moreover, why use costly methods that take several hours to produce an image if methods that run for only a few milliseconds produce a rendering with comparable quality. Especially in a context where energy saving is becoming a crucial point.

The key to efficiency in rendering is an adapted characterization of the light transport in a 3D scene. It is the variety of possible scene configurations that make this problem particularly complex. Thus, some rendering methods are only adapted to specific configurations (fog, smoke, water ...). The multiplication of specific method makes them particularly difficult to use without a thorough knowledge of the domain. In contrast, the methods proposed in this thesis are not specific to any particular configuration. They can be used from

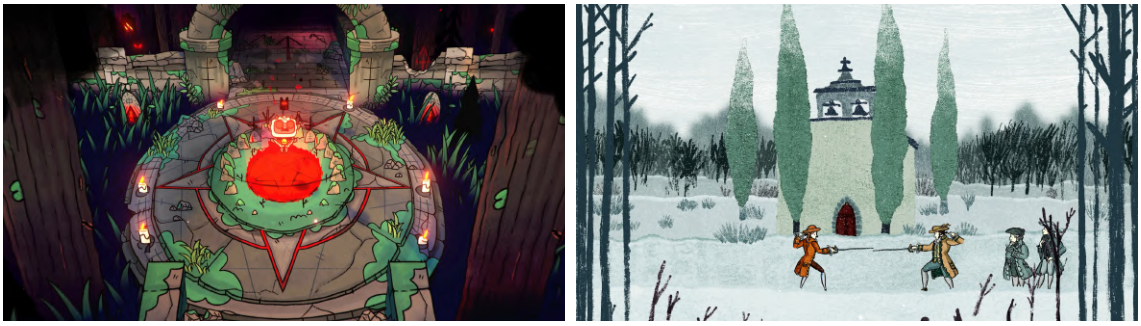


Figure 1.3: *Cult of the Lamb [Mas22] (left) and Card Shark [Ner22] (right) have renderings that are inspired by the laws of physics but modify them for artistic or scenaristic purposes. Such renderings are not the main focus of this thesis but can benefit from the work it proposes.*

Figure 1.4: *Halo 3 [Bun07, CL08] is one of the first notable game to deploy spherical harmonics rendering technique.*

indoor to outdoor scenes (Figure 1.1). As well as from large luminous landscapes to small dark caves (Figure 1.2). This thesis focuses on physically realistic rendering, rendering approaches that are inspired by physics but modify its behavior for artistic or scenaristic purposes (Figure 1.3). Thus, these rendering approaches are not the core of this thesis but can be inspired by this thesis to propose original content.

1.1 Spherical harmonics rendering

It is in this context of efficiency that the spherical harmonics (SH) have taken all their meaning in computer graphics. SH are orthogonal basis functions allowing to approximate any spherical function with a vector of coefficients. Each coefficient being attached to one SH. The orthogonality of the basis functions makes their use particularly interesting for rendering in computer graphics, we will see the mathematical reasons in Chapter 2. Moreover, the decomposition of a signal into SH is a frequency decomposition. Thus, the approximation of the decomposition increases in frequency when more SH are taken into account, thus increasing (quadratically) the size of the vector of coefficients. Any signal can be accurately represented because there is an infinite number of SH allowing to go up to any frequency, with an appropriate vector size. However, if the number of SH used is too small to represent the signal accurately, artifacts, analogous to the *Gibbs phenomenon*, appear that must be handled carefully. The SH, being a complete basis, encode the signal globally and thus do not to encode it only locally. Due to this property, diffuse lighting is particularly efficient to compute with SH, requiring only a few coefficients (9). It becomes more problematic for specular transport which requires a large number of coefficients, ranging from 25 for low specularity to several hundred coefficients for highly specular surfaces. Thanks to these properties, spherical harmonics were used very early in computer graphics [CMS87]. However, their most notable application come a little later with the method of *Precomputed Radiance Transfer* (PRT) [SKS02]. In this method, the spherical harmonics encode the light transport at each shaded points. The computation of the lighting is fast for diffuse surface but becomes expensive when specular surfaces are involved because of the number of SH needed. Nevertheless, it is especially the storage of the coefficients which becomes problematic, indeed the initial method stores the coefficients at each vertex of the mesh of the

Figure 1.5: *The cities at night of Cyberpunk 2077 [CD 20] (left) and Stray [Blu22] (right) have a special atmosphere with their hundred of area lights.*

scene, hence the storage becomes intractable with the actual 3D scenes with millions of vertices.

The initial formulation of PRT fixes the geometry of the scene as well as the materials at each vertex of the mesh. Multiple improvements have been proposed to tackle these limitations and a more in-depth introduction to the mathematics of spherical harmonics and their use in computer graphics is given in Chapter 2. In contrast, the work proposed in this thesis do not store any costly vector of coefficients to enable fully dynamics scenes (viewpoint, light, geometry and material) and in particular, takes advantage of the numerous recursive properties of the spherical harmonics to efficiently compute the SH coefficients encoding the light, where the length of the vector is arbitrary fixed. *Halo 3* is one of the first notable video games using spherical harmonics rendering (Figure 1.4). Despite their drawbacks, they are commonly used in today's games in particular to simulate the low frequency global illumination on dynamic objects [SIE18, Nau20, Bli22].

1.2 Applications and rendering techniques

Multiple configurations of 3D scenes and associated rendering techniques can be considered. Thus, this section gives a quick overview of the possible applications and rendering techniques considered in this thesis.

The high performance rendering context takes generally advantage of simple light primitives without areas, such as point lights. The light is emitted at a single point and thus light

Figure 1.6: *In video games, it is not rare to find exploration phases where the only source of lighting is a flashlight that the player controls. These situations appear in The Last of Us Part II [Nau20, Dog20] (left) and Uncharted 4: A Thief's End [Nau16] (right) where a special care is taken to the global illumination for an optimal immersion.*

received at any other point corresponds to a single direction. When using area light, the light is received in a measurable set of directions, which makes them more difficult to use efficiently. However, scene configurations based on this kind of light are numerous, *e.g.* the use of neon or billboard lights for urban night scenes is common (Figure 1.5). The multiplication of these light sources makes them more complex to implement in an interactive environment. Hence, in a first time, this thesis addresses a solution to simulate the direct lighting from a few dozen to a few hundred of spherical lights with real-time performance.

Real-time lighting algorithms are generally divided into two categories. One bounce illumination, where only the lighting of the primary sources of light is computed and the global illumination also computing the many bounces of lights that may occur in the scene. To generate the indirect lighting, a classically used method relies on the use of virtual sources, traditionally a set of *Virtual Point Lights* (VPL) distributed at the surface of the scene to capture the incident lighting and redistribute it to the shaded points visible to the camera. Thus, the indirect lighting is converted into a direct lighting problem with many virtual sources. These methods are classically used to simulate the global lighting produced by a flashlight because the VPLs are straightforward to distribute at the first light bounce (Figure 1.6). However, we will see in Chapter 5 that the gathering of the VPL lighting is not straightforward and must be done cautiously.

Another method reducing the complexity of the direct or indirect lighting computation is based on a set of probes distributed in the volume of the scene. These probes capture the illumination which is then interpolated at each shaded points within the volume covered by the probes. So instead of computing the lighting at each point of the scene, the lighting is computed only at a reduced number of points corresponding to the position of the probes. The efficiency of this method makes it particularly popular in competitive video games to generate the global lighting on dynamic objects (Figure 1.7).

So far, we have shown that the use of spherical harmonics, VPLs and probes are very common in video games, nevertheless film design and CAD also employ this kind of techniques (Figure 1.8). The work on global illumination proposed in this thesis seeks to efficiently combine the use of spherical harmonics with the VPL and also with the probes methods. In particular by seeking to take advantage of the work proposed on spherical lights.

Figure 1.7: *For dynamic object, global illumination in Tom Clancy's The Division [Mas16] (left) and Overwatch 2 [Bli22] (right) is captured by a set of probes, represented by white spheres, and the illumination is interpolated between a subset of probes closes to the shading point.*

Figure 1.8: *Rendering techniques discussed in this thesis are not only applied on video games but also for movies. The Na'vi from Avatar [Cam09] (left) is lighted with a spherical harmonics based method. Davy Jones from Pirates of the Caribbean: Dead Man's Chest [Ver06, Chr08] (right) is lighted with a point-based global illumination. The light emitted by the points is stored using spherical harmonics.*

1.3 Organization of this thesis and overview of the contributions

Before reviewing the literature on the use of spherical harmonics for rendering, it is necessary to introduce the main motivations for their use in computer graphics as well as to present the mathematical background of spherical harmonics (Chapter 2).

One of the major difficulties in using spherical harmonics for dynamic scenes is the on-the-fly SH projection of the radiance field to any illuminated point. Chapter 3 presents three distinct contributions enhancing the efficiency of the projection on spherical harmonics for spherical lights. The first contribution consists in an analytical and recursive computation of the projection of the radiance field produced by spherical light based on an efficient evaluation of the spherical harmonics. The second contribution is an analytical derivation of the evaluation of the spherical harmonics basis functions allowing to propose in a third contribution an analytical and recursive computation of the gradient of the projection of the radiance field produced by spherical light. We demonstrate the efficiency of this third contribution by computing the SH projection of the radiance field, with the gradient of the SH coefficients, only on a parsimonious subset of points volumetrically distributed in the scene. The gradient allows to interpolate accurately the SH coefficients representing the radiance field at each shading points. These contributions are motivated by recent works proposing similar contributions but adapted to polygonal lights [WR18, WCZR20].

However, the lighting that we perceive in a point is not simply the field of radiance that the point receives, but the result of the integral between the radiance field and the material on which the point is located. Depending on the use case, *e.g.* when the geometry of the scene is static, the convolution by spherical harmonics is straightforward to compute once the radiance field is projected onto the spherical harmonics. We designed a new framework to efficiently compute the convolution for dynamic scenes and with the use of materials commonly used in computer graphics that separates the diffuse and specular contribution (Chapter 4). Moreover, our methods for projecting the radiance field produced by spherical lights being more efficient than the proposed work for polygonal lights, we propose a method to approximate the projection of the radiance field produced by polygonal lights using our methods presented in the previous chapter.

Finally, Chapter 5 focuses on global lighting by combining the use of spherical harmonics

with point-based global lighting. We propose a new type of virtual source, called *Harmonics Virtual Light* (HVL), based on the methods of efficient projection of the radiance field produced by spherical lights proposed in Chapter 3. We also show that the combination of VPL and spherical harmonics can be particularly beneficial to efficiently compute the global radiance field received by probes volumetrically distributed in the scene. In order to introduce the global illumination methods used in this chapter, an overview of virtual lights and probe methods is presented at the beginning of this chapter.

1.4 Main results and publications

The work of this thesis has been presented in four distinct publications:

Projection efficace de lumières sphériques sur les harmoniques sphériques.
Pierre Mézières and Mathias Paulin.
Journées Françaises d'Informatique Graphique 2020. [MP20]

➥ National publication presenting an efficient SH projection of the radiance field produced by spherical lights (Section 3.1).

Efficient Spherical Harmonic Shading for Separable BRDF.
Pierre Mézières and Mathias Paulin,
Association for Computing Machinery 2021. [MP21]

➥ New framework to simulate efficiently the lighting with spherical harmonics with the use of materials that separate the diffuse and specular contribution (Section 4.1).

Harmonics Virtual Lights: fast projection of luminance field on spherical harmonics for efficient rendering.
Pierre Mézières, François Desrichard, David Vanderhaeghe and Mathias Paulin.
Computer Graphics Forum 2022. [MDVP22]

➥ Efficient point-based global illumination method (Section 5.2) based on an efficient SH projection of the radiance field produced by spherical lights (Section 3.1). The SH projection of the radiance field is the same as the one presented in the national publication (first publication of this list).

Recursive analytic spherical harmonics gradient for spherical lights.
Pierre Mézières, Nicolas Mellado, Loïc Barthe and Mathias Paulin.
Computer Graphics Forum 2022. [MMBP22]

➥ Efficient computation of the gradient of the SH coefficients representing the radiance field produced by spherical lights (Section 3.3). The method relies on an efficient computation of the gradient of the SH basis functions (Section 3.2). Similar methods focus on the radiance fields produced by polygonal lights [WR18, WCZR20], as our method is faster we propose a method to approach their radiance field using spherical lights (Section 4.2).

2 Lighting and Spherical Harmonics

This thesis focuses on physically based rendering. In order to allow a good reading of this document, this chapter introduces the technical notions of such rendering methods (Section 2.1). It turns out that the spherical harmonics are particularly well adapted to these methods, after an exposition of the mathematical background of the spherical harmonics (Section 2.2) a review of their use in computer graphics is presented (Section 2.3). Nevertheless, the spherical harmonics are not the only functions well suited for the rendering which have their own pros and cons compared to the spherical harmonics (Section 2.4).

2.1 Image synthesis

2.1.1 3D scene

This thesis focuses on the efficient simulation of light transport in a 3D scene. A 3D scene is based on four components which are the lights, the geometry, the materials and the sensors.

Lights Light is a form of electromagnetic wave that propagates through space and can be modeled as light rays when observed in a transparent medium. Light rays are in fact a macroscopic manifestation of electromagnetic waves and their behavior can be described using the laws of physics that govern electromagnetic waves. The light rays emitted by the sources will bounce on the scene to be finally caught and contribute to an image and make the objects visible. Thus, a point in space can receive light in all directions. This introduces a need to efficiently characterize the directional distribution of light. Spherical harmonics provide solutions to this problem. There are multiple ways to define light sources. For instance, for outdoor environments the sun can be modeled by an infinitely far light whose light rays reach the scene in parallel. To simulate distant environment lighting, environment maps are precomputed and store the light coming from all possible directions. In indoor environment, the use of infinitely small, area-less lights, are common due to their simplicity of use, for the design and ease of computation. Surface lights are more difficult to set up for the same two reasons but bring a more important physical coherence. However, without anything to light, the simulation is useless.

Geometry The geometry represents the shape of the objects. This information can be stored and rendered in multiple ways, such as triangular meshes, quadrangular meshes,

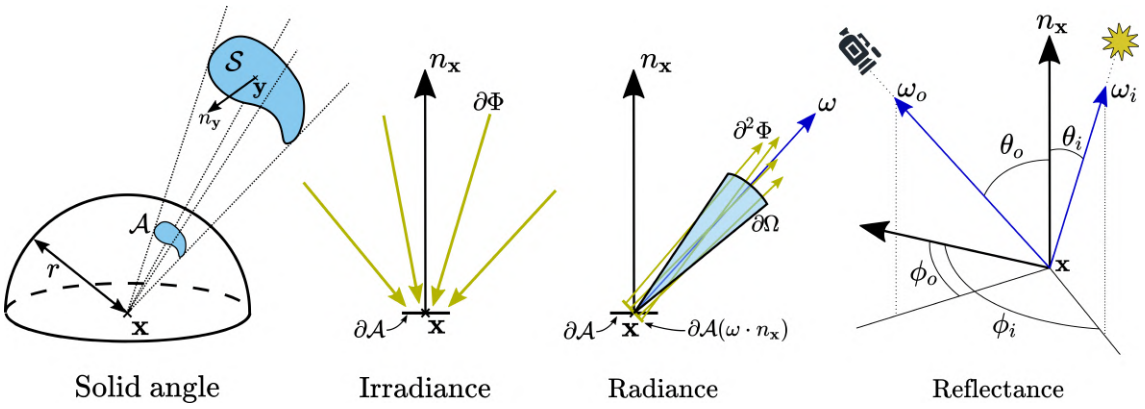


Figure 2.1: *Visualization of different quantities of the rendering. Flux and intensity are respectively similar to irradiance and radiance but without the surface element $\partial\mathcal{A}$ and can be visualized in an analogous way.*

implicit surfaces or voxels. However, graphics cards are specially designed to work with triangular meshes and this is the most common form of shape representation used in rendering. Each vertex of the mesh represents a point in space whose coordinates must be stored. Since rendering methods like PRT, precompute information per vertex, such as a color or visibility information of nearby vertices. This introduces an additional amount of information that becomes non-negligible for complex scenes with several thousand vertices. SH rendering approaches store information per vertex and are therefore confronted to this problem. However, without materials applied on the geometry, the simulation of the light transport is meaningless.

Materials The materials transform the incident radiance field into an outgoing radiance field depending on the nature of the material. Hence, as for light, this introduces a need for efficient description of directional distributions. The material can define the way in which an object will reflect, but also its refraction or transmission that will be in the object, as in a cloud or smoke. Materials can thus define a variety of objects ranging from diffuse opaque material, such as a sheet of paper to transparent and highly specular materials, such as iridescent bubbles.

Sensor The role of the sensor is to capture light rays to generate an image. The sensor can be seen as a virtual camera, composed of an optical system and an image plane. The use of complex optical system allows simulating various optical effects, ranging from depth of field to temporal blur when the notion of time is involved. Like the majority of the real-time rendering methods, this thesis only uses a pinhole camera model.

2.1.2 Rendering quantities

Physically based rendering relies on several quantities, including radiometric quantities (Figure 2.1). It is necessary to introduce them before formalizing the light transports equations. However, depending on the discipline, the definition of these quantities may slightly vary.

Solid angle The solid angle Ω gives the amount of field of view covered by the projection of an object at a point, *i.e.* the area covered by the projection of an object on the unit sphere,

therefore going from 0 to 4π. The unit of Ω is the *steradian* (sr) and given by

$$\Omega = \frac{\mathcal{A}}{r^2} , \tag{2.1}$$

where $\mathcal{A}$ is the area of the projected surface $\mathcal{S}$ on the sphere of radius r. In general, the solid angle is considered by default for a unit sphere ($r = 1$). The solid angle is then defined by the integration of the constant 1 for all directions ω that lie in the area $\mathcal{A}$:

$$\Omega = \int_{\mathcal{A}} \mathrm{d}\omega \ . \tag{2.2}$$

The solid angle can also be defined according to the elementary solid angle $\mathrm{d}\omega_{\mathbf{y}}$, *i.e.* considering the points $\mathbf{y}$ (and their normal $n_{\mathbf{y}}$) lying on the surface $\mathcal{S}$, the elementary solid angle is given by the ratio of the area projected at $\mathbf{x}$ and the square of the distance:

$$\Omega = \int_{\mathbf{y}\in\mathcal{S}} \mathrm{d}\omega_{\mathbf{y}} \quad \text{with} \quad \mathrm{d}\omega_{\mathbf{y}} = \frac{(n_{\mathbf{y}} \cdot \vec{\mathbf{xy}})\, \mathrm{d}\mathcal{S}_{\mathbf{y}}}{\|\mathbf{x} - \mathbf{y}\|^2} , \tag{2.3}$$

where $\mathrm{d}\mathcal{S}_{\mathbf{y}}$ is an infinitesimal surface element of $\mathcal{S}$. For simplicity, the integral on the unit sphere is denoted by $\int_\Omega \mathrm{d}\omega$. In such case, the solid angle is equal to 4π and 2π for the unit hemisphere. For the characterization of the directions, it is particularly useful to define the triplet (r, θ, ϕ) corresponding to the spherical coordinates (Figure 2.1). The conversion between Cartesian and spherical coordinates is defined by

$$\begin{cases} x = r\cos\phi\sin\theta \\ y = r\sin\phi\sin\theta \\ z = r\cos\theta \end{cases} . \tag{2.4}$$

The use of θ and ϕ alone is sufficient to characterize the set of directions, so the integral on the sphere can be rewritten by

$$\int_\Omega \mathrm{d}\omega = \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \sin\theta \, \mathrm{d}\theta \, \mathrm{d}\phi = 4\pi \tag{2.5}$$

where $\sin\theta$ is the Jacobian of the contraction of the spherical coordinates at the poles. Indeed, considering small solid angles defined by bounding θ and ϕ, for the same boundary intervals this solid angle will tend to 0 when θ tends to 0 (or π).

Spherical cap generalizes the notion of sphere and hemisphere. The spherical cap is defined by a direction ω and the set of directions that form an angle with ω smaller than a chosen θ. ω is the central direction of the cap and θ is its half angle. The solid angle of the cap is given by $\Omega = 2\pi(1 - \cos\theta)$.

Flux and intensity The radiant flux is the radiant energy emitted or received per unit of time through any surface

$$\Phi = \frac{\partial Q}{\partial t} \tag{2.6}$$

where Q is the energy quantity and then the flux is given in *Watt* (W). For rendering, the considered surface is usually the sphere to define the energy arriving or leaving its center.

The intensity is a directional quantity defined by the radiant flux per unit of solid angle. The unit of solid angle can be seen as a small spherical cap centered on ω

$$I(\omega) = \frac{\partial \Phi}{\partial \Omega} . \tag{2.7}$$

The unit of the intensity is thus $W.sr^{-1}$. Intensity therefore describes the light that arrives or leaves a point in a specific direction.

Irradiance and radiance The irradiance and radiance can be respectively seen as an extension of the flux and intensity with surface elements. Thus, the irradiance is the radiant flux per unit of area of an infinitesimal surface element around the point $\mathbf{x}$:

$$E(\mathbf{x}) = \frac{\partial \Phi}{\partial \mathcal{A}} \,. \tag{2.8}$$

The unit of irradiance is $W.m^{-2}$ and thus characterizes the total light received on a surface element. Radiosity follows the same principle as the irradiance but for the emitted light.

The radiance is the radiant flux per unit of projected area of an infinitesimal surface element around $\mathbf{x}$ and per unit of solid angle

$$L(\mathbf{x}, \omega) = \frac{\partial^2 \Phi}{\partial \Omega \partial \mathcal{A} \, |\omega \cdot n_{\mathbf{x}}|} \tag{2.9}$$

where $n_{\mathbf{x}}$ is the surface normal at $\mathbf{x}$ and $\mathcal{A}(\omega \cdot n_{\mathbf{x}})$ is the projected area. Its unit is $W.m^{-2}sr^{-1}$. Radiance thus characterizes the directional light received or emitted on a surface element. Considering $n_{\mathbf{x}}$ as the surface normal at $\mathbf{x}$, the irradiance is related to the radiance by the following equation

$$E(\mathbf{x}) = \int_{\Omega} L(\mathbf{x}, \omega) \, |\omega \cdot n_{\mathbf{x}}| \, \mathrm{d}\omega \,. \tag{2.10}$$

This means that the irradiance can be deduced from the radiance function. Spherical harmonics allow this deduction in a particularly efficient way [RH01].

Reflectance The reflectance is the function describing the reflective behavior of a material. This function is usually called the *Bidirectional Reflectance Distribution Function* (BRDF). For a point $\mathbf{x}$ on the object, this function is 4D, and yields the amoung of light reflected from an input direction ω_i to an output direction ω_o. BRDF is thus defined as the outgoing radiance per unit of irradiance received in the direction ω_i

$$F(\mathbf{x}, \omega_i, \omega_o) = \frac{\partial L_o(\omega_o)}{\partial E_i(\omega_i)} = \frac{\partial L_o(\omega_o)}{L_i(\omega_i)(\omega_i \cdot n_{\mathbf{x}}) \partial \omega_i} \,. \tag{2.11}$$

The unit of irradiance received in the direction ω_i corresponds to the projected radiance in the same direction. The unit of the reflectance is then sr^{-1}. Reflectance, like any function, can be stored and reconstructed analytically or with tabulated measures. Tabulated measures correspond to a sampling of the function associated to a reconstruction method. Databases offer dozens of tabulated materials [MPBM03, DHB17]. In contrast, analytical materials are defined in the form of an analytical expression, usually fast to compute on the fly and therefore particularly suitable for real-time rendering. *Decomposable BRDF* consider two parts, the diffuse and the specular components. Both components have a distinct reflectance lobes, where the diffuse corresponds to a Lambertian reflectance and the specular can be simulated by a cosine lobe, for its simplest form in the Blinn-Phong model, or by more complex forms for a more accurate physical simulation, such as the GGX model [WMLT07]. To improve the physical accuracy, additional terms are added to the equation, *e.g.* the Fresnel coefficient. Furthermore, physical based materials follow additional properties that are

- positivity: $F(\mathbf{x}, \omega_i, \omega_o) \geq 0$
- Helmholtz reciprocity: $F(\mathbf{x}, \omega_i, \omega_o) = F(\mathbf{x}, \omega_o, \omega_i)$
- energy conservation: $\forall \omega_o, \int_{\Omega} F(\mathbf{x}, \omega_i, \omega_o)(\omega_i \cdot n_{\mathbf{x}}) \, \mathrm{d}\omega_i \leq 1$

(2.12)

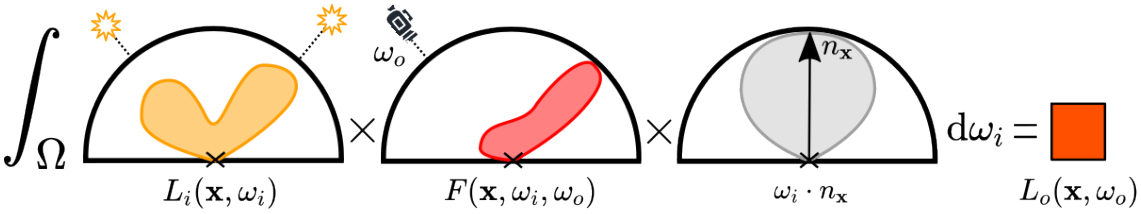


Figure 2.2: *Visualization of the integral components of the rendering equation (Equation 2.13).*

Defining a sphere whose north pole is fixed by the surface normal, the reflection is defined in the northern hemisphere and the refraction is defined in the southern hemisphere. Thus, the BRDF is defined only in the northern hemisphere, the refraction is modeled by the *Bidirectional Transmittance Distribution Function* (BTDF). Both are included in the *Bidirectional Scattering Distribution Function* (BSDF). Thus, the BxDF, referring to the three methods, are similar and define the light transfer between two directions. This thesis focuses on BRDF and the generalization to BxDFs is straightforward. However, these functions can be extended to take more parameters into account, like time or wavelength. However, these more complex materials are out of scope of this thesis.

2.1.3 Rendering equation

The rendering equation [Kaj86] is an energy balance at the surface of an object, it relates the incoming radiance to the outgoing radiance at a point $\mathbf{x}$. The formalization of light transport is based only on the rendering equation when the scene is immersed in vacuum. To simulate any atmospheric effects, it is necessary to take into account the equations of radiative transfer but this is beyond the scope of this thesis. Moreover, this thesis focuses on the stationary and monochromatic rendering equation, *i.e.* without time and wavelength variation.

Directional formulation The rendering equation expresses the outgoing radiance at one point and direction $L_o(\mathbf{x}, \omega_o)$ as a function of the incoming radiance from all possible directions (Figure 2.2 and 2.3)

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_\Omega L_i(\mathbf{x}, \omega_i) F(\mathbf{x}, \omega_i, \omega_o) \, |\omega_i \cdot n_\mathbf{x}| \, \mathrm{d}\omega_i \tag{2.13}$$

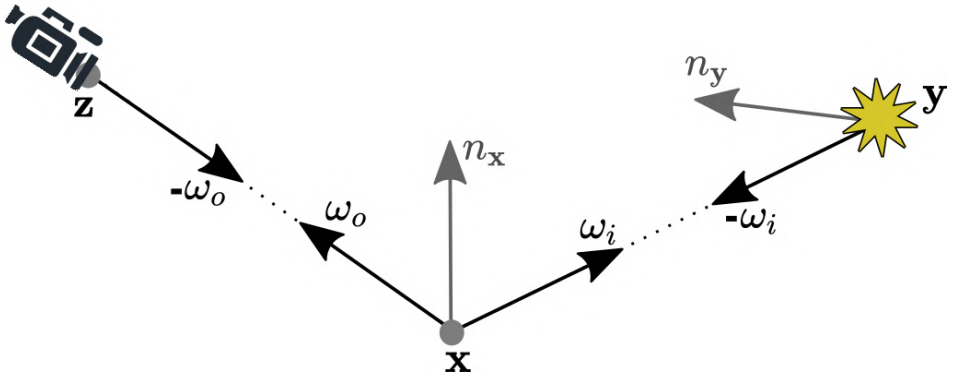


Figure 2.3: *The rendering equation can be written as a function of the input and output directions, ω_i and ω_o (Equation 2.13) but also as function of the points on the surface $\mathbf{x}, \mathbf{y}, \mathbf{z}$ (Equation 2.16) where the n vectors are the surface normals. The point $\mathbf{y}$ can be seen as a light and $\mathbf{z}$ as a sensor.*

where:

- $L_e(\mathbf{x}, \omega_o)$ is the emitted radiance at $\mathbf{x}$ in the direction ω_o (for instance when $\mathbf{x}$ is on a light),
- $L_i(\mathbf{x}, \omega_i)$ is the ingoing radiance at $\mathbf{x}$ from direction ω_i,
- $F(\mathbf{x}, \omega_i, \omega_o)$ is the reflectance function describing the energy transfer between the two directions ω_i and ω_o,
- $(\omega_i \cdot n_\mathbf{x})$ is a weakening factor to take into account that the area of a surface projected perpendicular to the direction ω_i is less than its area, where $n_\mathbf{x}$ is the surface normal. This factor is the cosine of the angle formed between ω_i and $n_\mathbf{x}$,
- $\int_\Omega [\cdots] \, \mathrm{d}\omega_i$ is an integral over the spherical domain Ω.

When the incident radiance L_i arriving to $\mathbf{x}$ is only the radiance emitted from the primary light sources, this case is called the *direct lighting*. The *global illumination* is when the incident radiance is also coming from the other points of the scene according to the radiance that they have received themselves. The light rays start from a primary light, then bounce in the scene before reaching the point $\mathbf{x}$ considered. Thus, the rendering equation is defined recursively between all possible points of the scene. Defining $t(\mathbf{x}, \omega_i)$ as a function that computes the first point $\mathbf{x}'$ hit by the ray leaving $\mathbf{x}$ in the direction ω_i. Without scattering in the atmosphere, the radiance arriving at $\mathbf{x}$ in the direction ω_i is the same radiance as the one leaving $\mathbf{x}'$ in direction $-\omega_i$

$$L_i(\mathbf{x}, \omega_i) = L_o(t(\mathbf{x}, \omega_i), -\omega_i) \; . \tag{2.14}$$

Managing the global illumination is the main complexity in solving the rendering equation. Find the points from which is emitted the incident radiance L_i arriving from any direction ω_i is also a source of complexity. The points are straightforward to compute using a ray-tracing based solution [WRC88]. However, other methods avoid this problem by reformulating the rendering equation as a function of the surface of the scene, which directly yields the set of points from which is emitted the incident radiance arriving at $\mathbf{x}$ [SAWG91].

Surface formulation The rendering equation can be also expressed in terms of the surface of the scene $\mathcal{S}$ where the ingoing, outgoing radiance and the reflectance can be written with points on the surface $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ instead of directions

$$L(\mathbf{y} \to \mathbf{x}) = L_o(\mathbf{y}, \vec{\mathbf{yx}}) = L_i(\mathbf{x}, \vec{\mathbf{xy}}) \qquad \text{and} \qquad F(\mathbf{y} \leftrightarrow \mathbf{x} \leftrightarrow \mathbf{z}) = F(\mathbf{x}, \vec{\mathbf{xy}}, \vec{\mathbf{xz}}) \tag{2.15}$$

where the vectors are all normalized. The rendering equation becomes

$$L(\mathbf{x} \to \mathbf{z}) = L_e(\mathbf{x} \to \mathbf{z}) + \int_\mathcal{S} L(\mathbf{y} \to \mathbf{x}) F(\mathbf{y} \leftrightarrow \mathbf{x} \leftrightarrow \mathbf{z}) V(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathcal{S}_\mathbf{y} \; . \tag{2.16}$$

where $V(\mathbf{x}, \mathbf{y})$ is a binary visibility function yielding 0 if an obstacle lies between $\mathbf{x}$ and $\mathbf{y}$, and 1 otherwise. The geometric factor $G(\mathbf{x}, \mathbf{y})$ multiplied by $\mathrm{d}\mathcal{S}_\mathbf{y}$ is related to the elementary solid angle $\mathrm{d}\omega_\mathbf{y}$ (Equation 2.3), *i.e.* the solid angle of the small surface element around $\mathbf{y}$ and seen from $\mathbf{x}$ multiplied by the weakening factor of the directional rendering equation (Equation 2.13)

$$\mathrm{d}\omega_\mathbf{y} = \frac{(\vec{\mathbf{xy}} \cdot n_\mathbf{x})(\vec{\mathbf{yx}} \cdot n_\mathbf{y})}{\|\mathbf{x} - \mathbf{y}\|^2} \, \mathrm{d}\mathcal{S}_\mathbf{y} = G(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathcal{S}_\mathbf{y} \; . \tag{2.17}$$

The surface formulation of the rendering equation is particularly useful for approaches based on *Virtual Point Lights* (VPL). Indeed, each point $\mathbf{y}$ plays the role of a virtual source

whose emission depends on the radiance their receive. Hence, a set of virtual sources distributed on the scene allows to approximate the rendering equation on each point $\mathbf{x}$. However, in this case, sampling Equation 2.16 is problematic due to the division by the squared distance in the geometric factor, producing artifacts when $\mathbf{x}$ and $\mathbf{y}$ are close. These are not the only artifacts produced by VPLs and they are discussed further in Chapter 5.

2.1.4 Monte-Carlo integration

The resolution of the rendering equation is based on the resolution of an integral defined on the spherical domain. Monte-Carlo methods are classically used to compute integrals numerically.

General Monte-Carlo Considering the space Ω, of any dimension, general Monte-Carlo methods allow to compute an estimator of the integral of any function f on Ω.

$$F = \int_{\Omega} f(x)\,\mathrm{d}x \tag{2.18}$$

Monte-Carlo methods distribute N points in Ω to estimate the integral, whose resolution depends on the way the points are distributed. Thus, a Monte-Carlo estimator is written as

$$\hat{F} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)} \quad \text{with} \quad \lim_{N\to\infty}\hat{F} = F \tag{2.19}$$

where p is the *Probability Density Function* (PDF) describing the way the samples x_i are distributed. Thus, the larger N is, the more $\hat{F}$ will converge to F, at the convergence rate of $\sqrt{N}$. For the convergence to happen sooner, the choice of the PDF is crucial but relies on an a priori knowledge of the function to integrate.

Quasi Monte-Carlo and quadrature General Monte-Carlo methods rely on a random drawing of the samples, whereas quasi Monte-Carlo relies on sequences with low discrepancy. Low discrepancy means that the N samples are distributed as evenly as possible. Thus, these methods have a convergence rate of N instead of $\sqrt{N}$ for general Monte-Carlo. However, this approach is based on deterministic algorithms carefully designed.

The general principle of the quadrature is based on methods that are simpler to implement. An integral can be seen as a sum of Riemann, in a 1D case, the area under the function can be seen as a set of rectangles, where the approximation of the area becomes better when the number of rectangles used is large. Thus, when the rectangles have the same width, *i.e.* the samples are evenly distributed, this method is particularly simple to implement and offers a convergence of the integral computation sufficient for many applications. However, unlike general or quasi Monte-Carlo, their generalization to any dimension is complex. An integral on the spherical domain can be performed with the *spherical Fibonacci mapping* [KISS15] that allows a good approximation of an even distribution of the samples on the sphere.

2.1.5 Orthogonal basis functions

Orthogonal basis functions are a mathematical tool allowing to represent a function by a vector of coefficients where the i-th coefficient $\mathbf{S}_i$ is attached to the i-th function of the base B_i. The function is projected onto each of the basis functions to compute the coefficients (Figure 2.4 - top).

$$\mathbf{S}_i = \langle S, B_i \rangle = \int S(x)B_i(x)\,\mathrm{d}x \tag{2.20}$$

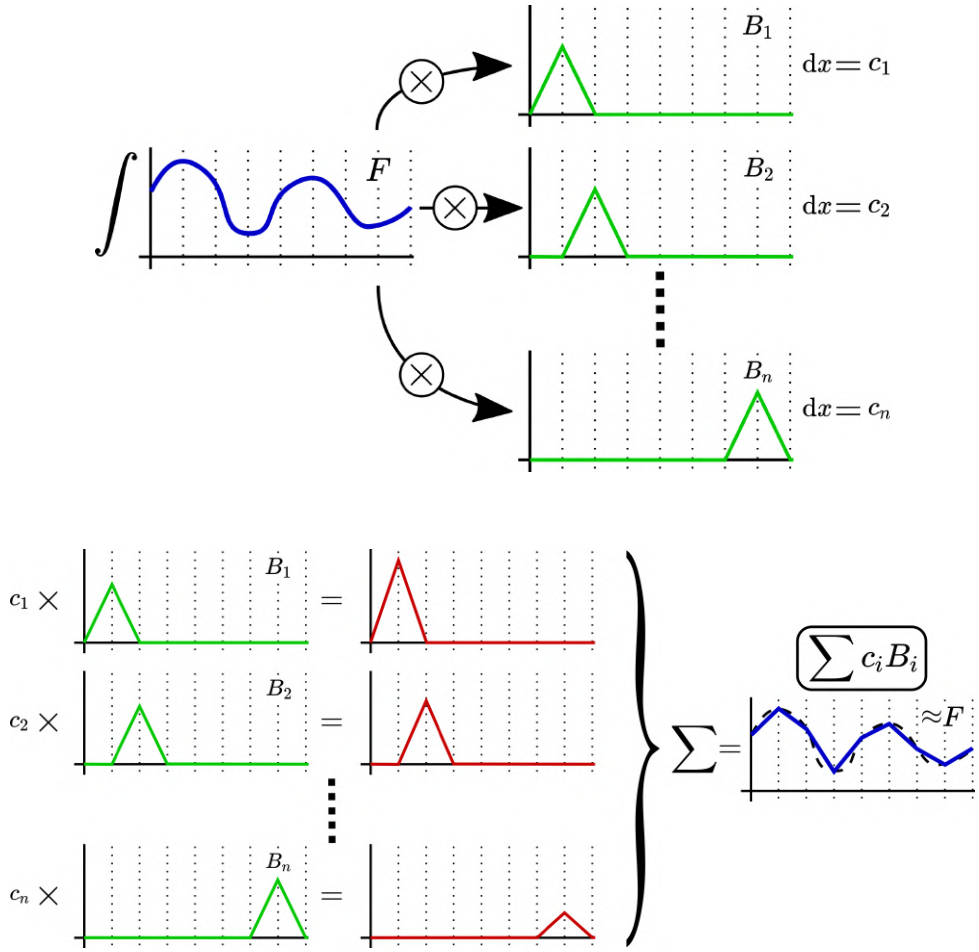


Figure 2.4: *The projection (top) of a function (blue) on basis functions (green) yields a coefficient by basis function (c_i). The reconstruction (bottom) of the function consists of scaling the basis function (red) by the coefficients and summing them up to compute an approximation of the function, whose quality depends on the basis functions.*

where $\langle , \rangle$ is the inner product that generalizes the dot product, which computes the projection of a vector on another, to continuous functions. Then the function is reconstructed by summing the multiplication of the coefficients with the basis function (Figure 2.4 - bottom).

$$S(x) \approx \sum_i \mathbf{S}_i B_i(x)\,\mathrm{d}x \tag{2.21}$$

The reconstruction is generally an approximation but can be an exact reconstruction depending on the basis functions used. Some basis have functions that are orthogonal to each other, these function bases follow the following property.

$$\int B_i(x) B_j(x)\,\mathrm{d}x = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{2.22}$$

This property allows simplifying the integral of the product of two functions projected beforehand on the basis functions

$$\begin{aligned}\int L(x)F(x)\,\mathrm{d}x &\approx \int \Big(\sum_i \mathbf{L}_i B_i(\omega)\Big)\Big(\sum_j \mathbf{F}_j B_j(x)\Big)\,\mathrm{d}x \\ &\approx \sum_i \sum_j \mathbf{L}_i\mathbf{F}_j \int B_i(x)B_j(x)\,\mathrm{d}x \\ &\approx \sum_i \mathbf{L}_i\mathbf{F}_i\ .\end{aligned} \tag{2.23}$$

Thus, the integral is simplified by a scalar product of the coefficient vectors of each function. This property works with any number of dimensions and thus works for a spherical domain defined in 2D (θ, ϕ). The choice to use the radiance L and the reflectance F is not innocent, since the similarity between the Equation 2.23 and the rendering equation (Equation 2.13) is obvious. As indicated in their name, the spherical harmonics are spherical functions that are orthogonal to each other. They are therefore very suitable to simplify the integral of the rendering equation by a simple scalar product of the coefficients vectors. Moreover, Monte-Carlo methods are known to make noise as long as the number of samples is not sufficient and this number can be very high. Solving the integral by using spherical harmonics will avoid this problem. However, if the integral of the rendering equation is avoided, the projection integral (Equation 2.20) becomes a problem. Solutions to compute an efficient projection on spherical harmonics are addressed in this thesis.

2.2 Mathematics of Spherical Harmonics

Spherical harmonics are basis functions [Hob31], defined first by their complex form. Real spherical harmonics, particularly used in computer graphics, are derived from their complex form. Thus, many of their properties, in particular the convolution and rotation, defined for their complex form have been adapted to their real form. Although their complex form can be very useful and it is important to mention them in order to fully understand some properties of real spherical harmonics, the efficient use of the complex spherical harmonics is out of the scope of this thesis. A significant amount of the information developed in this chapter can be found in the papers written by Green [Gre03] and Sloan [Slo08].

2.2.1 Spherical harmonics and notations

Spherical Harmonics (Figure 2.5) form an orthogonal basis function where each function is indexed by an order l and a degree m. SH are analogous to Fourier series, decomposition in sines and cosines but defined on a sphere. The SH are ordered as: the larger the order l is, the higher the angular frequencies. More specifically, the order l defines the number of oscillations along θ and the degree m along ϕ. The order l can go to infinity and the degree m varies between $-l$ and l, thus forming a pyramid of functions.

Spherical Harmonics take their name from the Laplace's equation. Indeed, the functions that verify the Laplace's equation are called harmonics, *i.e.* the divergence of the gradients for any point of the surface of the function is equal to zero. The Laplace's equation is verified at any point of the surface of the graph of each SH (Figure 2.5). They are spherical because defined on the unit sphere and using spherical coordinates, Complex SH $\mathcal{Y}_l^m$ are expressed as

$$\mathcal{Y}_l^m(\theta,\phi) = (-1)^m K_l^m \mathcal{P}_l^m(\cos\theta)e^{im\phi} \tag{2.24}$$

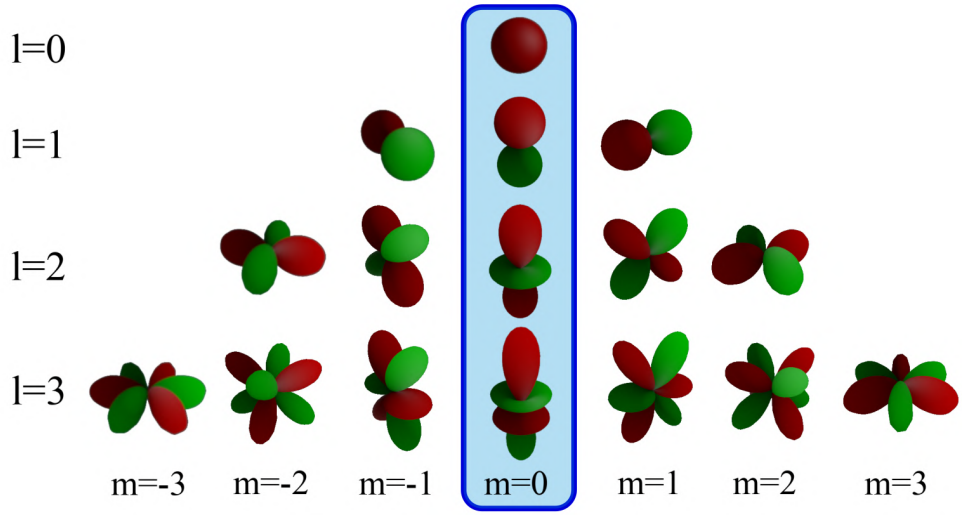


Figure 2.5: *Representation of the first real spherical harmonics basis functions. Lobes are colored according to their sign, with positive values in red and negative in green. The sub-basis of zonal harmonics for which $m = 0$ is outlined in the middle.*

where $\mathcal{P}_l^m$ denotes the associated Legendre polynomials (ALP), and K_l^m is a normalization factor:

$$K_l^m = \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}\,. \tag{2.25}$$

The consideration of the Condon–Shortley phase $(-1)^m$ is often a source of confusion depending on the field of study, sometimes included in the SH in quantum physics and other times directly in the ALP as in computer graphics. In this thesis, the phase $(-1)^m$ is integrated in the ALP which are then defined by

$$P_l^m(\cos\theta) = (-1)^m \mathcal{P}_l^m(\cos\theta) = \sin^m\theta \frac{d^m}{dx^m}(P_l(\cos\theta)) \tag{2.26}$$

where P_l are the Legendre Polynomials (LP) defined with the Condon-Shortley phase. Recurrence relations allow to evaluate the ALP (and LP) efficiently

$$(l-m)P_l^m(x) = x(2l-1)P_{l-1}^m(x) - (l+m-1)P_{l-2}^m(x) \tag{2.27}$$

$$P_m^m(x) = (-1)^m(2m-1)!!(1-x^2)^{m/2} \tag{2.28}$$

$$P_{m+1}^m(x) = x(2m+1)P_m^m \tag{2.29}$$

where $P_0^0(x) = 1$, $P_0^1(x) = x$ and $x!!$ is the double factorial that return the product of the odd integers less than or equal to x. Hence, these relations allow to compute efficiently the ALP per degree m. Equation 2.28 computes the first term of degree m, Equation 2.29 computes the second and Equation 2.27 computes the other.

In computer graphics, the use of complex functions is uncommon, it is then more convenient to use the real SH Y build from the complex SH $\mathcal{Y}$.

$$Y_l^m(\theta,\phi) = \begin{cases} \sqrt{2}\,\mathfrak{Im}[\mathcal{Y}_l^{|m|}] & m<0 \\ \mathcal{Y}_l^0 & m=0 \\ \sqrt{2}\,\mathfrak{Re}[\mathcal{Y}_l^m] & m>0 \end{cases} = \begin{cases} \sqrt{2}K_l^{|m|}P_l^{|m|}(\cos\theta)\sin(|m|\phi) & m<0 \\ K_l^0 P_l^0(\cos\theta) & m=0 \\ \sqrt{2}K_l^m P_l^m(\cos\theta)\cos(m\phi) & m>0 \end{cases} \tag{2.30}$$

where $\sqrt{2}$ is a normalization factor to preserve the orthonormality of the basis function, and hence maintain the following property

$$\int_\Omega Y_i(\omega) Y_j(\omega)\,\mathrm{d}\omega = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{2.31}$$

with δ_{ij} the Kronecker delta and the index i and j corresponds to a linearization of the order l and the degree m.

Notations

The SH are defined by the order l and the degree m, forming a pyramid of function (Figure 2.5). However, these index can be linearized, thus facilitating they indexation.

$$\begin{gathered} i = l(l+1) + m \\ Y_l^m(\omega) = Y_i(\omega) = Y_l^m(\theta,\phi) = Y_i(\theta,\phi) \end{gathered} \tag{2.32}$$

The same principle applies for the coefficient vectors of the projected function. Using the index i allows to simplify the notations on equations using SH, nevertheless, the use of indices l and m are sometimes necessary. Thus, the two notations will be used appropriately.

Any spherical function can be approximated by a linear combination of the SH basis function, encoded as a vector of coefficients (Section 2.2.3). Thanks to the orthogonality of the SH basis function, any linear operator that applies on the basis function also applies on the coefficients vector (*e.g.* the convolution or the rotation (Section 2.2.10 and 2.2.12)). The SH of any order and degree are naturally defined with spherical coordinates. Nevertheless, polynomial forms with Cartesian coordinates can be derived for each SH that are particularly convenient to use when a low number of SH bands is involved (*e.g.* 3 SH bands). However, Sloan [Slo13] shows an efficient computation method of the SH basis function based on their definition with spherical coordinates that we leverage to compute the gradient of the SH basis function. The Sloan's method and our improvement are detailed in Section 3.2.

2.2.2 Hemispherical harmonics

Gautron *et al.* [GKPB04] define the hemispherical harmonics (HSH) (Figure 2.6) by shifting the associated Legendre polynomials so that they are defined only for the northern hemisphere $\theta \in [0, \pi/2]$

$$H_l^m(\theta,\phi) = \begin{cases} \sqrt{2}\tilde{K}_l^{|m|}\, P_l^{|m|}(2\cos\theta - 1)\,\sin(|m|\phi) & m < 0 \\ \tilde{K}_l^0\, P_l^0(\cos\theta) & m = 0 \\ \sqrt{2}\tilde{K}_l^m\, P_l^m(2\cos\theta - 1)\,\cos(m\phi) & m > 0 \end{cases}. \tag{2.33}$$

Thus, the normalization factor is redefined for the hemisphere

$$\tilde{K}_l^m = \sqrt{\frac{2l+1}{2\pi}\frac{(l-m)!}{(l+m)!}}. \tag{2.34}$$

As they are concentrated on the hemisphere, hemispherical harmonics need less functions to represent hemispheric functions compared to spherical harmonics. It is therefore particularly useful to represent the radiance and its transfer on an opaque surface, since only the

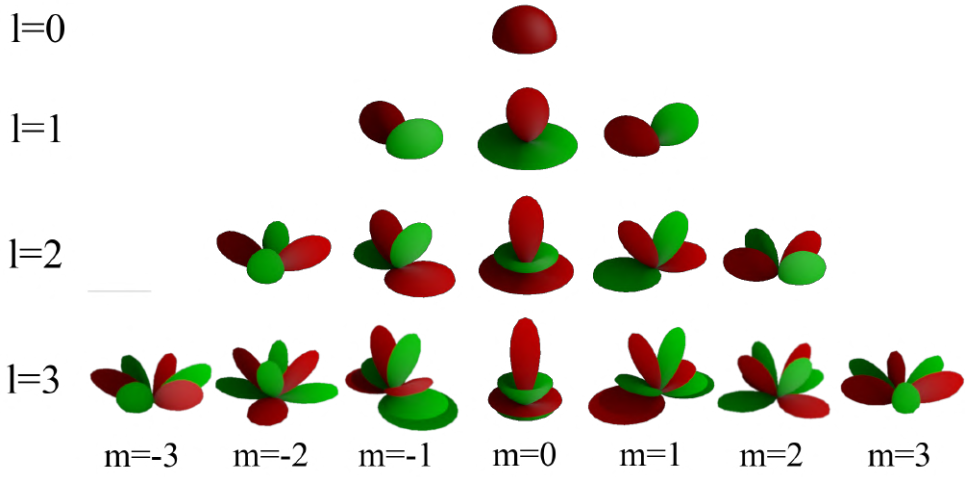


Figure 2.6: *Representation of the first real "hemispherical harmonics" basis functions [GKPB04]. Lobes are colored according to their sign, with positive values in red and negative in green.*

hemisphere directed according to the normal of the surface contains useful information. A matrix transfer $\mathbf{M}$ can be defined to convert the vector of spherical harmonics coefficients $\mathbf{Y}$ to the vector of hemispherical harmonics coefficients $\mathbf{H}$, and inversely, by projecting each HSH to each SH

$$\mathbf{M}_{ij} = \int_{\Omega} H_i(\omega) Y_j(\omega)\,\mathrm{d}\omega \qquad \rightarrow \qquad \mathbf{H} = \mathbf{Y}\mathbf{M}\,. \tag{2.35}$$

Gautron *et al.* [GKPB04] show that $\mathbf{M}$ is very sparse but it still adds a computation step when the two basis are used in same time, *e.g.* when using a spherical environment map projected on SH and a hemispherical reflectance function projected on HSH. This thesis is not limited to hemispherical functions and focuses on the use of spherical functions and thus on an efficient use of spherical harmonics.

2.2.3 Projection and reconstruction

Projection to and reconstruction from SH of the spherical function $S(\omega)$ obey the same rules as any basis function (Section 2.1.5) but adapted to the spherical domain:

$$\mathbf{S}_l^m = \langle S, Y_l^m \rangle = \int_{\Omega} S(\omega)\, Y_l^m(\omega)\,\mathrm{d}\omega \tag{2.36}$$

$$S(\omega) = \sum_{l=0}^{+\infty} \sum_{m=-l}^{l} \mathbf{S}_l^m\, Y_l^m(\omega) \approx \sum_{l=0}^{n} \sum_{m=-l}^{l} \mathbf{S}_l^m\, Y_l^m(\omega) \tag{2.37}$$

where $\mathbf{S}$ is the vector of the SH coefficients. In practice, SH coefficients are bands limited to $l = n$ and the reconstruction becomes an approximation with a fixed maximum frequency represented by the band n, thus using $N = n^2$ coefficients. The energy captured by the SH is equal to the sum of the square of the coefficients

$$E_s = \sum_{l=0}^{n} \sum_{m=-l}^{l} (\mathbf{S}_l^m)^2\,. \tag{2.38}$$

Therefore, the lost energy (approximation error) is equal to $\sum_{l=n+1}^{\infty} \sum_{m=-l}^{l} (\mathbf{S}_l^m)^2$.

Notations

In order to simplify the notations, when the double sum of the reconstruction is not necessary (Equation 2.37) the reconstruction can be noted

$$\sum_{l=0}^{n}\sum_{m=-l}^{l}\mathbf{S}_l^m\,Y_l^m(\omega)=\sum_{l,m}^{N}\mathbf{S}_l^m\,Y_l^m(\omega)=\sum_{i}^{N}\mathbf{S}_i\,Y_i(\omega) \tag{2.39}$$

where the last term use the linearization of the indices (Equation 2.32). The same notation but without the n (or N) indicates that the reconstruction uses an infinity of SH, *i.e.* l goes to infinity.

The integral of the projection (Equation 2.36) can be solved by multiple methods, analytically, with Monte-Carlo integration (Section 2.1.4), or any other way to solve integrals. It should be noted that when using the complex SH (Equation 2.24) its SH projection use the Fourier transform for the integral on ϕ and can thus benefit from the well known *Fast Fourier transform*

$$\mathbf{S}_l^m = K_l^m \int_\theta \underbrace{\Big(\int_\phi S(\theta,\phi)\, e^{im\phi}\,\mathrm{d}\phi \Big)}_{\text{Fourier transform}} P_l^m(\cos\theta)\sin(\theta)\,\mathrm{d}\theta\ . \tag{2.40}$$

The integral on θ is then resolved with a different method, a discrete Legendre transform based on recurrence relations [HRKM03], or a different quadrature [Sch13, MW11]. However, the projection is computed for $\mathcal{Y}$ and need to be converted to Y [CIGR99].

Using M uniformly distributed samples and N SH coefficients, the projection can be written with a matrix form

$$\begin{pmatrix}\mathbf{S}_0\\ \vdots\\ \mathbf{S}_N\end{pmatrix} \approx \underbrace{\frac{4\pi}{M}\begin{pmatrix}Y_0(\omega_0) & \dots & Y_0(\omega_M)\\ \vdots & & \vdots\\ Y_N(\omega_0) & \dots & Y_N(\omega_M)\end{pmatrix}}_{\mathbf{A}_{pdf}}\begin{pmatrix}S(\omega_0)\\ \vdots\\ S(\omega_M)\end{pmatrix} \tag{2.41}$$

where $4\pi/M$ is the *probability density function* for each sample. Using another distribution implies that the *pdf* of each sample is spread in the samples vector or can be embedded in $\mathbf{A}_{pdf}$ if the sampled directions are precomputed.

2.2.4 Least squares projection

The reconstruction from the SH basis (Equation 2.37) can be written in a matrix form when using M samples and N SH coefficients

$$\underbrace{\begin{pmatrix}S(\omega_0)\\ \vdots\\ S(\omega_M)\end{pmatrix}}_{b} \approx \underbrace{\begin{pmatrix}Y_0(\omega_0) & \dots & Y_N(\omega_0)\\ \vdots & & \vdots\\ Y_0(\omega_M) & \dots & Y_N(\omega_M)\end{pmatrix}}_{\mathbf{A}}\underbrace{\begin{pmatrix}\mathbf{S}_0\\ \vdots\\ \mathbf{S}_N\end{pmatrix}}_{x} \tag{2.42}$$

when $N = M$, the SH coefficients can be estimated by $x \approx \mathbf{A}^{-1}b$. For a general case where $N \neq M$, the Moore-Penrose generalized inverse is used $\mathbf{A}^+ = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$. However, in the experiments we have conducted, using a uniform sampling, the matrix $\mathbf{A}$ tends to be nearly singular when N tends to M, whether N is greater or less than M. A more in-depth analysis

of this problem as well as a comparison of the convergence of the SH projection between the method solving the scalar product (Equation 2.36) and this least square method is given in Section 5.3.1. The Equation 2.42 is equivalent to write the problem as a minimization problem

$$\|\mathbf{A}x - b\|^2 \ . \tag{2.43}$$

Sloan *et al.* [SHHS03, Slo08] prove that the global solution of this minimization problem is exactly the same as the one obtained by solving the integral for the SH projection (Equation 2.36). *i.e.* when N tends to infinity, the least squares method (Equation 2.42) and solving numerically the integral of projection (Equation 2.41) will yield the same SH coefficients. Hence, when the number of uniform samples N (and M) tends to infinity, we may understand $\mathbf{A}_{pdf}$ as the inverse of $\mathbf{A}$. As the SH are orthogonal functions, the columns of $\mathbf{A}$ tend to be orthogonal as well and the inverse of an orthogonal matrix $\mathbf{M}$ is $\mathbf{M}^T$. The least squares approach must invert a matrix, which is usually expensive [GvL13]. However, when $\mathbf{A}_{pdf}$ and $\mathbf{A}^{-1}$ are precomputed, both methods have the same complexity to estimate the SH coefficients, *i.e.* a matrix vector product. Nonetheless, constraints can be added to the least squares method, *e.g.* under the form of a regularization

$$\|\mathbf{A}x - b\|^2 + \lambda \|\Gamma x\|^2 \quad \Rightarrow \quad x = (\mathbf{A}^T\mathbf{A} + \lambda\Gamma^T\Gamma)^{-1}\mathbf{A}^T b \tag{2.44}$$

where Γ is the Thikonov matrix and λ control the amount of regularization. Constraints allow to make the estimation more robust and can take various forms, based on a target energy in order to make the estimation less sensitive to noise [LLW06], or to make it more robust with few samples [VSG+13]. Since the SH originates from the Laplace's equation, approaches rely on the Laplace-Beltrami operator to construct Γ in order to penalize strong oscillations [Slo08, NWNB15]. These constraints will act as a filter to eliminate artifacts due to the frequency representation of the SH.

2.2.5 Filtering

The frequencies are brutally truncated at the band limit when using a finite number of SH bands. This introduces a frequency discontinuity that will produce oscillations, commonly called *ringing*. The oscillations become stronger and stronger when the reconstruction band limit becomes small compared to the band limit of the projected signal [Slo08]. This is analogous to the *Gibbs Phenomenon*, *i.e.* projecting a discontinuous signal into a finite Fourier basis will exhibit ringing around the discontinuities. Hence, using a finite number of SH bands correspond to frequency filtering with a square signal, and in this case a common solution is to change the shape of the filter to attenuate the oscillations where each band is filtered independently. Many filters are proposed in the literature: sinc (1), Hamming (2), Hann (3) ...

$$(1)\ \frac{1+\cos(\pi\frac{l}{w})}{2} \qquad (2)\ \frac{\sin(\pi\frac{l}{w})}{\pi\frac{l}{w}} \qquad (3)\ 0.54 + 0.46\cos(\pi\frac{l}{w}) \tag{2.45}$$

These equations give the weight of the filter for each order l, where w is the size of the filter. In general, using n SH bands, the oscillations totally disappear when $w = n$ at the cost of a significant decrease in frequency. Typically, the optimal w is found by hand to avoid any *ringing* artifacts while keeping as much high-frequency as possible. Sloan [Slo17] proposes an automatic algorithm for 3 SH bands to deduce this optimal w for any function projected on SH.

Large oscillations tend to create *ringing* artifacts. Hence, constraints on the least squares approach penalizing these oscillations will help to reduce these artifacts. As shown by Sloan [Slo08], some constraints can be resolved analytically to find an equation for the filter that is applied after the estimation of the SH coefficients instead of a constraint applied during the estimation. The amount of constraint λ can also be adjusted according to the properties of the chosen constraint. Sloan [Slo08] also demonstrate that with the right amount of constraint, the traditional filters (Equation 2.45), and the constraint for least squares exhibit similar results.

Finally, Sloan [Slo08] proposes a filtering sensitive to the content. Indeed, considering the SH projection of the radiance, the most annoying *ringing* artifacts appear far from the principal lighting directions. Thus, stronger filtering can be applied in these areas, however this solution need an a priori knowledge of the lighting and is particularly inadequate to a general use of the SH.

2.2.6 Double and triple product

Computing the integral of the multiplication between two or three functions is very common in computer graphics. Using the SH definitions, the integral of the product of two functions (*double product*) projected on SH is straightforward to compute

$$\begin{aligned} D &= \int_\Omega L(\omega)F(\omega)\,\mathrm{d}\omega \\ &= \int_\Omega \Big(\sum_i \mathbf{L}_i Y_i(\omega)\Big)\Big(\sum_j \mathbf{F}_j Y_j(\omega)\Big)\,\mathrm{d}\omega \\ &= \sum_i \sum_j \mathbf{L}_i \mathbf{F}_j \int_\Omega Y_i(\omega)Y_j(\omega)\,\mathrm{d}\omega \\ &= \sum_i \mathbf{L}_i \mathbf{F}_i \ . \end{aligned} \tag{2.46}$$

The integral vanishes thanks to the orthogonality property of the SH (Equation 2.31). Hence the integral of the product of two functions is reduced to a scalar product of their SH coefficient vector. The integral of the product of three functions (*triple product*) is more complex. Indeed, the integral of the product of three SH basis functions is not directly simplified:

$$\begin{aligned} T &= \int_\Omega L(\omega)F(\omega)V(\omega)\,\mathrm{d}\omega \\ &= \int_\Omega \Big(\sum_i \mathbf{L}_i Y_i(\omega)\Big)\Big(\sum_j \mathbf{F}_j Y_j(\omega)\Big)\Big(\sum_k \mathbf{V}_k Y_k(\omega)\Big)\,\mathrm{d}\omega \\ &= \sum_i \sum_j \sum_k \mathbf{L}_i \mathbf{F}_j \mathbf{V}_k \int_\Omega Y_i(\omega)Y_j(\omega)Y_k(\omega)\,\mathrm{d}\omega \\ &= \sum_i \sum_j \sum_k \mathbf{L}_i \mathbf{F}_j \mathbf{V}_k C_{ijk} \end{aligned} \tag{2.47}$$

where C denotes the Clebsch-Gordan coefficients. These coefficients can be precomputed using a closed form solution of the integral $\int_\Omega Y_i(\omega)Y_j(\omega)Y_k(\omega)\,\mathrm{d}\omega$. This precomputation allows to generate an efficient code that apply the triple product. The triple sum implies a complexity of $\mathcal{O}(n^6)$, where n is the number of SH bands used. However, the number of zero in C reduce the complexity to $\mathcal{O}(n^5)$ [NRH04].

2.2.7 SH product and transfer matrix

Project on SH the product of two functions that are projected on SH is called the SH product. The SH product is analogous to the triple product (Equation 2.47) and involves

also the Clebsch-Gordan coefficients. Indeed, the SH projection of the multiplication of two functions is defined by

$$\begin{aligned}\mathbf{T}_k &= \int_\Omega L(\omega)F(\omega)Y_k(\omega)\,\mathrm{d}\omega \\ &= \int_\Omega \Big(\sum_i \mathbf{L}_i Y_i(\omega)\Big)\Big(\sum_j \mathbf{F}_j Y_j(\omega)\Big)Y_k(\omega)\,\mathrm{d}\omega \\ &= \sum_i\sum_j \mathbf{L}_i\mathbf{F}_j \int_\Omega Y_i(\omega)Y_j(\omega)Y_k(\omega)\,\mathrm{d}\omega \\ &= \sum_i\sum_j \mathbf{L}_i\mathbf{F}_j C_{ijk}\ .\end{aligned} \tag{2.48}$$

The double sum introduces a lot of calculation and if one of the two functions is known, *e.g.* $\mathbf{F}$, a lot of operations can be precomputed in a transfer matrix $\mathbf{M}$

$$\mathbf{T}_k = \sum_i \mathbf{L}_i \underbrace{\sum_j \mathbf{F}_j C_{ijk}}_{\mathbf{M}_{ik}}\ . \tag{2.49}$$

Furthermore, the function F can be directly integrated in the computation instead of its SH projection.

$$\mathbf{T}_k = \int_\Omega \Big(\sum_i \mathbf{L}_i Y_i(\omega)\Big)F(\omega)Y_j(\omega)\,\mathrm{d}\omega = \sum_i \mathbf{L}_i \underbrace{\int_\Omega F(\omega)Y_i(\omega)Y_k(\omega)\,\mathrm{d}\omega}_{\mathbf{M}_{ik}}\ . \tag{2.50}$$

Compared to Equation 2.49, F is not frequency approximated but, most of the time, each element of $\mathbf{M}$ for Equation 2.50 have to be resolved numerically. It's important to note that the transfer matrix is a linear operation, indeed applying a linear operator on the resulting vector $\mathbf{T}$ is the same as applying it on all the lines of the transfer matrix $\mathbf{M}$. For instance, applying the double product with each line of $\mathbf{M}$ is the same as applying it with $\mathbf{T}$. This is a very important property that allows to precompute a large amount of operations directly in the transfer matrix $\mathbf{M}$.

2.2.8 General SH product

Compared to the previous subsections, this subsection considers the integral of the multiplication of M functions

$$D_M = \int_\Omega F_1(\omega)F_2(\omega)\cdots F_M(\omega)\,\mathrm{d}\omega\ . \tag{2.51}$$

The brute force approach consists in the generalization of the double and triple product (Equation 2.46 and 2.47)

$$D_M = \sum_{i_1}\sum_{i_2}\cdots\sum_{i_n}(\mathbf{F}_{1,i_1}\mathbf{F}_{2,i_2}\cdots\mathbf{F}_{M,i_M}C_{i_1 i_2\cdots i_M}) \tag{2.52}$$

where C are the generalized Clebsch-Gordan coefficients defined by

$$C_{i_1 i_2\cdots i_M} = \int_\Omega Y_{i_1}(\omega)Y_{i_2}(\omega)\cdots Y_{i_M}(\omega)\,\mathrm{d}\omega\ . \tag{2.53}$$

Hence, using n SH bands, the complexity of the brute force approach (Equation 2.52) is $\mathcal{O}(n^{2M})$. A common acceptable alternative for real-time performance is the recursive application of the SH product (Equation 2.48) [ZHL+05]. *i.e.* F_1 and F_2 are multiplied together

and reprojected on SH, the result is multiplied with F_3 and so on until F_M. The complexity of this approach becomes $\mathcal{O}(Mn^5)$, however, this is an approximation of the brute force approach since each intermediate product is frequency approximated by the SH.

An alternative to these methods is to move into a space where the multiplication is simpler to compute. Thus, Ren *et al.* [RWS+06] propose to convert the SH function in log space to change the multiplication to a simple summation and then going back with an exponential operator. Recently, Xin *et al.* [XZA+21] have performed a 2D Fourier series of the SH coefficients. Thus, the product of functions can be written as a convolution of their Fourier series. Using the well known *Fast Fourier transform* to go in the so-called Fourier space of the coefficients, the convolution becomes a point-wise multiplication.

2.2.9 Zonal harmonics

Zonal Harmonics are a basis of a sub-space of the spherical harmonics defined for $m = 0$ (Figure 2.5). They are particularly useful to represent circularly symmetric signals

$$f(\omega) = f(\theta, 0) = \sum_{l}^{\infty} \mathbf{F}_l^0 Y_l^0(\omega) \; . \tag{2.54}$$

Hence, the number of ZH is linear to the number of SH bands used since when n SH bands are used, n ZH are involved. This is particularly efficient compared to the n^2 expansion on the set of SH, thus it has motivated multiple works on ZH in computer graphics in order to accelerate the computations. Especially by approximating any arbitrary function with different set of ZH [SLS05, NSF12, SBN15].

It turns out that all the SH of order l can be constructed from rotations and scaling of the ZH of same order, with at most $2l + 1$ different copies of the ZH (Figure 2.7)

$$Y_l^m(\omega) = \sum_{d=-l}^{l} \alpha_{l,-l}^m Y_l^0(\omega \to \omega_{l,d}) \tag{2.55}$$

where $\omega_{l,d}$ are the directions of rotations and $\alpha_{l,-l}^m$ are the weights to scale the ZH. This indicates that the SH projection can be deduced by computing the ZH projection for the different rotated copies of the ZH and then scaled by the weights. This equation can be

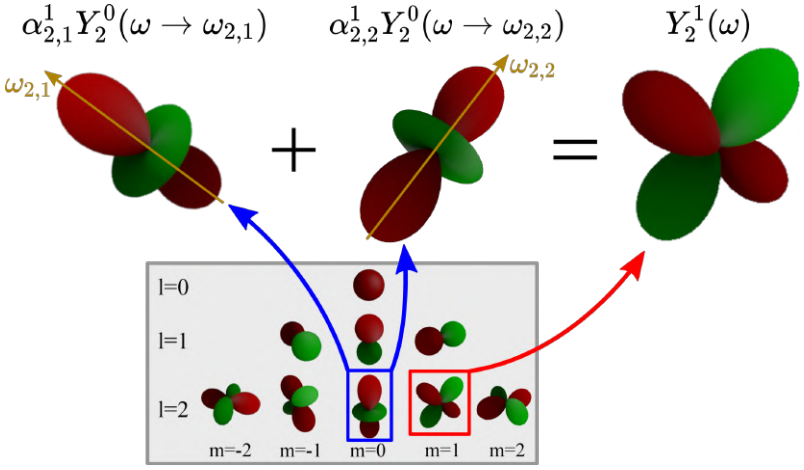


Figure 2.7: *The SH Y_l^m can be reconstructed from rotations $(\omega \to \omega_{l,m'})$ and scaling $\alpha_{l,m'}^m$ of the ZH of same order Y_l^0 [NSF12].*

rewritten in matrix form

$$\begin{pmatrix} Y_l^{-l}(\omega) \\ \vdots \\ Y_l^{l}(\omega) \end{pmatrix} = \underbrace{\begin{pmatrix} \alpha_{l,-l}^{-l} & \dots & \alpha_{l,l}^{-l} \\ \vdots & \ddots & \vdots \\ \alpha_{l,-l}^{l} & \dots & \alpha_{l,l}^{l} \end{pmatrix}}_{\mathbf{A_l}} \begin{pmatrix} Y_l^{0}(\omega \to \omega_{l,-l}) \\ \vdots \\ Y_l^{0}(\omega \to \omega_{l,l}) \end{pmatrix} \tag{2.56}$$

while an infinity of directions and weights can be used in this equation, Nowrouzezahrai *et al.* [NSF12] show that the sparsity of $\mathbf{A_l}$ can be maximized by choosing wisely the directions. Inversely, the *SH addition theorem* states that any ZH can be reconstructed by a weighted sum of spherical harmonics

$$Y_l^0(\omega \to \omega_{l,d}) = \sqrt{\frac{4\pi}{2l+1}} \sum_{m'=-l}^{l} Y_l^{m'}(\omega_{l,d}) Y_l^{m'}(\omega) \ . \tag{2.57}$$

As shown by Nowrouzezahrai *et al.* [NSF12] the *SH addition theorem* can be seen as the dual of Equation 2.55. Hence, injecting Equation 2.57 into Equation 2.55 give us

$$\begin{pmatrix} Y_l^{-l}(\omega) \\ \vdots \\ Y_l^{l}(\omega) \end{pmatrix} = \underbrace{\mathbf{A}_l \mathbf{D}_l}_{\hat{\mathbf{A}}_l} \underbrace{\begin{pmatrix} Y_l^{-l}(\omega_{l,-l}) & \dots & Y_l^{l}(\omega_{l,-l}) \\ \vdots & \ddots & \vdots \\ Y_l^{-l}(\omega_{l,l}) & \dots & Y_l^{l}(\omega_{l,l}) \end{pmatrix}}_{\mathbf{Y}_l} \begin{pmatrix} Y_l^{-l}(\omega) \\ \vdots \\ Y_l^{l}(\omega) \end{pmatrix} \tag{2.58}$$

where $\mathbf{D}_l = \sqrt{\frac{4\pi}{2l+1}}\mathbf{I}$, with $\mathbf{I}$ the identity matrix. In this case we observe that $\hat{\mathbf{A}}_l \mathbf{Y}_l = \mathbf{I}$, this means that replacing the SH basis functions by their coefficients, this equation will yield the same SH coefficients. Nowrouzezahrai *et al.* [NSF12] build on this property an efficient rotation algorithm (Section 2.2.12).

2.2.10 Convolution

The spherical convolution [DH94] is an important property of SH from which rotation of the SH coefficients is derived (Section 2.2.12). We note here some important notations for the following.

We note α, β, γ the Euler angles decomposition of a rotation R following the zyz decomposition [Sla99]:

$$R = R_z(\alpha) R_y(\beta) R_z(\gamma) \ . \tag{2.59}$$

Any axis ω can be written as the mapping of the z axis by the rotation R ($\omega = Rz$), then corresponding to one Euler angles decomposition. Hence, the Wigner D-matrix, traditionally written with the Euler angles can be rewritten by

$$D_{m'm}^l(\alpha, \beta, \gamma) = D_{m'm}^l(R). \tag{2.60}$$

Also, we note the spherical convolution between any function f and a kernel g rotated by R as

$$(f * g)(R) = \int_\Omega f(\omega) g(R\omega) \, \mathrm{d}\omega \ . \tag{2.61}$$

The spherical convolution using a SH decomposition for f and the kernel of convolution g can be written as

$$(f * g)(R) = \sum_{l,m} \sum_{m'=-l}^{l} \mathbf{F}_l^m \mathbf{G}_l^{m'} D_{m'm}^l(R) \ . \tag{2.62}$$

However, this relation is defined for the complex SH $\mathcal{Y}$ but can be adapted for their real counterparts Y. Unfortunately, the interesting geometrical properties of complex numbers are not as convenient using only real numbers. Indeed, rotation in the complex plane is just a scalar multiplication (Equation 2.62). Nevertheless, when g is a circularly symmetric kernel, this equation is non-null only for $m' = 0$ and in this case D is related to the real SH

$$D^l_{0m}(R) = \sqrt{\frac{4\pi}{2l+1}} Y^m_l(Rz) \; . \tag{2.63}$$

Thus, the convolution for the real SH with a circularly symmetric kernel can be simplified by

$$(f * \tilde{g})(R) = \sum_{l,m} \sqrt{\frac{4\pi}{2l+1}} \tilde{\mathbf{G}}^0_l \mathbf{F}^m_l Y^m_l(Rz) \; . \tag{2.64}$$

2.2.11 Linear transformation

The SH are invariant to linear transformation, in other words, the linear transformation can be applied before or after the SH projection, *i.e.* on the function or on its SH coefficients, the result will yield the same linearly transformed function. In addition, the linear transformation is applied as a matrix vector product, where the vector contains the SH coefficients and the matrix depend on the linear transformation. This can be proven by taking $\hat{g}$, a linearly transformed copy of g

$$g(\omega) = \hat{g}(T\omega) \; . \tag{2.65}$$

The SH projection of $\hat{g}$ and g have the following relations

$$\hat{\mathbf{G}}_i = \int_\Omega \hat{g}(T\omega) Y_i(T\omega) \, \mathrm{d}\omega = \int_\Omega g(\omega) Y_i(T\omega) \, \mathrm{d}\omega \; , \tag{2.66}$$

and replacing g by its reconstruction from the SH coefficients

$$\hat{\mathbf{G}}_i = \int_\Omega (\sum_j \mathbf{G}_j Y_j(\omega)) Y_i(T\omega) \, \mathrm{d}\omega = \sum_j \mathbf{G}_j \underbrace{\int_\Omega Y_i(T\omega) Y_j(\omega) \, \mathrm{d}\omega}_{\mathbf{M}_{ij}} \; . \tag{2.67}$$

Hence this equation shows that any linear transformation of the SH coefficients is a matrix vector product $\hat{\mathbf{G}} = \mathbf{GM}$. An example of linear transformation is the rotation (Section 2.2.12). Wang *et al.* [WXZ+06] seek to determine $\mathbf{M}$ such that the function g is shrunk or expanded around any axis with any angle value.

2.2.12 Rotation

Rotation being a linear transformation, it can be applied by multiplying the SH coefficients by a matrix. However, instead of resolving the integral of Equation 2.67, a link with the spherical convolution can be applied and it is strongly recommended to read the section on convolution (Section 2.2.10) before reading this section.

Let's consider that $\hat{g}$ is the rotated copy of g, $\hat{g}(\omega) = g(R\omega)$. The SH are bandwise stable by rotation, this means that projecting the rotated function will yield the same result as applying the rotation to the SH coefficients vector representing the non-rotated function, *i.e.* $\sum_i \mathbf{G}_i Y_i(R\omega) = \sum_j \hat{\mathbf{G}}_j Y_j(\omega)$. Also, it means that the coefficients of the band l will be modified only by the coefficients of the same band, in other words, the frequencies represented by the band l remain in the same band.

The convolution of a function f with a kernel g rotated by R is equal to the integral of their product (*i.e.* the double product (Section 2.2.6)). Thanks to the orthogonality of the SH, the convolution product is the dot product of their SH coefficients vector

$$\int_\Omega f(\omega)\hat{g}(\omega)\,\mathrm{d}\omega = (f * g)(R) = \sum_{l,m} \mathbf{F}_l^m \hat{\mathbf{G}}_l^m \ . \tag{2.68}$$

Replacing the definition of the convolution (Equation 2.62) into Equation 2.68 gives us

$$\sum_{l,m} \mathbf{F}_l^m \hat{\mathbf{G}}_l^m = \sum_{l,m} \sum_{m'=-l}^{l} \mathbf{F}_l^m D_{m'm}^l(R) \mathbf{G}_l^{m'} \ . \tag{2.69}$$

As the relation of Equation 2.68 is true for any f, we deduce that

$$\hat{\mathbf{G}}_l^m = \sum_{m'=-l}^{l} D_{m'm}^l(R) \mathbf{G}_l^{m'} \ , \tag{2.70}$$

which could be written in matrix form $\hat{\mathbf{G}} = \mathbf{M}\mathbf{G}$ where $\mathbf{M}$ is a *block diagonal sparse* rotation matrix for the complex SH coefficients, since each SH bands is rotated independently of the others

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & X & X & X & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & X & X & X & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & X & X & X & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & X & X & X & X & X & \dots \\ 0 & 0 & 0 & 0 & X & X & X & X & X & \dots \\ 0 & 0 & 0 & 0 & X & X & X & X & X & \dots \\ 0 & 0 & 0 & 0 & X & X & X & X & X & \dots \\ 0 & 0 & 0 & 0 & X & X & X & X & X & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} . \tag{2.71}$$

For the real SH, the bands are also independent under rotation but the real only implementation cannot be deduced directly as explained after Equation 2.62. However, as the rotation is a linear transformation (Section 2.2.11), each element can be defined by the integral of the multiplication between the SH and its rotated copy

$$\mathbf{M}_{ij} = \int_\Omega Y_i(R\omega) Y_j(\omega)\,\mathrm{d}\omega \ . \tag{2.72}$$

As demonstrated by Ivanic and Ruedenberg [IR96, IR98] each block of the matrix $\mathbf{M}$ can be computed recursively from the block of the previous order.

Rotation of Zonal Harmonics

As noted by Sloan *et al.* [SLS05], if the rotated function g is circularly symmetric, the rotation complexity is greatly reduced. The same procedure is applied as for the general rotation (Equation 2.68, 2.69 and 2.70). Except that the convolution (Equation 2.62) is replaced by the convolution with a circularly symmetric kernel (Equation 2.64). At the end, we obtain

$$\hat{\mathbf{G}}_l^m = \sqrt{\frac{4\pi}{2l+1}} \mathbf{G}_l^0 Y_l^m(Rz) \tag{2.73}$$

where, in this case, z is mapped to the symmetry axis of g by the rotation matrix R. Hence, the rotation becomes a point-to-point multiplication vector, instead of a vector matrix multiplication. As a result, any method that uses a ZH factorization can benefit of this efficient rotation (Section 2.2.9).

Other methods

The rotation of a SH coefficient $\hat{\mathbf{G}}_l^m$ around the z axis with an angle α is simply a linear combination of the same SH coefficients $\mathbf{G}_l^m$ and the one of opposite degree $\mathbf{G}_l^{-m}$:

$$\begin{cases} \hat{\mathbf{G}}_l^{-m} = \mathbf{G}_l^{-m}\cos(m\alpha) - \mathbf{G}_l^m\sin(m\alpha) \\ \hat{\mathbf{G}}_l^m = \mathbf{G}_l^{-m}\sin(m\alpha) + \mathbf{G}_l^m\cos(m\alpha) \end{cases} \tag{2.74}$$

and $\hat{\mathbf{G}}_l^0 = \mathbf{G}_l^0$. Furthermore, the sine and cosine factor can be computed recursively to avoid unnecessary computations of trigonometric functions:

$$\begin{cases} \sin(m\alpha) = 2\sin((m-1)\alpha)\cos(\alpha) - \sin((m-2)\alpha) \\ \cos(m\alpha) = 2\cos((m-1)\alpha)\cos(\alpha) - \cos((m-2)\alpha) \end{cases}. \tag{2.75}$$

Hence, using a traditional zyz decomposition for the rotation, the rotations around z are simple to compute on the fly but the rotation around y is complex to compute. The rotation around y can be decomposed by a rotation xzx, where the first rotation around x maps the z axis on the y axis and the second maps back the z axis to its initial location:

$$R_z(\alpha)R_y(\beta)R_z(\gamma) = R_z(\alpha)R_x(-90°)R_z(\beta)R_x(90°)R_z(\gamma) \tag{2.76}$$

where the rotations R_x are computed once for all. This method is known as the $zxzxz$ rotation in the literature [Gre03, KSS02]. While this method is an exact rotation for the SH coefficients, Krivánek *et al.* [KKP+06] propose a Taylor approximation for $R_y(\beta)$ instead of using its xzx decomposition which makes the method particularly competitive.

Finally, Nowrouzezahrai *et al.* [NSF12], use their zonal harmonics factorization (Equation 2.56 and 2.58) to propose an efficient rotation method. They show that the rotation of the SH coefficients can be done by simply rotate each element of $\mathbf{Y}_l$ (*i.e.* $Y_l^m(R\omega_{l,-l})$). However, this method requires a lot of SH evaluation and exhibits similar computation time as the $zxzxz$ rotation. Nevertheless, for a static function the product between $\mathbf{Y}_l$ and the vector of SH coefficients can be precomputed for each order l and for each rotation R, *i.e.* the per band SH projection is precomputed for each direction of factorization and for each rotation R (*i.e.* $\sum_{m=-l}^{l} Y_l^m(R\omega_{l,m})\mathbf{G}_l^m$). Since only the directions of factorization are rotated, each possible rotation are perfectly defined on a sphere and thus the precomputation for each R can be done on a cubemap. This improvement brings a significant gain for real-time computation.

2.3 Applications to computer graphics

The use of spherical harmonics in computer graphics has a rich history. Their potential covers a wide range of applications. *Precomputed Radiance Transfer* (PRT) is the starting point for many improvements in the efficient use of spherical harmonics. This section aims at tracing the history of spherical harmonics in computer graphics as well as an overview of the related work used in this thesis.

2.3.1 Environment map lighting

The first notable method that use SH for rendering was proposed by Cabral *et al.* [CMS87] to take advantage of the acceleration of the double product (Equation 2.46). The SH projection of the light, constituted by an environment map, is computed once for all the shaded points and the SH projection of the materials is computed once for each material in the

scene, the cosine of the rendering equation is packed with the material. Before applying the double product, a rotation is necessary at each shaded points since the light is in the world reference frame, while the material is in the local reference frame at the object surface. The light must be projected only once because the incident radiance is the same at any points when the visibility of the environment map is not taken into account. Indeed, for two points $\mathbf{x}$ and $\mathbf{y}$ of the scene the light coming from the map is exactly the same for the same direction $L_i(\mathbf{x}, \omega_i) = L_i(\mathbf{y}, \omega_i)$. Ramamoorthi and Hanrahan [RH01] show that the irradiance can be deduced very efficiently from the projection of the radiance because 3 SH bands are sufficient to capture the complexity of the irradiance, *i.e.* 9 SH coefficients. The irradiance on an opaque surface considers only the radiance received above the surface, and is thus rewritten by

$$E(\mathbf{x}) = \int_{\Omega} L(\mathbf{x}, \omega) \max(0, (\omega \cdot n_{\mathbf{x}})) \, \mathrm{d}\omega \qquad \text{(2.10 revisited)}$$

where the cosine becomes a clamped cosine. Hence, the irradiance is computed by a SH double product between the radiance and the clamped cosine. The clamped cosine is projected once on the local frame and is rotated efficiently since the clamped cosine is a circularly symmetric kernel, so a ZH function (Equation 2.73). As the clamped cosine is mainly captured on 3 SH bands, it is not necessary to store the environment map on more SH bands, *i.e.* only 9 SH coefficients are sufficient.

2.3.2 Material convolution

The evaluation of the color at each shading point of the scene corresponds to a convolution between the radiance L and reflectance F evaluated in direction ω_o (Equation 2.13)

$$L_o(\mathbf{x}, \omega_o) = \int_{\Omega} L_i(\mathbf{x}, \omega_i) F(\mathbf{x}, \omega_i, \omega_o) \, |\omega_i \cdot n_{\mathbf{x}}| \, \mathrm{d}\omega_i \ . \qquad (2.77)$$

In spherical harmonics rendering, there are two main methods to process the BRDF depending on how they are stored and computed.

Arbitrary BRDF An arbitrary reflectance F is a 4D function defined by two directions ω_i and ω_o. When one of the two directions is fixed, the material becomes a 2D function directly projectable on the SH. Let's consider that ω_o is fixed, Kautz *et al.* [KSS02] create a tabulation of the SH coefficients where ω_o is the table entry. The cosine in the integral can be directly included in the tabulation. As a result, Equation 2.77 is evaluated by a simple scalar product of the SH coefficients (Section 2.2.6). Arbitrary BRDF can also benefit from the zonal harmonics factorization, as detailed in Section 2.3.4. Appendix B provides the SH projection of the materials from the MERL database [MPBM03] as well as the database produced by Dupuy and Jakob [DJ18] using a similar technique than Kautz *et al.*. The appendix also contains a table referencing all the materials from these databases used in this thesis (Table B.1).

Decomposable BRDF The reflectance can also be the result of the separation of the diffuse and specular reflectance (Section 2.1.2). The diffuse reflectance is constant over Ω and can thus be taken out of the integral (Equation 2.77). In this case, it corresponds to the computation of the irradiance E multiplied by the diffuse color of the material. The irradiance can therefore be computed by the method of Ramamoorthi and Hanrahan [RH01] as explained above. The specular contribution can be computed in the same way as an arbitrary reflectance. However, if the specular lobe is circularly symmetric, the computation can be

greatly accelerated using Zonal Harmonics (Section 2.2.9). Indeed, in this case, the lobe can only be represented by ZH whose convolution can be efficiently computed (Section 2.2.10).

However, this introduces redundancies between the computations of the diffuse and specular contribution. These redundancies are identified and factored in Section 4.1. Finally, if the radiance is also a circularly symmetric function, as is the case when using spherical light, the convolution can be further accelerated, but this is explained in more detail in Section 5.2.7.

2.3.3 Precomputed radiance transfer

Initial formulation The methods described in Section 2.3.1 do not take into account the global lighting. Sloan *et al.* [SKS02] precompute transfer functions at each vertex of the mesh to compute the interreflections that appear in the scene (Figure 2.8). First, let's consider only direct lighting, where L_i is the radiance emitted by an environment map. Thus, L_i is projected on SH once for the whole scene, but does not take into account the shadowing at each point of the scene. A visibility term V is introduced in the rendering equation to obtain a correct shadowing

$$L_o(\mathbf{x},\omega_o) = \int_\Omega L_i(\mathbf{x},\omega_i) F(\mathbf{x},\omega_i,\omega_o) \underbrace{|\omega_i \cdot n_\mathbf{x}| V(\mathbf{x},\omega_i)}_{T(\mathbf{x},\omega_i)} \mathrm{d}\omega_i \ . \tag{2.78}$$

V is a binary function equal to 1 when there is no object in direction ω_i from $\mathbf{x}$ and 0 otherwise. For direct lighting, the transfer function T is the visibility multiplied with $|\omega_i \cdot n_\mathbf{x}|$ and is projected on SH at each vertex of the mesh. At runtime, the outgoing radiance is computed at each vertex and then interpolated for each triangle. The organization of the final lighting computations depends on the nature of the material. For the diffuse part, *i.e.* constant over the sphere, F can be taken out of the integral. Thus, the outgoing radiance is equal to the material multiplied by the dot product of the SH coefficients of the incoming radiance and the transfer function $L_o = F \times (\mathbf{L} \cdot \mathbf{T})$. For the specular contribution, a SH product is computed between $\mathbf{L}_i$ and $\mathbf{T}$ and the result is convolved with the material. Precomputing the transfer matrix (Section 2.2.7) instead of the SH projection of T at each vertex requires less operation at runtime to compute the SH product but consumes more storage.

Global illumination is computed by taking into account the interreflections of the scene in the transfer function. The diffuse interreflections are view independent, then the trans-

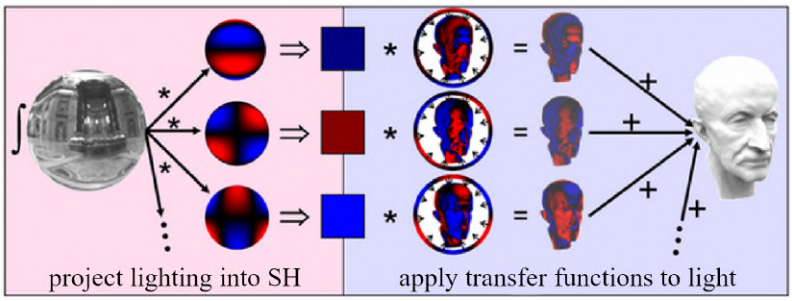


Figure 2.8: Precomputed Radiance Transfer *(PRT) precomputes a transfer function, representing the interreflections, at each vertex of a mesh and apply it to light at run time to compute the shading [SKS02]. The application is only a dot product for diffuse contribution but is more complex for specular contribution.*

fer function at $\mathbf{x}$ is a scaling of the transfer function of the other points $\mathbf{y}$ of the scene visible from $\mathbf{x}$ according to the color of the diffuse material at $\mathbf{y}$ and the cosine of the rendering equation. The double product between the transfer function and the incident radiance is applied to compute the diffuse interreflections $\mathbf{L} \cdot \mathbf{T}$. For the specular contribution, being view dependent, the transfer function at $\mathbf{y}$ is convolved with the material at the same location and evaluated in the direction of reflection to yield the radiance transferred from $\mathbf{y}$ to $\mathbf{x}$. Unlike for the direct lighting, it is not a SH product between the incident radiance and the transfer function that is computed at $\mathbf{x}$ but a linear transformation to convert the incident radiance coming from the environment map into an incident radiance coming from the scene. Which is then convolved with the material to compute the specular interreflections.

Hence, at runtime the shading at $\mathbf{x}$ considers only the incident radiance arriving at the same point. This creates a strong assumption that the radiance received at $\mathbf{x}$ must be the same as at any point $\mathbf{y}$. This is true with an environment map but becomes false with close-range lighting. Nevertheless, this remains a consistent approximation with diffuse materials and close interreflections. Furthermore, the light sources cannot be located inside the scene at the risk of falsifying the precomputed (binary) visibility. Let's consider that $\mathbf{y}$ is visible from $\mathbf{x}$ ($V(\mathbf{x}, \vec{\mathbf{xy}}) = 0$), if the light source is between $\mathbf{x}$ and $\mathbf{y}$, thus visible from $\mathbf{x}$, the precomputed visibility indicates that the light is not visible. In such a case, the SH projection of the visibility must be recomputed.

In order to use local light sources, whose radiance field is not constant on the whole scene, the incident radiance of such sources is projected on SH on a sparse set of samples and interpolated between the samples at run time. To use dynamics light sources, the SH projection needs to be computed at each frame.

This initial PRT formulation allows to use a dynamic camera but the geometry of the scene is fixed since the transfer function $\mathbf{T}$ is precomputed at each vertex. The output direction ω_o is fixed for each vertex if the camera is static. In this case, the specular contribution of the material, being view dependent, becomes a spherical function that can be packed directly in the transfer function. This greatly simplifies the runtime computation since there are only two functions left, the radiance L_i and the transfer T to convolve by the SH double product.

Dynamic scenes In order to use dynamic geometry, the SH projection of the visibility must be computed at each frame. To avoid computing the SH projection from scratch

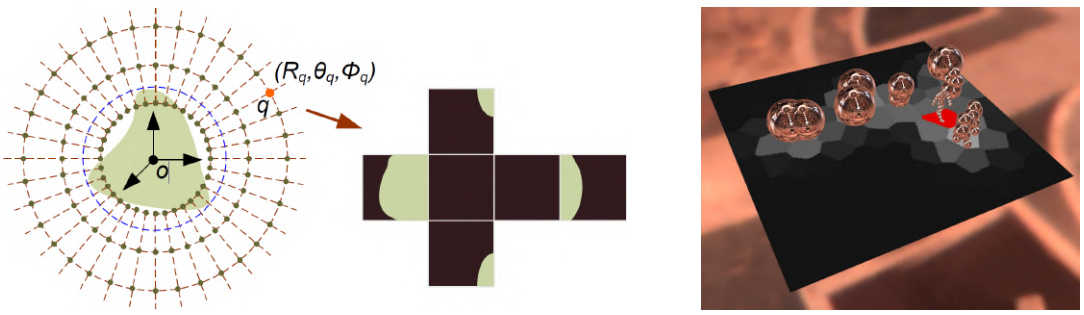


Figure 2.9: *To enable dynamic visibility in PRT system, Zhou* et al. *[ZHL+05] encode the visibility from different viewpoint q. Ren* et al. *(left) [RWS+06] (right) use an approximation of the scene with spheres where the visibility is the product of the visibility of each sphere.*

at each frame Zhou *et al.* [ZHL+05] precompute the *Object Occlusion Field* (OOF) of each dynamic, but rigid object. The OOF gives the SH coefficients representing the visibility for any point q that sees the center of the object O in a direction ω (Figure 2.9 - left). The OOF is a set of SH coefficient vectors precomputed for different distances between q and O and different directions ω. Hence, the OOF structure yields the SH visibility for a single object. Considering N objects, the total visibility is the multiplication of the visibility of each object

$$V(\mathbf{x},\omega) = V_1(\mathbf{x},\omega)V_2(\mathbf{x},\omega)\cdots V_N(\mathbf{x},\omega) \tag{2.79}$$

where $V_i(\mathbf{x},\omega)$ is the visibility of the i-th object according to the position $\mathbf{x}$ and direction ω. Precisely, the OOF yields the SH projection of the visibility $\mathbf{V}$ of each object. Thus, the total visibility on SH is obtained by applying the general SH product (Section 2.2.8) or by applying the SH triple product recursively (Section 2.2.6) with the SH visibility of each object.

For each point $\mathbf{x}$, the shading is managed as follows. The occluders and local light sources are sorted in a list according to their distance to $\mathbf{x}$ and the current visibility is the visibility of the object on which $\mathbf{x}$ is located. Then the list is processed, when it is an occluder, the current visibility is multiplied by the visibility of the occluder. When it is a light source, the convolution is computed with the current visibility. This method is limited to few rigid objects to remain efficient. Ren *et al.* [RWS+06] discretize the scene by a set of spheres (Figure 2.9 - right). Thus, the OOF is the same for each sphere of same radius and can be precomputed once for a set of radius. Instead of using the classical SH product, they convert the SH projection into an exponential space where the multiplication becomes an addition. This was described earlier (Section 2.2.8) as well as the proposal of Xin *et al.* [XZA+21] who switch to a Fourier space to perform the multiplication. The SH exponentiation proposed by Ren *et al.* has been used to develop a practical shading of height fields [GN15].

These improvements allow the use of dynamic visibility but do not take into account the interreflections of dynamic objects. Iwasaki *et al.* [IDYN07] propose a solution using a similar principle to that of OOF but adapted for interreflections.

However, the OOF method requires to compute the SH coefficients from different points q (Figure 2.9 - left). These points q are distributed on 3 dimensions, the direction (θ,ϕ) and the distance of the object (r). Wang *et al.* [WXZ+06] propose to use only one r and approximate the SH coefficients for the other r by scaling the visibility.

In general, rendering requires efficient rotations, *e.g.* to convert a function from world coordinates to local coordinates of the surface. In some cases of SH rendering, a rotation is sufficient to render a dynamic scene, for example by computing the rotation of the transfer function initially computed in the local space to send it to the rotated local space according to the movements of the object. In order to take advantage of the efficient rotations allowed by ZH, Sloan *et al.* [SLS05] approximate the transfer function by a set of ZH lobes (Section 2.2.12). However, this is not the only method relying on ZH factorization and these methods are more discussed in the dedicated section (Section 2.3.4).

Diverse improvements The initial PRT formulation benefit of the ZH (Section 2.2.10) to compute an efficient convolution between the radiance and the reflectance. This limits the reflectance to circularly symmetric specular lobes, such as the blinn-phong material. Ramamoorthi and Hanrahan [RH02] avoid this limitation by preconvolving the radiance

and the reflectance with each possible rotation of the radiance. This method is not limited to any material but very costly in terms of memory in particular for anisotropic materials. The BRDF is a function that depends on two directions (ω_i, ω_o), so when one direction is fixed, the resulting function is spherical and can therefore be projected onto SH. Kautz *et al.* [KSS02] use a spherical map where each element corresponds to a direction ω_o and stores the corresponding reflectance projected onto the SH. The memory cost is much more negligible than the Ramamoorthi and Hanrahan method but the convolution have to be evaluated at each shaded points. Thus, for both methods, at runtime, they fetch the necessary SH coefficients from the memory. This can represent a lot of memory accesses, and thus be a bottleneck, especially when many SH bands are involved.

When using large scenes, the PRT methods are particularly memory-intensive because of the storage of the SH coefficients and the transfer matrix of the transfer function at each vertex. In order to reduce the memory cost, Sloan *et al.* [SHHS03] build cluster on vertices with similar transfer function with a principal component analysis. This reduces the storage, which keeps only one transfer function per cluster, but also the execution time since the number of convolutions is also reduced to the number of clusters and not to the number of vertices.

The initial PRT formulation computes the shading at each vertex to interpolate it at each pixel. However, the use of normal maps change the local reference frame at each point of the triangle forcing the explicit computation of the lighting at each pixel. Sloan [Slo06] avoids this problem by storing the spherical convolution between radiance and reflectance at each vertex which is interpolated at each pixel and evaluated in the desired direction according to the new normal.

Dubouchet *et al.* [DSJN19] introduce a new framework to take into account the lighting effect that volumetric media will have to the surface of the scene. For additional information on PRT methods, the reader is referred to state-of-the-art documents [Leh07, Ram09].

2.3.4 Advanced rendering

SH allow an efficient computation of the rendering equation and can therefore be applied to a whole range of techniques. However, if some methods are satisfied with a naive use of spherical harmonics, some methods take advantage of advanced properties of the SH.

Zonal Harmonics factorization

The main interest of using ZH instead of SH is to take advantage of the efficient convolution (Section 2.2.10) and rotation (Section 2.2.12) properties. Thus ,any circularly symmetrical signal can only be represented using ZH and can therefore take advantage of these properties. For instance, the use of a circularly symmetric specular lobe simplifies the computation of the convolution with any radiance [SKS02]. For an arbitrary signal, there are several factorization methods on the ZH in order to represent the signal only with ZH lobes. Sloan *et al.* [SLS05] use an optimization to approximate the signal with an arbitrary number of lobes that will best represent the function. Nowrouzezahrai *et al.* [NSF12] show that any SH basis function can be represented with the ZH of same order with a fixed number of lobe directions (Section 2.2.9). Thus, any signal can be factored onto the same ZH lobes. Wang and Ramamoorthi [WR18] use this principle to propose an efficient method to project the radiance emitted by polygonal lights onto the SH. Belcour *et al.* [BXH+18] also use this principle to propose a method of integration on a clipped spherical polygon. The work of

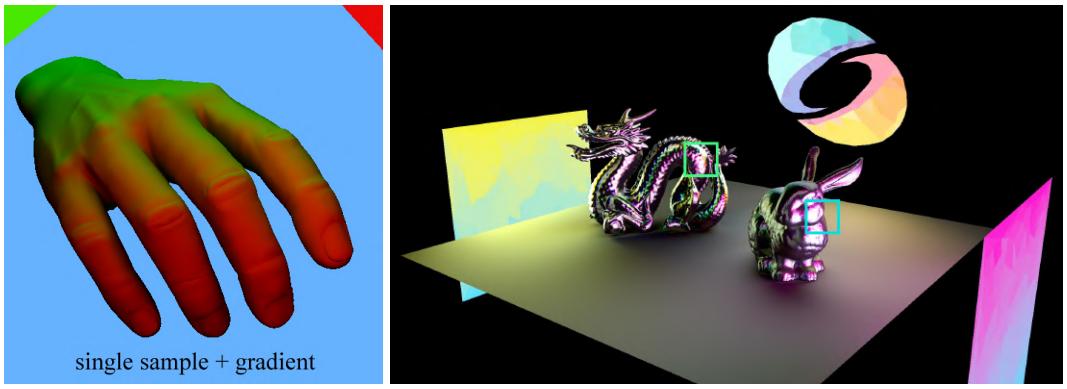


Figure 2.10: *The work of Annen* et al. *[AKDS04] (left) and Wu* et al. *[WCZR20] (right) use the gradient of the SH coefficients to interpolate them accurately using a Taylor or Hermite interpolation.*

Soler *et al.* [SBN15] can be seen as a generalization of the principle of Nowrouzezahrai *et al.*, indeed they generalize the factorization with any circularly symmetric kernel.

Spherical harmonics gradient

In computer graphics, many methods interpolate the radiance, so computing the radiance gradient helps to improve the interpolation quality. Thus, methods are interested in computing the gradient of the SH coefficients, *e.g.* to interpolate the radiance projected on SH (Figure 2.10). Annen *et al.* [AKDS04] propose to compute the SH coefficients of the incident illumination with their spatial gradient on a sparse set of samples and then reconstruct the SH coefficients at any shaded point. However, gradients are computed with a numerical integration over the area light, which significantly increases computation times when considering higher SH frequency bands. On the other hand, based on the initial results to compute the SH coefficients representing the radiance emitted by polygonal light [WR18], Wu *et al.* [WCZR20] derive an analytical solution to compute the spatial gradient of the SH coefficients representing also the radiance emitted by polygonal lights. This thesis follows the same principle, but adapted to spherical lights (Section 3.3).

2.3.5 Area lights

Simple light primitives, such as point, directional or spot lights are convenient to use. Indeed, the lighting received at a point of such light sources corresponds only to a single direction, where the lighting received from an area light correspond to a set of direction (Figure 2.11). Hence, their use for real-time rendering is more complex and expensive but has a greater physical sense. The integration of the lighting function over an area light source was studied for different representations of the light source. While Monte-Carlo integration provides a ground-truth solution for lighting, analytical integration schemes introduce less stochastic error while being specific to the area light representation. Dealing with polygonal representation of the area light, Heitz *et al.* [HDHN16] propose a method called *Linearly Transformed Cosines* (LTC) to approximate the lighting by transforming the projection of the polygon onto the unit sphere toward a clamped cosine according to the BRDF distribution. Multiple improvements of LTC have been proposed, for instance Peters couples LTC with importance sampling of the BRDF [Pet21], and Allmenröder and Peters [AP21] combine LTC with SH. The initial formulation of LTC limits them to constant polygonal lights, whose emission color is constant on the surface of the light source,

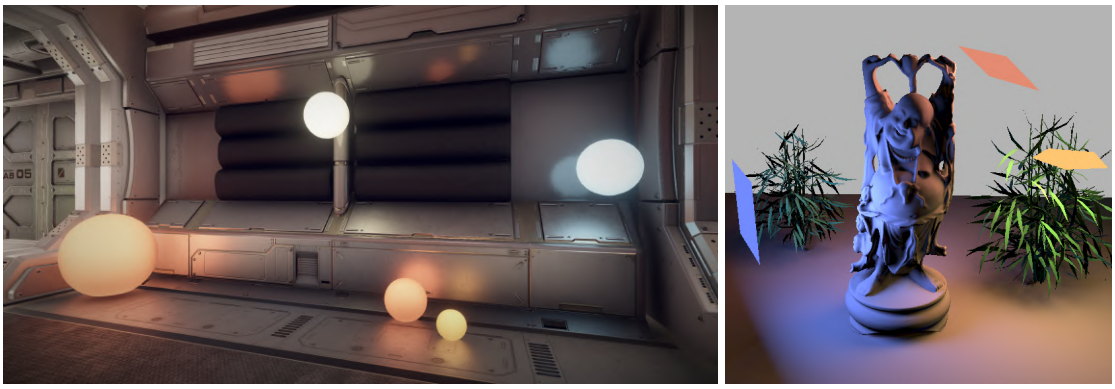


Figure 2.11: *The model of Dupuy* et al. *[DHB17] (left) for the spherical light contribution allows analytical integration for real time as well as efficient sampling for offline. Wang and Ramamoorthi [WR18] (right) project the radiance field produced by polygonal lights onto SH.*

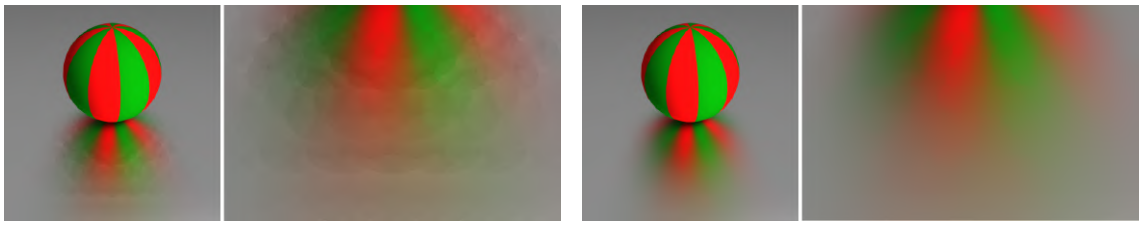
as well as limited to isotropic materials. Kuge *et al.* [KYM22] propose to lift the first limitation by extending LTC to any free-form area light whose emission color can vary on the surface. KT *et al.* [KHDN22] lift the second limitation by allowing the use of anisotropic GGX materials with LTC. Dealing with spherical representation of the area light, Dupuy *et al.* [DHB17] propose a spherical cap parameterization enabling the use of an analytic integration and an importance sampling scheme of the lighting function.

These methods are however limited to a small number of lights to provide real-time performances and the precomputation of the radiance transfer is an efficient solution to compute lighting with numerous lights. For SH based lighting, some approaches introduce an analytical solution to compute the SH coefficients of polygonal lights [WR18, BXH+18]. Following the approach introduced in Wang and Ramamoorthi [WR18], Wu *et al.* [WCZR20] propose to precompute and store the SH coefficients and their spatial gradients on the vertices of a low resolution 3D grid. These coefficients are then interpolated during rendering, which allows to handle several hundreds of polygonal lights in real-time.

The approaches developed to project the radiance field of polygonal sources [WR18, WCZR20] are based on the ZH factorization principle proposed by Nowrouzezahrai *et al.* [NSF12] (Section 2.2.9). Zhou and Wei [ZW20] use the same principle but adapted for spherical lights, although their approach seems to be usable for real-time, since their algorithm has a similar complexity to the one of Wang and Ramamoorthi [WR18], they demonstrate impressive offline results. In contrast, the methods developed in this thesis for spherical lights are not based on the method of Nowrouzezahrai *et al.* but rely on the fact that the radiance field emitted by a spherical light and received at a point is a circularly symmetric signal, thus only representable with ZH.

2.3.6 Radiance caching and probes lighting

Irradiance and radiance caching are widely used to compute global illumination by reconstructing the lighting function from a sparse set of caches storing the global light field [WRC88, KGBP05, KGW+08] (Figure 2.12). When the caches are distributed on the surface of opaque objects, only the radiance received or emitted above the surface is considered. Thus, the initial methods of radiance caching use the hemispherical harmonics [KGBP05, KGW+08] (Section 2.2.2). To improve the sparsity of the caches and enhance the interpolation accuracy, a solution is to use the function gradients [WH92, KGBP05, JDZJ08, JZJ08].



(a) Radiance caching from [KGBP05] *(b) Radiance caching from [KGW+08]*

Figure 2.12: *Radiance caching methods capture the radiance on caches distributed over the scene surface and interpolate the radiance from the nearest caches. Both results are rendered with the same caches, only the computation of the radiance gradient changes [KGW+08].*

These methods estimate the gradient numerically using Monte-Carlo; in contrast, we compute the gradient of the SH-encoded light field analytically. Several recent approaches rely on offline precomputation of radiance cache for real-time rendering [DSJN19, ZBN19] or even with neural networks [MRNK21].

Based on a volumetric approach introduced by Greger *et al.* [GSHG98], caches are under the form of probes distributed volumetrically in the scene (Figure 2.13). To simplify the interpolation of the probes, they are usually placed on a uniform 3D grid. However, the placement of the probes is critical because the management of the visibility of the probes is particularly problematic, indeed a probe not visible by the point of a surface should not generally contribute to the radiance at that point. Cupisz [Cup12] uses a tetrahedral grid which allows to simplify the placement of the probes. However, since the probes are in space and not on the surface of the objects, an interpolation is generally not sufficient for highly specular materials, but a reprojection must be computed so that the radiance directions are corrected [RLP+20] or a better placement of probes can also avoid this problem [WKKN19]. Many approaches store the data from each probe in the form of a map [MMNL17, SL17, MGNM19]. Depending on the materials in the scene, it is unnecessary to use high resolution maps, so filtering methods can be implemented [MS16]. Furthermore, the probes can also be projected onto the spherical harmonics [IKA+21]. For additional information, the reader is referred to the recent state of the art on probes [MMSM21].

Yet, even using techniques that exploit a sparse set of probes [SL17], the cost of storage becomes high for large and complex scenes. To overcome this limitation, probes data can be compressed with only a low loss in quality [SS21].

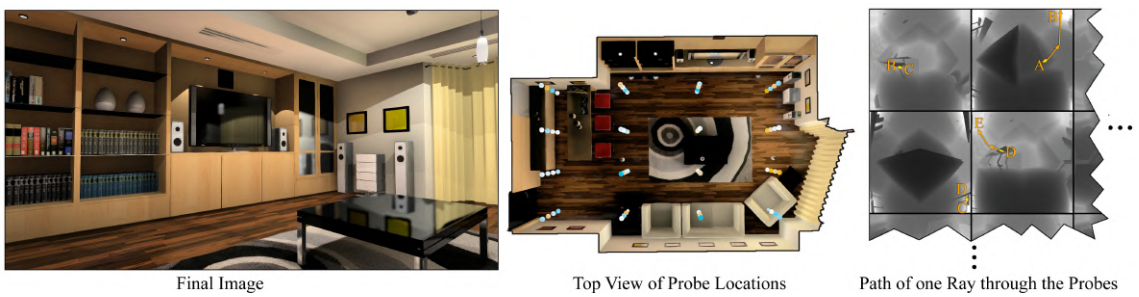


Figure 2.13: *McGuire* et al. *[MMNL17] use a set of probes volumetrically distributed in the scene capturing the radiance arriving at the probe position in order to interpolate it to the point of the nearby scene. This method uses a map representation for the radiance but these maps can be projected on the SH.*

Figure 2.14: *The occlusion characterizes the accessibility of a point according to the surrounding geometry (left: without occlusion, middle: only occlusion and right: combination of both). Result given by Jimenez* et al. *[JWPJ16].*

2.3.7 Occlusion

Occlusion is derived from the visibility at small distance scales (Figure 2.14). The distinction is necessary for real-time methods, as techniques handling visibility at the scene scale are not suitable for rendering convincing occlusion at the object scale. Thus, it is necessary to develop other methods to compute the occlusion. The occlusion defines the accessibility of a point with respect to the geometry that surrounds it, the occlusion is therefore a scalar and not a spherical function. Occlusion is sufficient for diffuse materials, since a diffuse material does not need to know where the shadow comes from. However, for specular materials directional occlusion is necessary. Directional occlusion can be defined as the presence of geometry in one direction over a small distance, so it is a spherical function that can be approximated by SH. Directional occlusion is perfectly suited for SH since it is generally spatially low frequency. Herholz *et al.* [HSSS12] adapt the traditional *Screen Space Ambient Occlusion* (SSAO) [BS08] to project the occlusion on SH at each pixel. Jimenez *et al.* [JWPJ16] take advantage of the convolution with ZH by approximating the directional occlusion by a set of ZH lobes.

2.3.8 Differentiable rendering and neural radiance field

Differentiable rendering and *Neural Radiance Field* (NeRF) are two hot topics in computer graphics. While these two topics are not directly related to SH, they can still benefit from the various improvements on the use of SH.

Differentiable rendering Computing the gradient of numerous rendering quantities is at the heart of differentiable rendering approaches (Figure 2.15 - left). Aiming at solving inverse problems [LTL+19, LZBD21, NDDJK21], differentiable rendering frameworks mainly rely on path-space [ZMY+20] or physical formulation of light transport [ZWZ+19].

While works proposed in this thesis are not directly related to such differentiable approaches, they may serve as building blocks for improving gradient computations of the incident light field. Indeed, several approaches based on differentiable rendering take advantage of SH. For instance, Liu *et al.* [LTL+19] directly compute the gradient of the SH basis function. A method to compute it efficiently is developed in this thesis (Section 3.2). Lyu *et al.* [LHL+21] use a spherical representation of the scene to compute differentiable shadow by projecting the spheres on SH. This approach can thus directly benefit from our

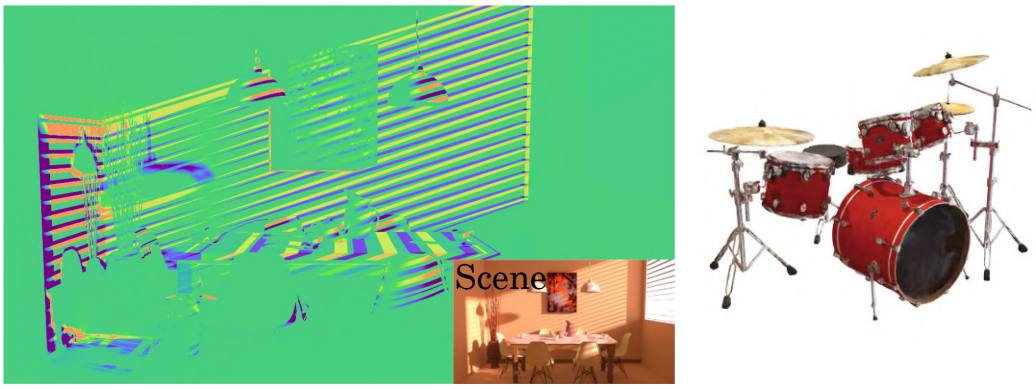


Figure 2.15: *Differentiable rendering [ZMY+20] (left) and neural radiance field [MST+20] (right) are two hot topics in computer graphics. The scene is differentiated according to the sun position and the result is displayed as false colors. The goal of neural radiance field approaches is to reconstruct the radiance field with a set of capture from different viewpoints.*

method on spherical light (Chapter 3). Technically, while most of the methods rely on finite differences or automatic differentiation frameworks, we introduce an analytical expression optimizing the evaluation of the SH coefficient gradients.

Neural radiance field The goal of NeRF approaches (Figure 2.15 - right) [MST+20] is to render a scene from a new point of view from a finite set of viewpoints. In other words, these methods seek to reconstruct efficiently the radiance field at any point in the scene.

Similarly, the goal of some of the methods developed in this thesis is to reconstruct the radiance field, *e.g.* the radiance field produced by spherical lights (Chapter 3). Some recent NeRF methods use the SH and their gradients for this purpose [YLT+21, STC+22]. Hence, the method we develop to compute efficiently the gradient of the SH basis functions constitute a direct improvement for these methods.

In a way, NeRF methods seek to deconvolve the lighting in order to recover the radiance and reflectance at each point of the scene. Mahajan *et al.* [MRC07] achieve this deconvolution with a SH representation.

2.4 Other basis functions to encode light transports

Spherical harmonics have been widely used in computer graphics to encode the local light transports. If their use allows a particularly efficient low frequency approach, they do not allow a low cost all frequency approach. Wavelets and spherical Gaussians allow to lift this limitation but with their own drawbacks (Section 2.4.1). However, these approaches are designed to encode only local light transports. Recent approaches propose a paradigm shift from these methods by encoding global light transports instead (Section 2.4.2).

2.4.1 Local light transports

Wavelet Wavelets Ψ are basis functions, historically competing with spherical harmonics on PRT methods. Indeed, wavelets allow an all-frequency approximation more easily than SH which are particularly adapted to a low frequency approximation (Figure 2.16) [NRH03]. Since both wavelets and spherical harmonics are orthogonal basis functions, the double product is of equal complexity. The triple product has a complexity of $\mathcal{O}(N \log N)$ when using the N first basis function, while the triple product complexity with SH is $\mathcal{O}(N^{5/2})$

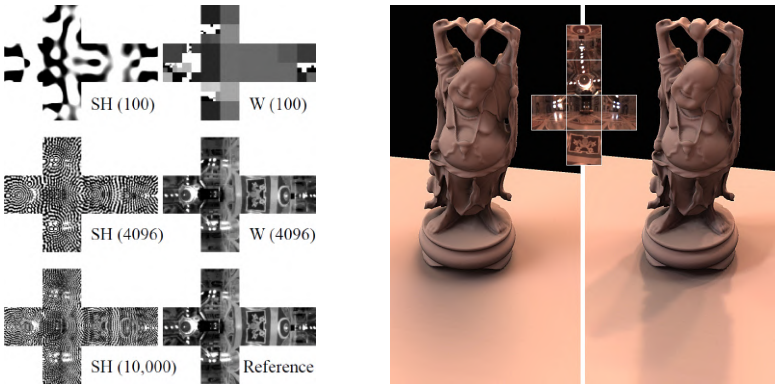


Figure 2.16: *Wavelets encode high-frequency information with fewer coefficients than SH, allowing all-frequency illumination more easily, where SH are more likely to produce a low-frequency illumination. However, wavelets are more costly for the precomputation and at runtime. The environment map is encoded on SH on the left and wavelets on the right, also the final rendering uses SH on the left and wavelets on the right [NRH03].*

(Section 2.2.6). This is due to the property of wavelets which often nullifies the integral involved in the triple product [NRH04]

$$T = \sum_i \sum_j \sum_k \mathbf{L}_i \mathbf{F}_j \mathbf{V}_k \int_\Omega \Psi_i(\omega)\Psi_j(\omega)\Psi_k(\omega)\,\mathrm{d}\omega \ . \tag{2.80}$$

However, even with few basis functions, SH can still bring consistent results but limited in frequency, where wavelets must use a minimum number of basis functions before having a consisting result since the basis function are non linear. The use of near and dynamic light is complex in PRT systems, especially when the radiance emission function is a complex function, such as a textured area light. The initial PRT formulation [SKS02] samples the radiance projected on SH on a sparse set of samples. Instead, Sun and Ramammorthi [SR09] approximate the radiance field projected on wavelets at any point in the scene by an affine transformation of the radiance field emitted by the surface light.

Importance sampling is an important issue in many fields that requires an a priori knowledge of the function. Clarberg *et al.* [CJAMJ05] propose to distribute samples according to an approximation of the function by wavelets, this work has been adapted to SH [JCJ09]. These methods are not directly related to PRT methods since they are intended to place samples for Monte-Carlo techniques.

Spherical Gaussian Spherical Gaussians are not basis functions like SH (Figure 2.17), but a mixture of spherical Gaussians can fit any spherical signal. Like ZH, spherical Gaussians can only represent circularly symmetric functions. However, even limited to circularly symmetric functions, ZH can represent complex functions where spherical Gaussians have to use a mixture of Gaussians to approximate them [WRG+09]. Xu *et al.* [XSD+13] introduce anisotropic spherical Gaussian which are particularly useful for representing more complex specular lobes than simply circularly symmetric ones. Like wavelets, that trade steerability for high frequency and locality, spherical Gaussians allow an all-frequency approach more suitable than SH. However, convolution and product operations of spherical Gaussians (anisotropic or not) generally do not have simple analytical solutions, even if fast and accurate approximations are developed [WRG+09, XSD+13]. Spherical Gaussians are used in various global lighting simulation applications. For instance, Xu *et al.* [XCM+14] compute global illumination by fitting spherical Gaussians representing the radiance emitted

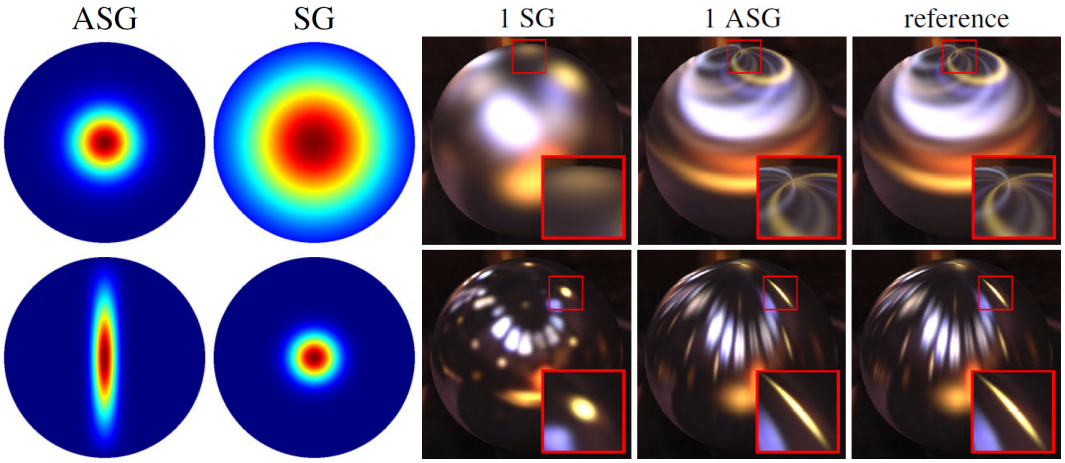


Figure 2.17: *The Spherical Gaussian (SG) or Anistropic Spherical Gaussian (ASG) [XSD+13] are not basis functions like spherical harmonics. But the illumination can be computed by representing the radiance and reflectance on a mixture of Gaussian lobes. Here, only one spherical Gaussian is used, the left part represent different Gaussian and anisotropic Gaussian lobe and the red part shows renderings results.*

by each triangle in the scene. Currius *et al.* [CDAS20] propose an application of precomputed radiance transfer where mixtures of Gaussians are used to approximate the local light field at each point on the scene surface.

2.4.2 Global light transports

The PRT methods encode the transfer functions and lighting at each illuminated point of the scene, so this represents various difficulties mentioned so far and especially storage. In order to overcome this limitation, some approaches try to encode the global light transports of the underlying scene (Figure 2.18). Rainer *et al.* [RBRD22] rely on neural networks to encode the incident lighting on one side and the light transfer on the other side. An other network takes both encoding to compute the final lighting to each pixel. Belcour *et al.* [BDBS22] extract basis functions directly from the global light transfers that arise from various luminous configuration. Thus, both methods reduce the light transports to a small

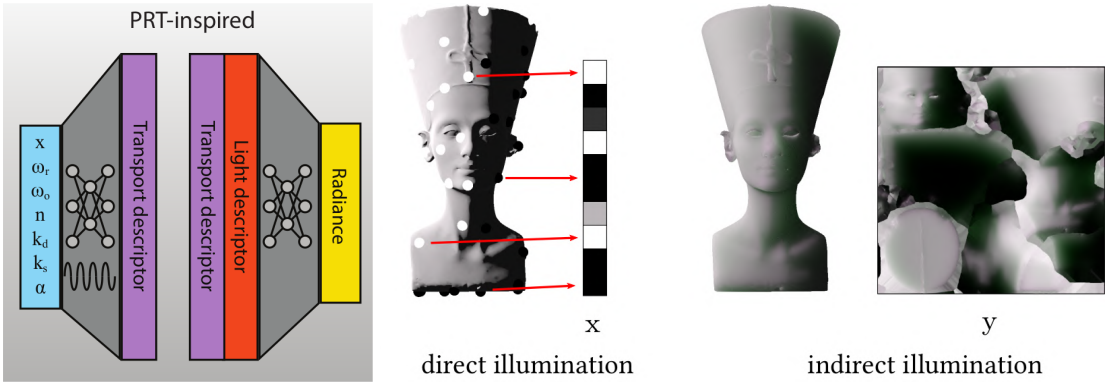


Figure 2.18: *Neural Precomputed Radiance Transfer [RBRD22] (left) is inspired by PRT by encoding the transport and incident radiance using neural networks. The work of Belcour* et al. *[BDBS22] encodes the light transports using basic functions directly deduced from the transports present in the scene,* i.e. *direct and indirect illumination from various luminous configurations.*

set of coefficients independent of the scene size. However, these methods are specifically designed to work with relatively small scenes. Indeed, since these methods encode global light transports, they cannot be used in scenes where the possible lighting combinations are large, which is the case in large scenes.

3 Spherical Light

Surface lights are physically more plausible than area-less lights, however the integration of such light sources requires the consideration of their surface, which makes them particularly complex to consider in high performance rendering contexts. Recently, work has demonstrated an efficient analytical projection of the radiance produced by polygonal lights onto the SH [WR18, WCZR20]. The goal of this chapter is to extend these works to spherical lights (Section 3.1) while reducing the number of arithmetic operations required for the SH projection, allowing a better scalability with respect to the number of coefficients. Our method being more efficient than that of polygonal lights, we propose a method to approximate the projection of their radiance fields on the SH in chapter 4 in order to take advantage of the performances of our method.

Our method to compute the SH projection of the radiance field produced by spherical light requires the evaluation of the SH basis functions, which is done by the efficient Sloan's method [Slo13]. We propose an extension of Sloan's method to efficiently compute the gradient of the SH basis functions (Section 3.2). We leverage this to compute efficiently the gradient of the SH coefficients representing the radiance field produced by spherical lights (Section 3.3). It is the same idea as Wu *et al.* [WCZR20] who compute the gradient of the coefficients for polygonal lights in order to compute the projection only on a sparse grid. The gradient of the SH coefficients allows implementing a more accurate Hermite interpolation than a simple linear interpolation to reconstruct the SH coefficients at each shading points.

Figure 3.1: *San-miguel with only 10 spherical lights whose projection is computed on 10 SH bands. Image rendered in 69ms with the resolution 3400×900.*

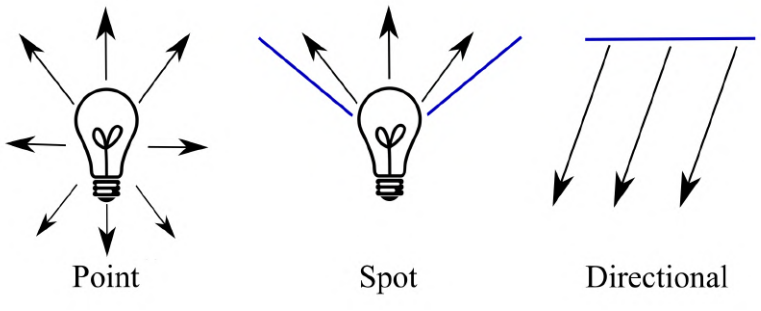


Figure 3.2: *Primitive of simple lights. The emission of each source corresponds to a Dirac distribution.*

3.1 Spherical Light

Spherical lights are a typical example of light sources that we naturally find inside 3D scenes (Figure 3.1). A form factor can be computed to characterize the light contribution of such sources but does not keep the directional distribution information of the incident radiance field. For this purpose, we propose an efficient method to project the radiance field produced by such sources on the spherical harmonics. The performance of this method heavily relies on the evaluation of the spherical harmonics basis functions.

3.1.1 SH projection of simple light primitive

The luminance contribution of a simple light source, such as point light, spot, or directional light, corresponds to a Dirac distribution (Figure 3.2). Thus, the computation of the projection amounts to evaluate the SH in the Dirac direction and scaling by the light source's intensity.

3.1.2 SH projection of spherical light

The projection of a spherical light is more expensive. The framework proposed here aims at handling thousand of them. Zonal Harmonics offers an efficient projection and reconstruction algorithm for spherical lights.

For the hemisphere around a shaded point, the solid angle Q subtended by the light source is a spherical cap (Figure 3.3). Considering the incident luminance as constant, and equal to 1, over Q and 0 elsewhere, the projection on SH of the incident luminance function simplifies to:

$$\mathbf{L}_l^m = \int_Q Y_l^m(\omega)\,\mathrm{d}\omega \ . \tag{3.1}$$

Let ω_l be the direction from the shaded point to the center of the spherical light. Q induces a circularly symmetric luminance field around ω_l. Considering the shading frame with ω_l as z-axis, the computation of Zonal Harmonics boils down to the computation of $\tilde{\mathbf{L}}_l$ coefficients, since the other coefficients are null,

$$\tilde{\mathbf{L}}_l = K_l \int_Q P_l(\omega_l \cdot \omega)\,\mathrm{d}\omega \tag{3.2}$$

where $K_l = K_l^0$ and $P_l = P_l^0$. Using spherical coordinates, this equation becomes

$$\tilde{\mathbf{L}}_l = K_l \int_{\theta=0}^{a} \int_{\phi=0}^{2\pi} P_l(\cos(\theta))\sin(\theta)\,\mathrm{d}\phi\,\mathrm{d}\theta \tag{3.3}$$

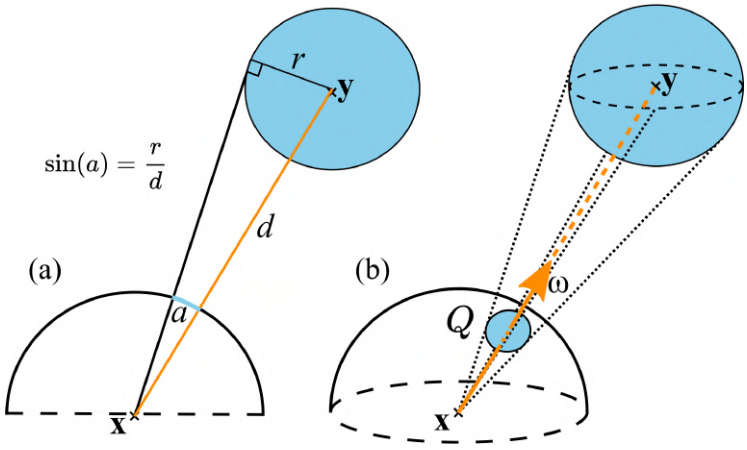


Figure 3.3: *Geometric setup of a spherical light luminance contribution. (a) 2D cut showing the spherical light radius r, the distance d with the spherical light, and the half-angle a of the cap. (b) The spherical light in blue projects to a spherical cap Q on the unit sphere centered at the shading point* $\mathbf{x}$.

where a is the half-angle subtended by the spherical light source (Figure 3.3). Integrating on ϕ, we obtain

$$\tilde{\mathbf{L}}_l = 2\pi K_l \int_{\theta=0}^{a} P_l(\cos(\theta)) \sin(\theta)\, \mathrm{d}\theta \ . \tag{3.4}$$

We search for a closed form to the integral of Legendre polynomials over a spherical cap of angle a:

$$\int_{\theta=0}^{a} P_l(\cos(\theta)) \sin(\theta)\, \mathrm{d}\theta \ . \tag{3.5}$$

Substituting the variable $u = \cos(\theta)$ yields

$$\int_{\theta=0}^{a} P_l(\cos(\theta)) \sin(\theta)\, \mathrm{d}\theta = \int_{u=\cos(a)}^{1} P_l(u)\, \mathrm{d}u \ . \tag{3.6}$$

We get the primitive from the recursive formula

$$\int P_l(x)\, \mathrm{d}x = \frac{P_{l+1}(x) - P_{l-1}(x)}{2l+1} \tag{3.7}$$

and as $\forall\, l \geq 0,\ P_l(1) = 1$ we obtain

$$\begin{aligned} \int_{\cos(a)}^{1} P_l(u)\, \mathrm{d}u &= \left[\frac{P_{l+1}(u) - P_{l-1}(u)}{2l+1} \right]_{\cos(a)}^{1} \\ &= \frac{-P_{l+1}(\cos(a)) + P_{l-1}(\cos(a))}{2l+1} \ . \end{aligned} \tag{3.8}$$

In summary, we have established that

$$\boxed{\int_{\theta=0}^{a} P_l(\cos(\theta)) \sin(\theta)\, \mathrm{d}\theta = \frac{-P_{l+1}(\alpha) + P_{l-1}(\alpha)}{2l+1}} \tag{3.9}$$

where $\alpha = \cos(a)$. By re-injecting this into Equation 3.4, the computation of the zonal harmonics coefficients for a spherical light in the coordinates frame aligned with the emission direction only requires evaluating Legendre polynomials and by simplifying, we obtain

$$\tilde{\mathbf{L}}_l = \begin{cases} \sqrt{\dfrac{\pi}{2l+1}} \left(P_{l-1}(\alpha) - P_{l+1}(\alpha)\right) & \text{if } l \neq 0 \\ \sqrt{\pi}(1-\alpha) & \text{otherwise} \end{cases} \ . \tag{3.10}$$

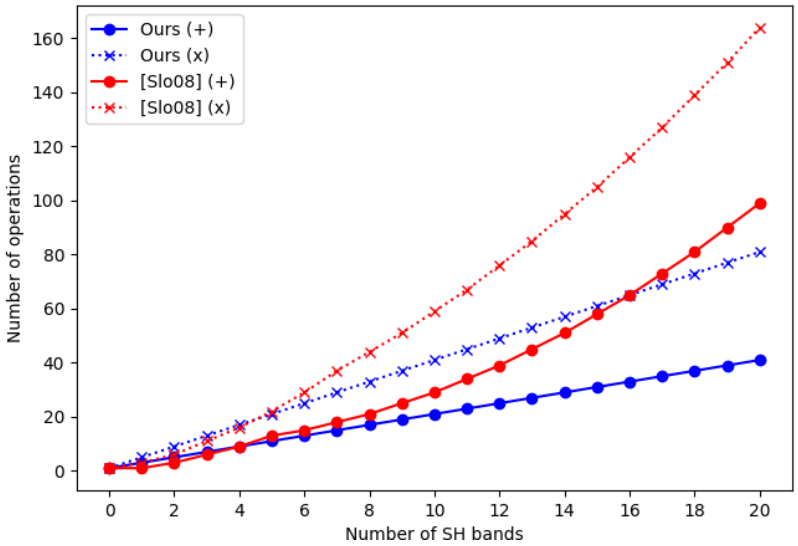


Figure 3.4: *Number of arithmetic operations needed to compute the vector of ZH coefficients up to a certain band. Our method (Equation 3.10) is compared with the analytic Sloan's method [Slo08]. Number of operations calculated assumes that the recurrence loop is unrolled to avoid operations corresponding to constants for each ZH bands (*e.g. $\sqrt{\pi/(2l+1)}$ *counts as one multiplication for each band.)*

The SH coefficients are then obtained by rotating the ZH coefficients (Equation 2.73):

$$\mathbf{L}_l^m = \sqrt{\frac{4\pi}{2l+1}} Y_l^m(\omega_l)\tilde{\mathbf{L}}_l \ . \tag{3.11}$$

As shown in Figure 3.4, beyond 5 SH bands, computing the ZH coefficients $\tilde{\mathbf{L}}_l$ using the closed form derived by Sloan [Slo08] exhibits more arithmetic operations than our general recurrence (Equation 3.10). To evaluate the Y_l^m basis function, we use the work of Sloan [Slo13], that is explained in Section 3.2.

3.1.3 Evaluation of the proposal

We have implemented our model in a prototype of rendering engine using OpenGL / GLSL 4.5 on an Intel Xeon 2.10 GHz processor and an RTX 2080 graphic card.

In our implementation, we did not take into account the visibility and we implemented two representations for the reflectance functions and the associated convolution methods exposed in Section 2.3.2. The SH coefficients of the received radiance field and the convolution with the material are computed at each pixel. To avoid ringing artifacts (Section 2.2.5), we have applied an aggressive filtering by using a window size equal to the number of SH bands used.

Using our formulation, the computation time of the ZH coefficients is very low, dominated by the evaluation of the SH functions (Equation 3.11). Nevertheless, as shown (Figure 3.5 and 3.6), our projection algorithm supports a dozen of spherical lights projected on a dozen of SH bands while allowing a refresh rate compatible with real-time. In the results presented here, the ZH coefficients are computed analytically at each frame (Equation 3.10). However, these coefficients can be easily tabulated since for each band, the ZH coefficients depends only on the cosine of the angle subtended by the spherical light. When

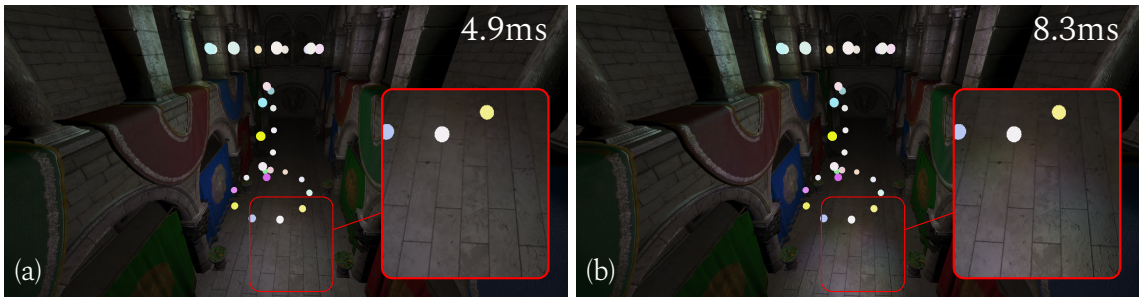


Figure 3.5: *Sponza with 30 spherical sources rendered in 1980×995. Rendered with (a) 5 SH bands and (b) 8 bands. A more precise delineation of the reflections on the ground can be observed, as well as a more important appearance of the colors when the limit band is increased.*

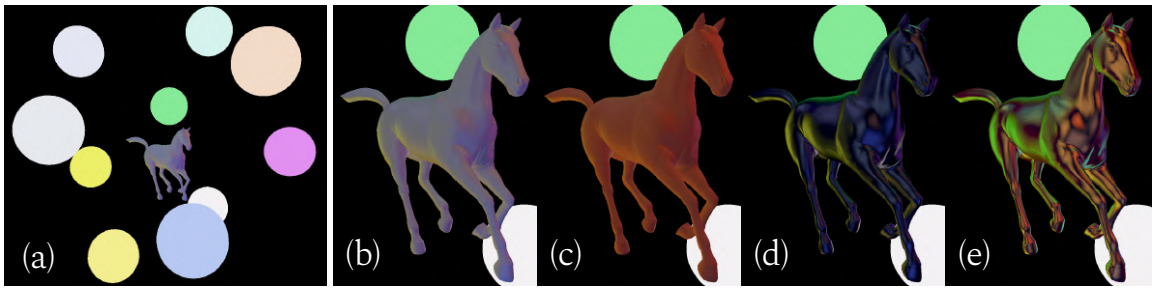


Figure 3.6: *Rendered with different material from the database produced by Dupuy and Jakob [DJ18] and 10 spherical lights, convolution is computed on 10 SH bands. (a) overview of the scene with the material paper_white, (b) ilm_solo_millennium_falcon, (c) acrylic_felt_orange, (d) ilm_solo_m_68 and (e) cc_amber_citrine. Images from (b) to (e) are rendered in 600×816 in less than 3 ms each.*

we tried rendering with the tabulation of the coefficients, we only obtained a time difference of the order of half a millisecond with the analytical version. Our method is therefore interesting for real-time rendering since the additional memory required for storing the tabulated coefficients, although of the order of only a few tens of kilobytes, is not accompanied by a significant time saving. Nevertheless, this is true for the number of SH bands used (between 5 and 8). Beyond that, the tabulation method may be limited by the memory bandwidth creating a non-negligible overhead compared to the analytical computation of the ZH coefficients. In order to reduce the computation time, we have proposed a similar solution to the one discussed by Wang et Ramamoorthi [WR18] and implemented by Wu *et al.* [WCZR20] whose idea is to exploit the smooth spatial variation of the SH coefficients by computing the coefficients and their spatial gradients on a 3D grid and then interpolating them for each shading points (Section 3.3). These works use polygonal lights and a performances comparison between spherical lights and these polygonal lights is brought in Section 3.3.5.

3.2 Translational gradient of the spherical harmonics

Our method to compute the contribution of spherical lights, like many other applications, is based on the evaluation of the spherical harmonics basis functions. Computing them efficiently is therefore not negligible for the performance of these approaches. After explaining the key principle behind the efficient evaluation proposed by Sloan [Slo13], we propose to extend it to compute the spatial gradient of the spherical harmonics basis func-

tion in order to develop in Section 3.3 an efficient method to compute the gradient of the SH coefficients representing the radiance produced by spherical lights.

3.2.1 Efficient spherical harmonics evaluation

In this section, we summarize the key principle proposed by Sloan [Slo13]. The evaluation of the SH basis functions requires the evaluation of the Associated Legendre Polynomials (ALP) P_l^m:

$$P_l^m(\cos\theta) = \sin^m\theta \frac{d^m}{dx^m}(P_l(\cos\theta)) \,. \tag{3.12}$$

Sloan proposes to pair the $\sin^m\theta$ from the ALP with $\sin(m\phi)$ and $\cos(m\phi)$ from the SH expression (Equation 2.30). These pairings lead to computationally simpler recurrence relations:

$$\begin{aligned} U_m &= \sin^m\theta\cos(m\phi) = xU_{m-1} - yV_{m-1} \\ V_m &= \sin^m\theta\sin(m\phi) = xV_{m-1} + yU_{m-1} \,, \end{aligned} \tag{3.13}$$

where U_m corresponds to the case where $m > 0$ and V_m corresponds to $m < 0$, $x = \sin\theta\cos\phi$, and $y = \sin\theta\sin\phi$. As a consequence, $Q_l^m = P_l^m(\cos\theta)/\sin^m\theta$ is a polynomial in z, with $z = \cos\theta$, that does not require the explicit computation of $\sin^m\theta$ and is defined by these five recurrence relations:

$$\begin{aligned} Q_m^m &= (1-2m)Q_{m-1}^{m-1} \\ Q_{m+1}^m &= (2m+1)zQ_m^m \\ Q_l^m &= \frac{(2l-1)zQ_{l-1}^m - (l+m-1)Q_{l-2}^m}{l-m} \\ Q_{m+2}^m &= \frac{(2m+3)(2m+1)Q_m^m z^2 - (2m+1)Q_m^m}{2} \\ Q_{m+3}^m &= \frac{zQ_m^m((2m+5)(2m+3)(2m+1)z^2 - 3(4m^2+8m+3))}{6} \,. \end{aligned} \tag{3.14}$$

where the second and third equations are the same as the traditional recursive formulas for the ALP (respectively Equation 2.29 and 2.27) because both are used after the first equation which divides P_m^m by $\sin^m\theta$. The fourth and fifth equations are not fundamentally necessary to unfold the calculations, but reduce in practice the number of arithmetic operations because these equations are precomputed for each l and m. The same principle could be applied for $m+4$ and beyond but Sloan stopped at $m+3$. In order to improve the final computation, the normalization factors $\sqrt{2}$ and K_l^m of SH (Equation 2.30) should directly be integrated in the recurrence relations of Q, these final forms are given in Section 3.2.2.

3.2.2 Efficient spherical harmonics gradient

Defining the two points $\mathbf{x}$ and $\mathbf{y}$, we seek $\nabla_{\mathbf{y}} Y_l^m(\mathbf{y}-\mathbf{x})$, *i.e.* the translationnal gradient of the spherical harmonics basis functions with respect to $\mathbf{y}$. For an efficient computation, we first perform this computation in spherical coordinates and then make the conversion to Cartesian coordinates by multiplying the gradient computed in spherical coordinates with the Jacobian of the spherical coordinates with respect to the Cartesian coordinates. Since the SH functions vary only in θ and ϕ, we have $\partial_r Y_l^m = 0$, and:

$$\begin{pmatrix} \partial_x Y_l^m \\ \partial_y Y_l^m \\ \partial_z Y_l^m \end{pmatrix} = \begin{pmatrix} \frac{xz}{\Psi} & \frac{-y}{x^2+y^2} \\ \frac{yz}{\Psi} & \frac{x}{x^2+y^2} \\ \frac{-x^2-y^2}{\Psi} & 0 \end{pmatrix} \begin{pmatrix} \partial_\theta Y_l^m \\ \partial_\phi Y_l^m \end{pmatrix} \,, \tag{3.15}$$

where $(x, y, z) = \mathbf{y} - \mathbf{x}$ and $\Psi = (x^2 + y^2 + z^2)\sqrt{x^2 + y^2}$.

Partial derivatives with respect to ϕ

The SH partial derivative with respect to ϕ is :

$$\partial_\phi Y_l^m(\theta, \phi) = \begin{cases} -mY_l^{-m}(\theta, \phi) & m < 0 \\ 0 & m = 0 \\ -mY_l^{-m}(\theta, \phi) & m > 0 \end{cases}, \tag{3.16}$$

and can be computed without overhead when the SH Y_l^m is already evaluated.

Partial derivatives with respect to θ

The SH partial derivative with respect to θ is:

$$\partial_\theta Y_l^m(\theta, \phi) = \begin{cases} \sqrt{2}K_l^{|m|}\mathcal{T}\sin(|m|\phi) & m < 0 \\ K_l^0\mathcal{T} & m = 0 \\ \sqrt{2}K_l^m\mathcal{T}\cos(m\phi) & m > 0 \end{cases}. \tag{3.17}$$

where $\mathcal{T} = \partial_\theta P_l^{|m|}(\cos\theta)$. Equation 3.17 might be computed directly, giving the thereafter called ***Direct*** method. This method improves the one proposed in [LTL+19] using a less expensive formula evaluating the derivatives of the ALP based on the recursion formula given by Hansen [Han88]:

$$\mathcal{T} = \begin{cases} P_l^1(\cos\theta) & m = 0 \\ -\frac{1}{2}\{(l-m+1)(l+m)P_l^{m-1}(\cos\theta) - P_l^{m+1}(\cos\theta)\} & m > 0 \end{cases}. \tag{3.18}$$

We however propose a more efficient computation method, called the ***Sloan's like*** method, based on the same principle as the one used for the SH basis function computation (Section 3.2.1). Instead of computing $\mathcal{T}$, we compute $\mathcal{T}/\sin^m\theta$ to remove $\sin^m\theta$ in other SH terms. When deriving $\mathcal{T}$ and factorizing to have a $\sin^m\theta$ factor, we obtain:

$$\frac{\mathcal{T}}{\sin^m\theta} = \frac{P_l^m(\cos\theta)}{\sin^m\theta}\frac{m\cos\theta}{\sin\theta} + \partial_\theta\Big(\frac{P_l^m(\cos\theta)}{\sin^m\theta}\Big). \tag{3.19}$$

The first term of the right member of Equation 3.19 is exactly Q_l^m corresponding to the recurrence relations given by Sloan (Section 3.2.1) and the last term is $\partial_\theta Q_l^m = T_l^m$. We can thus rewrite Equation 3.19 as:

$$\frac{\mathcal{T}}{\sin^m\theta} = Q_l^m\frac{m\cos\theta}{\sin\theta} + T_l^m. \tag{3.20}$$

To compute T_l^m, we derive the recurrence relations of Q_l^m (Equation 3.14):

$$\begin{aligned} T_m^m &= 0 \\ T_{m+1}^m &= -(2m+1)(\sin\theta Q_m^m) \\ T_l^m &= \frac{(2l-1)(\cos\theta T_{l-1}^m - \sin\theta Q_{l-1}^m) - (l+m-1)T_{l-2}^m}{l-m} \\ T_{m+2}^m &= -(2m+3)(2m+1)(\sin\theta\cos\theta Q_m^m) \\ T_{m+3}^m &= \frac{(\sin\theta Q_m^m)((-(2m+5)(2m+3)(2m+1)\cos^2\theta) + (4m^2+8m+3))}{2}. \end{aligned} \tag{3.21}$$

Final forms We need to integrate the normalization factors of SH in the recurrences relations Q (Equation 3.14) and T (Equation 3.21) to obtain their final efficient forms. Let's write:

$$J_l^m = \begin{cases} \sqrt{2}K_l^m & m > 0 \\ K_l^0 & m = 0 \end{cases}, \quad \begin{aligned} X_l^m &= J_l^m Q_l^m \\ Z_l^m &= J_l^m T_l^m \end{aligned} . \tag{3.22}$$

Multiplying Equation 3.20 by the normalization factor J_l^m, it gives us:

$$J_l^m \frac{\mathcal{T}}{\sin^m \theta} = X_l^m \frac{m \cos\theta}{\sin\theta} + Z_l^m . \tag{3.23}$$

Then, we extend the recurrence relations Q to integrate the normalization factors.

$$\begin{aligned} X_m^m &= J_m^m (1-2m) Q_{m-1}^{m-1} \\ X_{m+1}^m &= J_{m+1}^m (2m+1) \cos\theta Q_m^m \\ X_l^m &= \frac{(2l-1)\cos\theta X_{l-1}^m \frac{K_l^m}{K_{l-1}^m} - (l+m-1) X_{l-2}^m \frac{K_l^m}{K_{l-2}^m}}{l-m} \\ X_{m+2}^m &= J_{m+2}^m \frac{(2m+3)(2m+1)Q_m^m \cos^2\theta - (2m+1)Q_m^m}{2} \\ X_{m+3}^m &= J_{m+3}^m \frac{\cos\theta Q_m^m ((2m+5)(2m+3)(2m+1)\cos^2\theta - 3(4m^2+8m+3))}{6} . \end{aligned} \tag{3.24}$$

These relations are the final forms developed in the code given by Sloan [Slo13]. The final recurrences for the derivatives T are then:

$$\begin{aligned} Z_m^m &= 0 \\ Z_{m+1}^m &= -J_{m+1}^m (2m+1)(\sin\theta Q_m^m) \\ Z_l^m &= \frac{(2l-1)(\cos\theta Z_{l-1}^m \frac{K_l^m}{K_{l-1}^m} - \sin\theta X_{l-1}^m \frac{K_l^m}{K_{l-1}^m}) - (l+m-1) Z_{l-2}^m \frac{K_l^m}{K_{l-2}^m}}{l-m} \\ Z_{m+2}^m &= -J_{m+2}^m (2m+3)(2m+1)(\sin\theta\cos\theta Q_m^m) \\ Z_{m+3}^m &= J_{m+3}^m \frac{(\sin\theta Q_m^m)((-(2m+5)(2m+3)(2m+1)\cos^2\theta) + (4m^2+8m+3))}{2} . \end{aligned} \tag{3.25}$$

Evaluation and validation of our methods

To compare the performance of our two methods, ***Direct*** and ***Sloan's like***, we perform several unit tests on CPU (Table 3.1), with a reference evaluation of the SH functions and their gradients using the polynomial form of SH. The goal is to compare the mean time to evaluate the SH coefficients and their gradient in *one* direction. This comparison uses the same validation procedure than the one proposed by Sloan [Slo13]. For each method, the mean computation time is measured using the same set of thousand directions. For each method, timings are averaged over a hundred executions for each direction. We observe that both the ***Direct*** and the ***Sloan's like*** methods are faster than the polynomial form when having 3 SH bands or more, including the Cartesian conversion. Also, the ***Sloan's like*** method is always faster than the ***Direct*** method. The gradient of the SH basis function obtained by the ***Sloan's like*** method is correct, as demonstrated by Figure 3.9.

Algorithm		Band limit					
		3	4	5	6	8	10
Polynomial	Eval	27.82	32.22	40.57	54.63	84.51	144.01
	Gradient $\nabla_{x,y,z}$	56.91	83.34	122.28	190.25	386.91	720.26
	Sum	84.74	115.56	162.85	244.87	471.42	864.27
Eval & Gradient $\nabla_{\theta,\phi}$	***Sloan's like*** (ours)	25.86	33.77	44.51	59.95	106.35	169.82
	Direct (ours)	42.28	60.66	81.15	112.32	196.33	296.94
Cartesian conversion		41.63	43.67	48.97	53.82	66.19	84.41
Eval & Gradient $\nabla_{x,y,z}$	***Sloan's like*** (ours)	67.50	77.43	93.48	113.78	172.55	254.23
	Direct (ours)	83.92	104.32	130.13	166.14	262.52	381.35

Table 3.1: *Timings (in ns, on the CPU) comparison to evaluate SH and their corresponding gradients at different orders. Both our methods are efficient compared to the traditional Cartesian approach. However, our* **Sloan's like** *approach is the most efficient, achieving a speedup of more than ×3 on 10 SH bands. Our gradients expressed with respect to the spherical direction are converted to obtain the Cartesian spatial gradients (Section 3.2.2). For both our methods, SH are evaluated with Sloan's method [Slo13] and are not separated from the gradient calculations because ∇_ϕ is directly calculated from the SH evaluation (Equation 3.16).*

3.3 Translational gradient of the projected radiance field

We want to compute the gradient of the coefficients of the SH representing the radiance produced by spherical light. The efficiency of this method is demonstrated in an application of interpolation of the SH coefficients. The SH coefficients (and their gradient) are computed only on a sparse grid and interpolated at each shaded point (Figure 3.7). The gradient allows a more accurate interpolation than a simple linear interpolation. This method can handle several hundreds of dynamic lights in real-time.

3.3.1 Formulation

We denote $\nabla_{\mathbf{x}}\mathbf{L}_l^m(\mathbf{x})$ the spatial gradient of the SH coefficients (Equation 3.11) with respect to the shaded point $\mathbf{x}$ (Figure 3.3). It is defined as:

$$\nabla_{\mathbf{x}}\mathbf{L}_l^m(\mathbf{x}) = -\sqrt{\frac{4\pi}{2l+1}}\Big(\underbrace{\nabla_{\mathbf{y}}Y_l^m(\omega)}_{\text{Section 3.2.2}}\ \overbrace{\tilde{\mathbf{L}}_l(\mathbf{x})}^{\text{Section 3.1.2}} + \overbrace{Y_l^m(\omega)}^{\text{Section 3.2.1}}\ \underbrace{\nabla_{\mathbf{y}}\tilde{\mathbf{L}}_l(\mathbf{x})}_{\text{Section 3.3.2}}\Big). \tag{3.26}$$

$\nabla_{\mathbf{x}}\mathbf{L}_l^m(\mathbf{x})$ represents the gradient according to the translation of a point $\mathbf{x}$. Its computation involves the evaluation of the gradient $\nabla Y_l^m(\omega)$ of the SH basis functions. To compute the latter, it is more convenient to consider that the point $\mathbf{y}$ moves: moving $\mathbf{x}$ along a direction $\mathbf{d}$ is equivalent to move $\mathbf{y}$ along $-\mathbf{d}$. This is what introduces the minus sign and the gradient $\nabla_{\mathbf{y}}$ in Equation 3.26.

The definition of $\nabla_{\mathbf{x}}\mathbf{L}_l^m(\mathbf{x})$ contains four main terms: the SH functions $Y_l^m(\omega)$, the ZH projection of the spherical light $\tilde{\mathbf{L}}_l(\mathbf{x})$, and their respective gradients $\nabla Y_l^m(\omega)$ and $\nabla\tilde{\mathbf{L}}_l(\mathbf{x})$. An efficient computation of the gradient $\nabla Y_l^m(\omega)$ of the SH basis functions was proposed earlier (Section 3.2.2). In the following, the computation of the gradient $\nabla\tilde{\mathbf{L}}_l(\mathbf{x})$ of ZH coefficients is derived first (Section 3.3.2).

It should be noted that $\tilde{\mathbf{L}}_l(\mathbf{x})$ and $Y_l^m(\omega)$ are two functions that could be written in polynomial Cartesian form for each order and degree [Slo08]. If doing so, computing the spa-

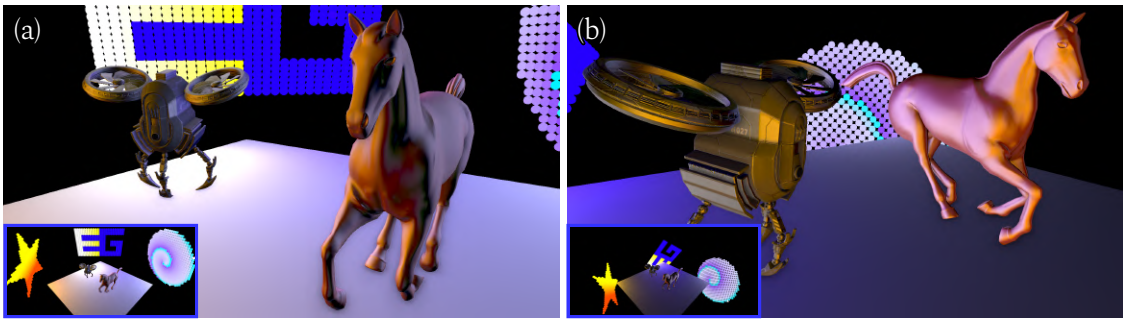


Figure 3.7: *This fully dynamic scene is rendered at 77 fps with 1793 lights in an image of resolution* 1920×1080*. This is achieved by introducing new analytic formulae that efficiently computes the spherical harmonics (SH) gradients for lighting with uniform spherical lights. We integrate our new computations in a SH based rendering system able to compute the lighting from several hundreds of light sources in real time. This rendering system first computes the SH coefficients and their gradients on a 3D grid, and then compute the shading at each visible point by interpolating the samples grid. Our formulae reduce the gradients computational cost and allow to: increase the frame rate, increase the number of SH bands, use more lights and/or increase the 3D grid resolution.*

tial gradient would essentially be a matter of differentiating each equation. However, our method is based on a more efficient recursive analytical equations than these polynomial forms (Table 3.1 and 3.2).

3.3.2 Gradient of the zonal harmonics coefficients

Our analytical definition of the gradient of the ZH coefficients is derived using the chain rule as follows.

Subtended angle gradient Recalling that the half-angle subtended by a spherical light is

$$a(\mathbf{x}) = \arcsin\left(\frac{r}{\|\mathbf{y}-\mathbf{x}\|}\right). \tag{3.27}$$

The gradient of this half-angle with respect to to the position of the spherical light $\mathbf{y}$ can be written as:

$$\nabla_{\mathbf{y}} a(\mathbf{x}) = \frac{-r}{\|\mathbf{y}-\mathbf{x}\|\sqrt{\|\mathbf{y}-\mathbf{x}\|^2 - r^2}} \frac{\mathbf{y}-\mathbf{x}}{\|\mathbf{y}-\mathbf{x}\|}. \tag{3.28}$$

Angle gradient for an oriented disk

If we change the definition of the half-angle by the one subtended by a disk whose normal would always be directed towards the shaded point, the half-angle its gradient become respectively

$$a_{disk}(\mathbf{x}) = \arctan\left(\frac{r}{\|\mathbf{y}-\mathbf{x}\|}\right), \qquad \nabla a_{disk}(\mathbf{x}) = \frac{-r}{r^2 + \|\mathbf{y}-\mathbf{x}\|^2} \frac{\mathbf{y}-\mathbf{x}}{\|\mathbf{y}-\mathbf{x}\|} \tag{3.29}$$

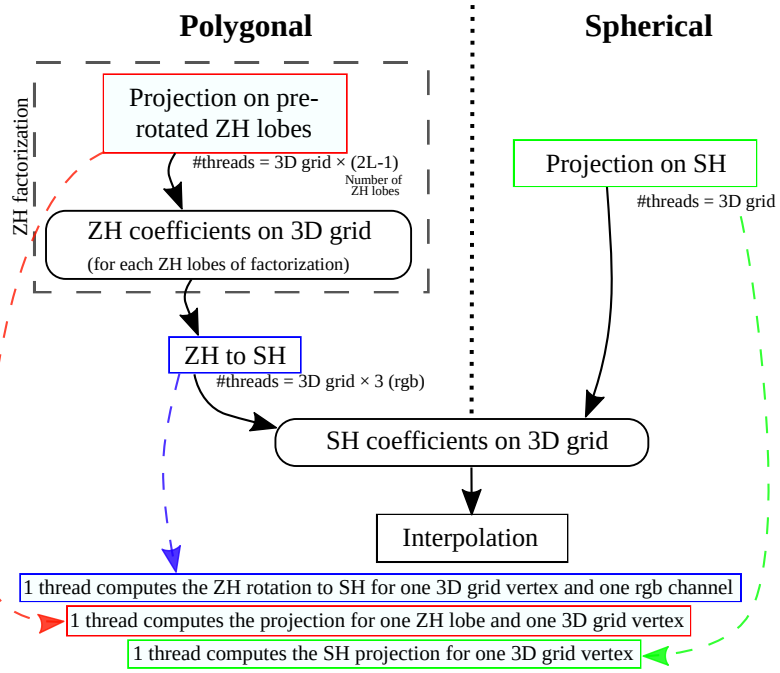


Figure 3.8: *Parallelization scheme for polygonal (left) and spherical lights (right). The projection of polygonal lights relies on the ZH factorization [NSF12], which is efficiently parallelized and provide a good tradeoff between the number of threads and the register pressure. Our method cannot follow the same parallelization model, another factorization method is proposed to lower the pressure on the registers (Section 3.3.3).*

Gradient of the ZH coefficients Given Equation 3.10, the gradient of the ZH coefficients corresponding to the projection of a spherical light is:

$$\nabla_{\mathbf{y}}\tilde{\mathbf{L}}_l(\mathbf{x}) = \begin{cases} -\sqrt{\frac{\pi}{2l+1}}\nabla_{\mathbf{y}}a(\mathbf{x})\sin(a(\mathbf{x}))O'_l & \text{if } l \neq 0 \\ \sqrt{\pi}\nabla_{\mathbf{y}}a(\mathbf{x})\sin(a(\mathbf{x})) & \text{otherwise} \end{cases}, \tag{3.30}$$

where $O'_l = (-P'_{l+1}(\alpha) + P'_{l-1}(\alpha))$. Considering the derivative of the Legendre polynomials:

$$\partial_x P_l(x) = -\frac{(l+1)(xP_l(x) - P_{l+1}(x))}{x^2 - 1}, \tag{3.31}$$

we obtain:

$$O'_l = \frac{(l+2)(\alpha P_{l+1}(\alpha) - P_{l+2}(\alpha)) - l(\alpha P_{l-1}(\alpha) - P_l(\alpha))}{\alpha^2 - 1}. \tag{3.32}$$

3.3.3 Efficient parallel grid construction

The method of Wu *et al.* [WCZR20] relies on the ZH factorization (Section 2.2.9). The projection is done on a set of independent ZH lobes whose directions are fixed for all lights. Each ZH lobe can be processed in parallel, which allows a good distribution of the workload (Figure 3.8) on the GPU. In our approach, each light defines a ZH lobe direction, which makes the workload difficult to distribute over the different threads, thus increasing the pressure on registers, especially when increasing the SH bands.

To better parallelize the computations, we reverse the main loops of the algorithm. First, as our SH recurrence relations do not use relations between two different degrees m, except for the opposite degree $-m$, we only need to fix a degree m and loop through the orders

SH bands limit	Spherical lights				Polygonal lights [WCZR20]
	Analytical		Ours		
	forward	reverse	forward	reverse	
5	4.48	4.32	4.08	2.22	7.21
6	7.25	6.37	5.98	2.96	13.08
7	13.11	8.54	7.24	3.77	16.21
8	19.30	10.41	12.73	4.76	29.07
9	33.23	12.91	22.26	5.18	164.01

Table 3.2: *Timings (in ms) for the computation of the SH and gradients coefficients in a 8^3 3D grid, using 648 lights. Columns labeled "analytical" correspond to the SH computations with their Cartesian form, ZH projection is computed in closed form [Slo08] and gradients are computed with analytical derivation. Reverse columns correspond to the inversion of the loops (Section 3.3.3). Our method is much more efficient to increase the SH band limit compared to the reference method for polygonal lights.*

l. So, the natural approach to compute the SH coefficients and their gradients for all the lights, denoted as *Forward* in the following, is to loop first over the lights, and then over all the degrees m and orders l of SH. As we need to accumulate the SH coefficients for all the lights, this *Forward* strategy needs to keep in GPU registers all the SH coefficients. This requires so many temporary storage that the parallel workload between GPU cores is limited. To improve parallel workload on the GPU, we permute the loops on the lights and on degree m, denoted as *Reverse* in the following. Doing this, we only need to keep in registers the SH coefficients deduced for a given degree m, decreasing the register pressure.

Forward (natural) → Reverse (ours)

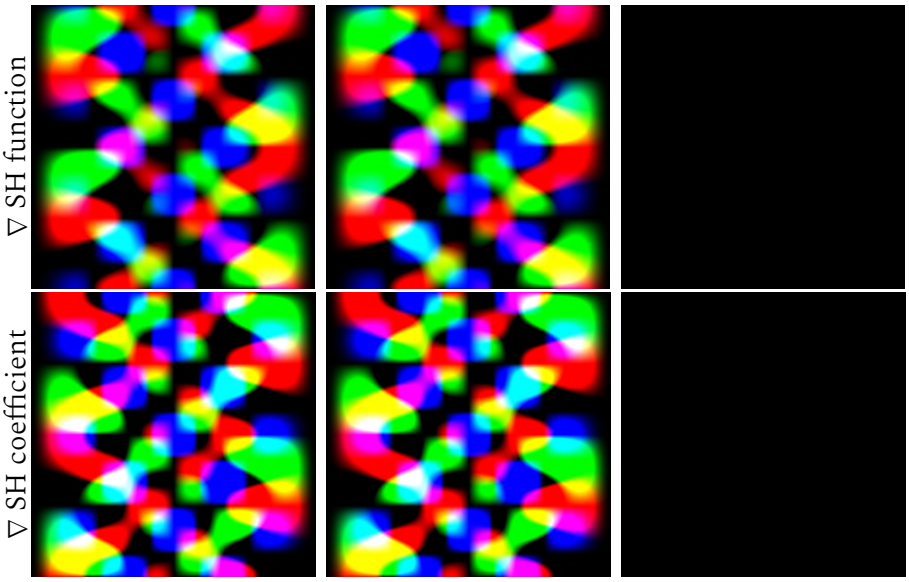
```
FOR(∀ lights)                FOR(∀m)
    FOR(∀m)          →           FOR(∀ lights)
        FOR(∀l)                      FOR(∀l)
```

This improves the overall performance of our method, especially when increasing the SH band limit (Table 3.2). The drawback is that, for each light, we have to recompute all the terms that depend on m. Despite that, the register pressure drop is such that the extra cost of calculation is largely absorbed by a better usage of GPU cores.

3.3.4 Interpolation and rendering pipeline

The rendering pipeline performs two independent passes at each frame (Section 2.3.4). The first pass computes SH coefficients and their gradients at each vertex of a 3D grid enclosing the scene. The second pass interpolates the SH coefficients and computes the shading at each visible fragment.

Our rendering pipeline is a clone of the one proposed by Wu *et al.* [WCZR20] in which polygonal lights are replaced by spherical ones. As they do, we can use any of the three interpolations they propose to reconstruct the SH coefficients from the grid values (the linear interpolation, the Taylor interpolation [AKDS04] and the Hermite interpolation [WCZR20]). The final shading is computed with the SH framework proposed in Section 4.1. We skip the visibility in our computations, and add a pass of SSAO [BS08] to compensate for the lack of shadows. We also show in Figure 3.16 that our method is versatile and can, for instance, be applied with captured data-driven materials following the principles given in Section 2.3.2.



(a) Analytical *(b)* Ours *(c)* 100× Abs. diff.

Figure 3.9: *Validation of our method to compute the gradients of SH basis functions (top line, Section 3.2.2) and the gradients of SH coefficients for spherical lights (bottom line, Equation 3.26) using* $(l, m) = (6, 3)$. *Column (a) computed using the differentiation of the analytical equations [Slo08](Section 3.3.1) and (b) computed with our methods. We show in (c) the difference between (a) and (b) to highlight the accuracy of our methods. Spatial gradients are encoded as colors such that each pixel corresponds to a different pair* (θ, ϕ), θ *varies according to the columns, and* ϕ *varies according to the rows of each image. The gradients for the SH coefficients are computed using a spherical light at a distance of 2 and a radius equal to* 0.5.

3.3.5 Evaluation of the proposal

We implemented our method in an experimental rendering engine (Section 3.3.4) using OpenGL / GLSL 4.5 running on an Intel Xeon 2.10 GHz processor and a RTX 2080 NVIDIA graphics card. For our experiments and comparisons, we use the same band limitations as Wu *et al.* did for the polygonal lights [WCZR20]. Lighting is band-limited on 5 SH bands and the convolution with the GGX specular lobes are computed on 9 SH bands. In the following, we denote as "SH reference" the results obtained *without interpolation*. They are obtained by computing the exact projection of the radiance on SH at each pixel.

Quality As pointed out in the overview (Section 3.3.1), the SH gradient's coefficients for spherical lights (Equation 3.26) can be computed with Cartesian analytic equations for each SH. While being as accurate as these Cartesian analytical forms (Figure 3.9), our recursive analytical equations are faster to evaluate (Table 3.2).

From the three interpolation schemes (trilinear, Taylor and Hermite), the Hermite interpolation is, as demonstrated by Wu *et al.* [WCZR20], the more accurate and the accuracy increases as the grid is denser (Figure 3.10). When an equal time is allocated to generate the SH coefficients and their gradients on a 3D grid, or to generate only the coefficients on a 3D grid, the latter grid can be denser. Nevertheless, linear interpolation on such a grid does not reach the quality provided by the Hermite interpolation on a sparser grid (Figure 3.11) while using, in this case, 3.35× more memory to store the grid data.

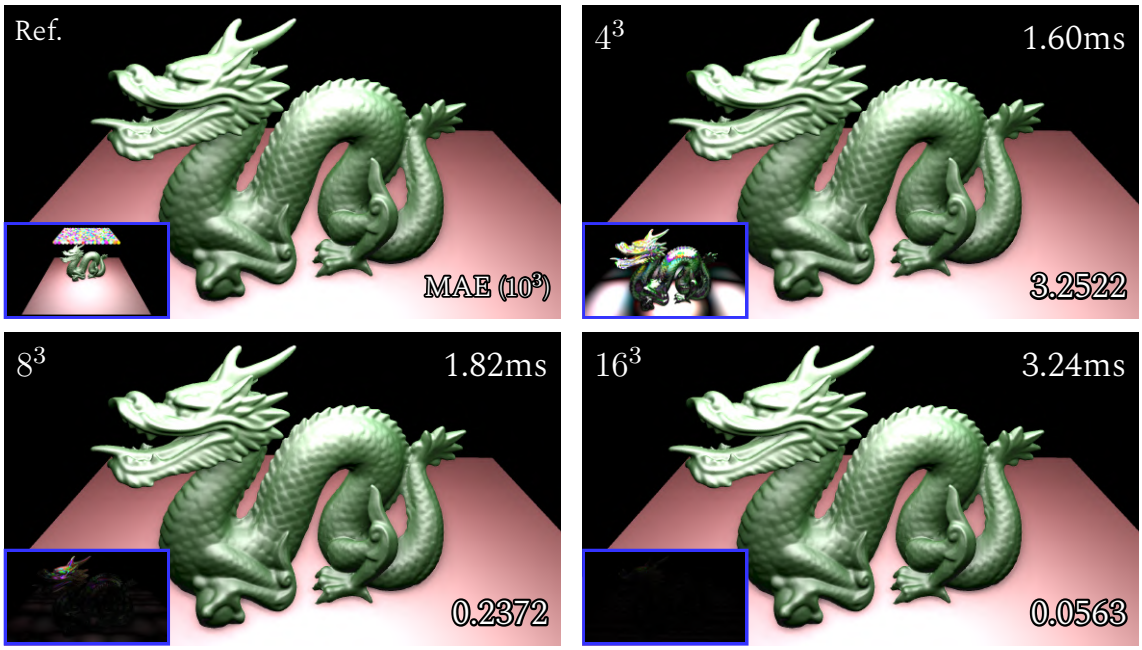


Figure 3.10: *Results obtained by varying the 3D grid resolution with Hermite interpolation and 578 lights. Timings at the top right of each image indicate the computation time of the SH coefficients and their gradients on the 3D grid. The reference is computed with SH without interpolation. The inset of the reference image shows the scene configuration and other insets show* 100× *absolute errors.*

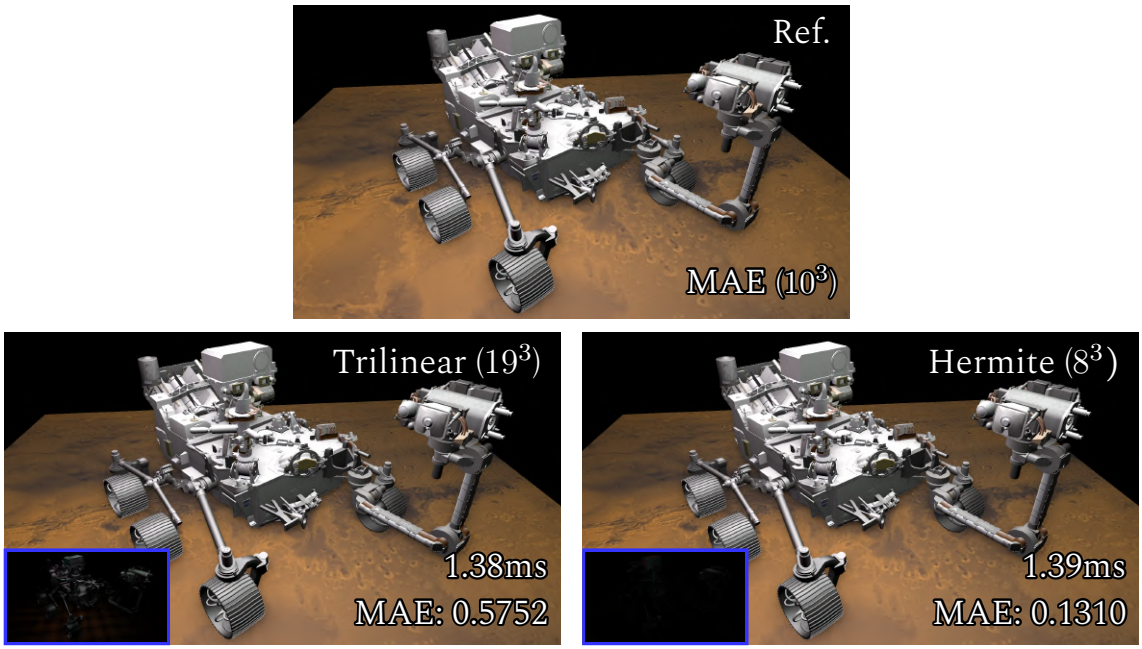


Figure 3.11: *Equal-time grid generation comparison using 512 lights. The same time is required to compute a* 19^3 *3D grid without gradient (for trilinear interpolation) (bottom-left) and a* 8^3 *3D grid with gradients (for Hermite interpolation) (bottom-right). When trilinearly interpolating the very large grid is still not dense enough to surpass the quality of the Hermite interpolation. Insets show* 40× *absolute errors.*

Scene	Figure	Lights	Output resolution	Eval SH coeff.	Eval coeff & grad.	Interp.	ssao	Total	Reference		
									Time	Speed up	MAE ($\times 10^3$)
Drone & Horse	Figure 3.7a	1793	1920 × 1080	3.18	5.30	6.73	0.9	12.93	139.11	10.8×	0.2022
Dragon	Figure 3.10	578	1920 × 1080	0.84	1.82	12.41	0.98	15.21	76.54	5.0×	0.2372
Perseverance	Figure 3.11	512	1920 × 1080	0.86	1.39	10.17	1.06	12.62	55.40	4.4×	0.1310
Armadillo & Buddha	Figure 3.12	242	1000 × 1000	0.34	0.66	10.51	0.39	11.56	30.50	2.6×	0.1920
Conference	Figure 3.13a	900	1920 × 1080	1.49	2.63	11.53	0.77	15.65	115.51	7.4×	0.9656
Living Room	Figure 3.13b	2500	1920 × 1080	4.36	7.84	11.30	0.88	20.02	325.07	16.2×	0.7445
Fertility	Figure 3.16	800	1920 × 735	1.46	2.26	28.56	0.65	31.47	133.01	4.2×	0.0340
White Room	Figure 4.8	722	1920 × 1080	1.19	2.07	14.58	1.21	17.86	119.49	6.7×	0.0779

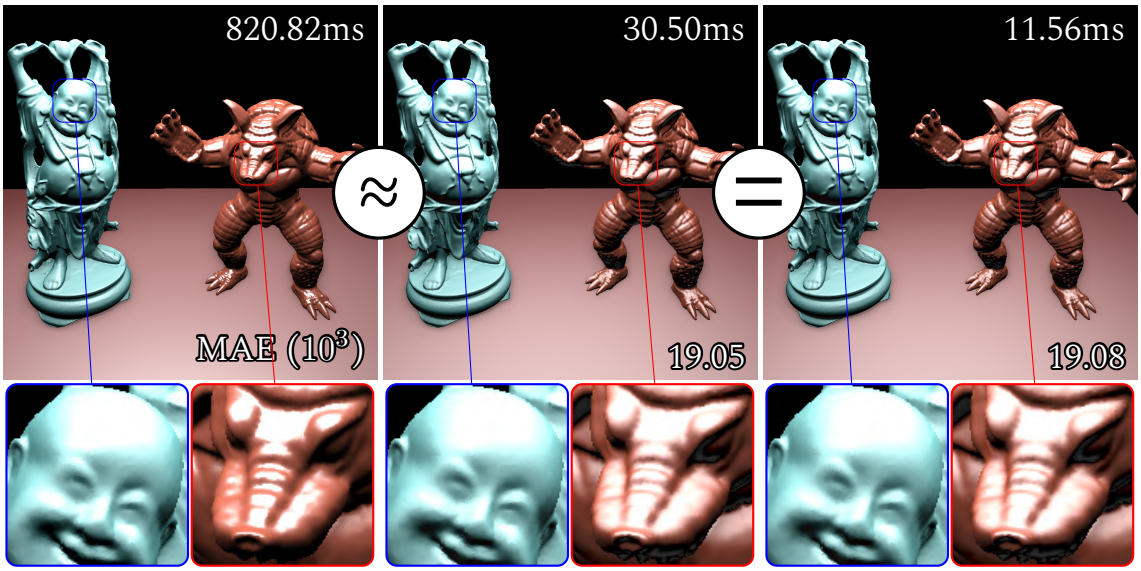
Table 3.3: *Timings in ms obtained on our different scenes using a 8^3 3D grid. The rendering without interpolation is used as reference. We give the computation times for both the coefficients and the coefficients plus their gradients to evaluate the extra cost of gradient computations.*

As shown in Figure 3.12, reconstructing the SH coefficients with Hermite interpolation from grid samples generates results that closely match the per-pixel direct computation of SH coefficients. We also illustrate a benefit of using spherical lights when representing textured lights in Figure 3.13. In this case, a spherical light is placed at each texture pixel, and the results always closely match the SH reference (Table 3.3).

Performance We present in Table 3.3 the results produced on different scene configurations. We observe that with ≈ 500 lights we obtain an acceleration of $\approx 5\times$ compared to the SH reference. This acceleration is even higher when the number of sources increases. On average, we obtain an overhead of $\approx 1.8\times$ to evaluate the gradient of the coefficients in addition to the computation of the coefficients themselves. This shows that, as depicted in the previous sections (Equation 3.16), we are able to reuse temporary results between the computations of the coefficients and their gradients.

In our renderings, the light is projected on a full set of SH and the material is an isotropic GGX-based microfacet model projected only on ZH. The convolution between light and material is computed respectively between a function projected on SH (for the light) and another only on ZH (for the material) (Section 2.3.2). When interpolating SH coefficients in a grid, the radiance function cannot be reduced to a ZH representation. We thus cannot take advantage of the fast convolution computation between two ZH proposed in Section 5.2.7. Despite this, the gain provided by the grid interpolation over the direct projection of light sources pays off, and the overhead produced by the convolution SH by ZH becomes negligible as the number of lights increases (Figure 3.14).

In addition, the convolution between two ZH functions has multiple inconveniences. Firstly, the cosine of the rendering equation is approximated by the cosine between the



(a) Ray casting *(b) SH reference* *(c) Ours*

Figure 3.12: *Comparison against ray casting with a Lambertian Buddha, a glossy Armadillo and 242 lights. As demonstrated for polygonal lights [WCZR20], our approximation with interpolation (c) almost perfectly match the result without interpolation (b). Both results are close to the ray cast reference. Looking at the armadillo, results only differ on glossy material due to the order limit on SH approximation. The ray cast result is generated using 100 importance samples for each light.*

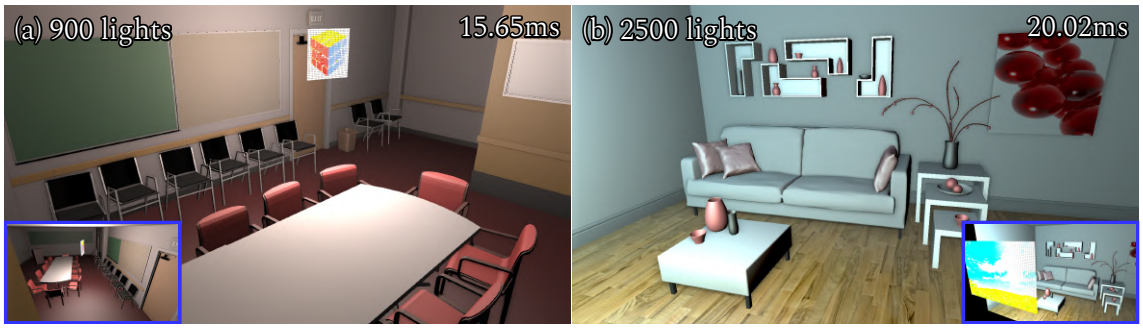


Figure 3.13: *Textured lights rendering with 900 lights (a) and 2500 lights (b). The insets illustrate the scene configurations.*

Compared method	Figure 3.12			Figure 4.8
	Number of lights			
	288	512	800	722
Comparison with **spheres** (Equation 3.11)				
Spheres (ZH approx.) (Section 5.2.7)	1.866	1.865	1.865	1.540
Spheres interp. (Equation 3.26)	0.192	0.192	0.193	0.079

Table 3.4: *MAE* (10^3) *with different scene configurations with an interpolation in a* 8^3 *3D grid.* ***Bold text*** *highlights rendering without interpolation. The comparisons show that the interpolation for spherical sources produces a better approximation than using the ZH acceleration (Section 5.2.7) that approximate the cosine in the lighting equation.*

surface normal and the direction to the center of the sphere, which significantly affects the Mean Absolute Error (MAE) (Table 3.4). Secondly, we cannot use this method with arbitrary materials (Section 2.3.2) (Figure 3.16) since SH coefficients for the BRDF cannot be directly reduced to ZH.

Performance comparison with polygonal lights The purpose of this paragraph is to bring a quick comparison of performances between spherical lights and polygonal lights, without interpolation [WR18] and with interpolation [WCZR20] for both types of lights. Section 4.2 discuss two methods to approximate the SH projection of the radiance field produced by polygonal lights with spherical lights, therefore a more in-depth comparison is made at this point. For simplicity, we only compare spherical lights with triangular lights and for a fair comparison a similar radiance field is produced by distributing the spherical light according to the geometry of the triangles following the method exposed in Section 4.2.2. Also, to produce a similar radiance field regardless of the number of lights, the triangles are broken into smaller ones to increase the number of lights and the spherical lights are redistributed according to the new triangles. Thus, with or without interpolation, the SH projection of the radiance field produced by spherical lights (as well as the gradient of the SH coefficients) is more efficient per light than the methods for polygonal lights (Figure 3.14 and 3.15a). In addition, using a larger interpolation grid or increasing the SH band limit is much less expensive with spherical lights than with polygonal lights (Figure 3.15b and Table 3.2).

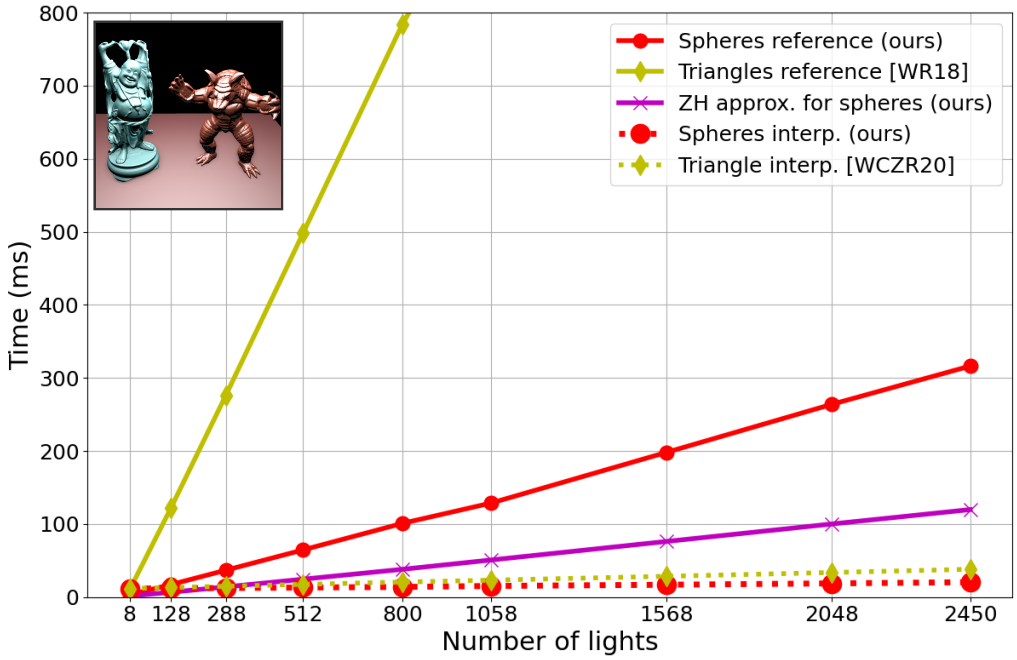


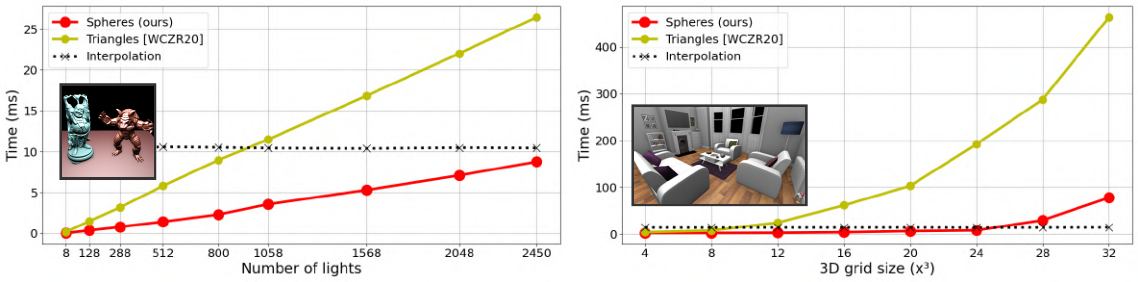
Figure 3.14: *Timings (in ms) to compute the final shading,* i.e. *SH coefficients and gradients plus interpolation, on a 8^3 3D grid using different number of lights on Figure 3.12. We compare spherical lights, triangular lights and our approximation of triangular lights, with and without interpolation. References correspond to the computation without interpolation. The ZH approximation for spheres proposed in Section 5.2.7 cannot be used with interpolation but efficiently approximates spherical light at low cost.*

3.4 Limitations and future work

Our methods to compute the SH coefficients representing the radiance field produced by spherical lights face the limitations inherent to the SH representation. Some of the limitations listed here are similar to those found for polygonal lights [WR18, WCZR20].

Analytical and numerical integration As evoked by Wu *et al.* [WCZR20], it could be envisioned to switch between analytic and numerical integration to compute SH coefficients (with or without their gradients). Indeed, when lights are far away from the shaded point, only one sample should be sufficient for numerical integration. Moreover, importance sampling of spherical lights is well studied in literature [DHB17]. However, in practice, sampling numerical materials, *i.e.* data-driven materials (Appendix B), on the GPU (Figure 3.16) is difficult and often very expensive. Using SH thus remains a pertinent solution as SH allow the use of data-driven materials with only a very small overhead.

Textured and anisotropic lights Our methods handles textured light in a very simple manner, by replacing each texel by a spherical light. However, neighboring texels in the texture may have a similar color, which introduces redundancy in the spherical light colors. This redundancy can be used to drastically reduce the number of required spherical lights, e.g. by building a quadtree approximating the texture and placing a spherical light at the center of each leaf. Being orthogonal to our contributions, this is left for future work. Considering anisotropic lights, *i.e.* angularly non uniform emissive lights, the SH projection, as the gradient, should be adapted to take into account the radiance variations within the solid angle subtended by the spherical source (Figure 3.3).



(a) Timings according to number of lights *(b) Timings according to grid size*

Figure 3.15: *Timings (in ms) comparison between spherical light and triangular light. (a) Timings to compute SH coefficients and their gradients in a 8^3 3D grid using different number of lights on Figure 3.12. (b) Timings to compute SH coefficients and their gradients for different 3D grid resolutions on Figure 4.8 with 722 lights.*

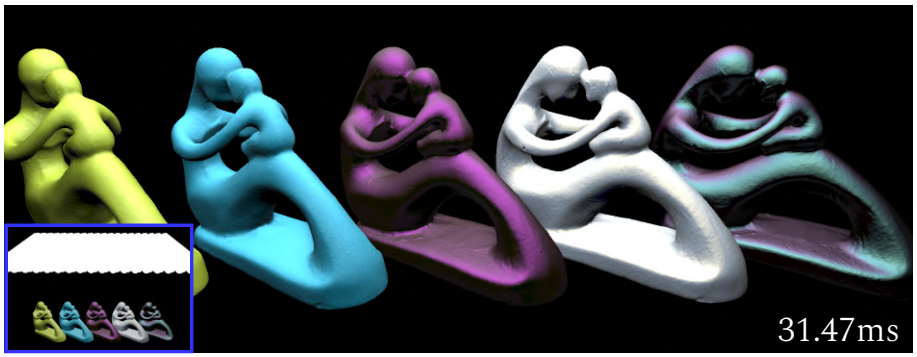


Figure 3.16: *Rendering using a 8^3 3D grid for interpolation with captured materials from [DJ18]. From left to right: cm_toxic_green, paper_blue, vch_frozen_amethyst, satin_gold, cc_nothern_aurora.*

Shadow and visibility Our results do not integrate the visibility other than by a rough ambient occlusion approximation. Two problems must be faced to integrate the visibility in our framework: (1) projecting the dynamic visibility on SH at each frame and (2) modifying the light-field projection at each shaded point. While the former is related to efficient SH projection of arbitrary functions and is an active research area, the latter is related to handling anisotropic emission as noted above. It should also be noted that integrating visibility would add a costly general SH triple product (Section 2.2.7). Solving this visibility problem is an important future research direction while being orthogonal to our proposal that focuses on efficient gradient computation.

SH band limit for real-time rendering The number of bands can be a bottleneck for both the SH grid computation, and the SH interpolation at shading stage. In the work for polygonal light [WCZR20], scalability has not been investigated when increasing the number of SH bands. In this thesis, we have shown that using a higher number of bands to construct the 3D grid is faster with spherical lights than with polygonal lights (Table 3.2). However, during interpolation, the number of memory accesses to retrieve the SH coefficients and gradients is high (especially when using a high number of SH bands) and creates a non-negligible bottleneck (Table 3.5). We believe that a careful low-level implementation on GPU could reduce this limitation.

SH bands	Eval coeff & grad.	Interp.	Total	Reference	
				Time	Speed up
5	1.39	7.65	9.04	45.25	5.0×
6	1.87	8.90	10.77	53.77	5.0×
7	2.43	26.14	28.57	77.18	2.7×
8	3.07	55.57	58.64	98.79	1.7×
9	3.35	132.29	135.64	148.04	1.1×

Table 3.5: *Timings (in ms) with different number of SH bands using Figure 3.11. Interpolation is difficult to compute in reasonable times when increasing the SH bands due to the large number of memory accesses required to retrieve the SH coefficients. Polygonal lights [WCZR20] suffer from the same problem since interpolation has also to be performed.*

Applications of SH gradients The goal of our last proposal is to efficiently reconstruct the radiance field using a gradient based interpolation method (Section 3.3). This is also the goal of Neural Radiance Field approaches (NeRF) (Section 2.3.8). Some recent methods use the SH and their gradients for this purpose [YLT+21, STC+22], hence they constitute a direct application of our method. Jimenez *et al.* [JWPJ16] propose to compute the ambient occlusion on SH using spherical kernels, *i.e.* the same framework as for spherical lights. Occlusion being generally smooth, this approach could take advantage of our gradient computation method. However, the use of gradients in a general visibility context is a complex problem because the visibility is not generally a continuous and smooth function.

Second-order derivatives Instead of only using the gradient, existing methods go further and compute the Hessian matrix [SJJ12, MJJG18]. This matrix might be computed by deriving again the recursive relations we developed. As our method exhibits auto-similarities in its formulation, *e.g.* deriving SH basis function results in an expression using the same subterms but at another degree Equation 3.30, 3.16 and 3.18, this might ease this derivation. However, the mathematical and computational complexity of these derivations need to be carefully investigated and we leave this interesting direction for future investigations.

3.5 Conclusion

This chapter proposes three distinct contributions. The first one is the recursive computation of the ZH coefficients representing the radiance field produced by spherical lights in the reference frame oriented according to the light sphere position. The second is the extension of the Sloan's method of efficient computation of SH basis functions [Slo13] to compute their gradient. Finally, the third one is derived from the two previous ones in order to propose an efficient method to compute the gradient of the SH coefficients representing the radiance field produced by spherical lights.

However, the computation of the gradient of the SH coefficients requires a careful implementation on GPU. Indeed, a naive implementation of the method produces a strong pressure on the registers, strongly impacting the performances. An additional development effort is necessary to lower this pressure and thus drastically improve the performances, in particular with the increase of the SH band limit. However, even without this improvement, the methods developed in this chapter for spherical lights are more efficient than the state-of-the-art method for polygonal lights [WR18, WCZR20]. In order to take advantage of these better performances with spherical lights, the next chapter proposes to approach the SH projection of the radiance field produced by polygonal lights with spheres.

4 Direct Spherical Harmonics Lighting

In chapter 3, it was demonstrated an effective SH projection of the radiance field emitted by spherical lights. To take full advantage of the performance gains of these methods, the outgoing light at any point in the scene must be computed efficiently. In this chapter, an efficient framework for computing lighting with *decomposable BRDFs* is proposed (Section 4.1). This framework is particularly useful when using dynamic scenes, without any scene preconditions, where traditional SH lighting methods require a sufficiently refined mesh depending on the camera viewpoint.

In computer graphics, the use of polygonal lights is very common, for instance several methods in the literature estimate the global illumination by considering the triangles of the scene mesh as light sources [XCM+14, LDdLRB16]. It has been shown in chapter 3 that the methods of SH projection of the radiance field were more efficient with the methods proposed in this thesis for spherical lights compared to polygonal lights. In order to take advantage of this acceleration, several methods are proposed to approximate the SH projection of the radiance field produced by polygonal lights. Thus, future methods can take advantage of our acceleration with a good quality for the radiance field approximation.

4.1 Efficient shading with decomposable BRDF

Precomputed Radiance Transfer (PRT) (Section 2.3.3) precomputes and stores the lighting using SH at the vertices of a mesh and interpolate the SH over the triangles to compute per-pixel shading. 3D scenes are nowadays made of very refined meshes, and thus require long precomputation time and large memory space to store and interpolate the SH coefficients. The proposal of this section makes it possible to dispense with this costly precomputation step.

In addition, SH have limited ability to represent high frequencies with limited memory requirements. As such, they first have been used to store and compute low frequency lighting contribution when using *decomposable BRDFs*. Recent approaches have been proposed to use SH for the specular contribution, but at the cost of increasing the rendering time. The main objective of this work is to reformulate SH operations to speed up the shading computation.

Using *decomposable BRDF*, we observed that diffuse and specular contributions are always treated separately in the literature (Section 2.3.2), but in practice there is a clear com-

Figure 4.1: *Computing per-pixel shading using Spherical Harmonics (SH) is a costly task that requires to compute a SH product when evaluating specular contribution. In this work, we derive a new formulation of this product for* decomposable BRDFs. *Our approach reduces computation timings by two orders of magnitude and improves the quality of the final image (as measured by RMSE). Top: images obtained with our approach using 9 SH bands. Bottom: Reference images generated by sampling 10,000 rays for each light.*

putational similarity between them. In this work we demonstrate the gain obtained by taking advantage of this similarity, and expose an efficient algorithm to compute the SH product between any function and the clamped cosine one, allowing a significant acceleration (Figure 4.1).

At first, we quickly recall the SH properties we need for these contributions (Section 4.1.1). Then, we present how to compute the diffuse contribution from the specular one (Section 4.1.2) and the way we can compute efficiently the SH product between an arbitrary function and a clamped cosine (Section 4.1.3). We conclude by an exploration of the different use case for both methods (Section 4.1.4) and an evaluation (Section 4.1.5).

4.1.1 Recall and reformulation of SH properties

Before introducing the contributions of this section, it is necessary to recall some properties of spherical harmonics as well as the traditional method of calculation of the diffuse and specular contributions with spherical harmonics.

Double and SH product Double product and SH product are common operations when using SH as described in Section 2.2.6 and 2.2.7. Double product computes a spherical convolution between two functions, and is defined as

$$\begin{aligned} D &= \int_\Omega L(\omega)F(\omega)\,\mathrm{d}\omega \\ &= \int_\Omega \Big(\sum_{l1,m1} \mathbf{L}_{l1}^{m1} Y_{l1}^{m1}(\omega)\Big)\Big(\sum_{l2,m2} \mathbf{F}_{l2}^{m2} Y_{l2}^{m2}(\omega)\Big)\,\mathrm{d}\omega \\ &= \sum_{l1,m1}\sum_{l2,m2} \mathbf{L}_{l1}^{m1}\mathbf{F}_{l2}^{m2} \int_\Omega Y_{l1}^{m1}(\omega)Y_{l2}^{m2}(\omega)\,\mathrm{d}\omega \\ &= \sum_{l,m} \mathbf{L}_l^m \mathbf{F}_l^m \ . \end{aligned} \qquad \text{(2.46 revisited)}$$

The SH product computes the SH coefficients of the multiplication of two spherical functions:

$$\begin{aligned}\mathbf{T}_l^m &= \int_\Omega \Big(\sum_{l1,m1} \mathbf{L}_{l1}^{m1} Y_{l1}^{m1}(\omega) \Big) \Big(\sum_{l2,m2} \mathbf{F}_{l2}^{m2} Y_{l2}^{m2}(\omega) \Big) Y_l^m(\omega)\, \mathrm{d}\omega \\ &= \sum_{l1,m1} \sum_{l2,m2} \mathbf{L}_{l1}^{m1} \mathbf{F}_{l2}^{m2} \int_\Omega Y_{l1}^{m1}(\omega) Y_{l2}^{m2}(\omega) Y_l^m(\omega)\, \mathrm{d}\omega \end{aligned} \quad . \qquad \text{(2.48 revisited)}$$

Estimating diffuse and specular contributions Let's consider a light function L, a visibility V (cosine weighted according to surface normal) and a decomposable BRDF M. The diffuse contribution is computed as the double product between $\mathbf{L}$ and $\mathbf{V}$ and multiplied by the albedo. The specular contribution is given by computing the SH product between $\mathbf{L}$ and $\mathbf{V}$ and then apply the double product between this result and the BRDF specular lobe. This process requires to compute double and SH product on the same two functions, which makes numerous computations redundant.

4.1.2 Diffuse from specular

In this section, we show how to compute the diffuse contribution of the lighting knowing the specular one by replacing the double product (D) by a single multiplication. Let's call O, the unit function over the sphere:

$$O(\omega) = \mathbb{1} = \sum_{l,m} \mathbf{O}_l^m Y_l^m(\omega) \; . \tag{4.1}$$

Due to SH properties, only coefficient $\mathbf{O}_0^0$ is non-null and equal to

$$\mathbf{O}_0^0 = \int_\Omega Y_0^0 \, \mathrm{d}\omega = \sqrt{\frac{1}{4\pi}} 4\pi = 2\sqrt{\pi} \approx 3.5449077 \; . \tag{4.2}$$

The integral of the product between two functions is equal to the double product D between O and the result of the SH product between the same two functions. Thus, since the vector $\mathbf{O}$ has only one non-null coefficient ($\mathbf{O}_0^0$), only the coefficient $\mathbf{T}_0^0$ from the SH product is needed. As the SH Y_0^0 is a constant, it can be taken out of the integral corresponding to the coefficient $\mathbf{T}_0^0$

$$\mathbf{T}_0^0 = Y_0^0 \sum_{l1,m1} \sum_{l2,m2} \mathbf{L}_{l1}^{m1} \mathbf{F}_{l2}^{m2} \int_\Omega Y_{l1}^{m1}(\omega) Y_{l2}^{m2}(\omega)\, \mathrm{d}\omega = Y_0^0 D \; . \tag{4.3}$$

By reversing, we obtain

$$\mathbf{T}_0^0 2\sqrt{\pi} = \mathbf{T}_0^0 \mathbf{O}_0^0 = D \; . \tag{4.4}$$

This signifies that when computing the SH product between two functions, the computation of the double product between the same functions is done with a single multiplication ($D \approx 3.5449077 \mathbf{T}_0^0$) which corresponds to the double product between $\mathbf{T}$ and $\mathbf{O}$.

4.1.3 Efficient clamped cosine product

Our method drastically reduces the number of arithmetic operations needed to compute the SH product of a function $\mathbf{L}$ and a clamped cosine. The clamped cosine is a circularly symmetric signal, hence it can be represented only with ZH (Section 2.2.9).

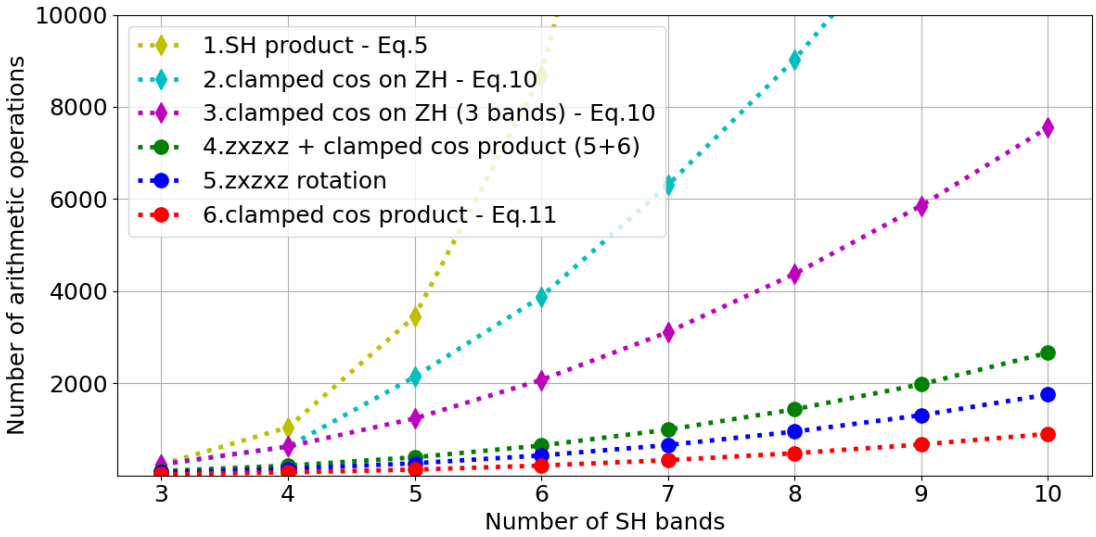


Figure 4.2: *Number of arithmetic operations needed to compute different SH product up to a certain band.*

Then, considering $\mathbf{F}$ as a clamped cosine, we replace it in the SH product (Equation 2.48) by the equation of rotation of ZH functions (Equation 2.73), we obtain:

$$\mathbf{T}_l^m = \sum_{l_1,m_1} \sum_{l_2,m_2} \mathbf{L}_{l_1}^{m_1} Y_{l_2}^{m_2}(n) \left(\sqrt{\frac{4\pi}{2l_2+1}} \mathbf{F}_{l_2}^0 C_{l_1\ l_2\ l}^{m_1\ m_2\ m} \right) \tag{4.5}$$

where n is the normal surface and $C_{l_1\ l_2\ l}^{m_1\ m_2\ m}$ are the Clebsch-Gordan coefficients corresponding to the integral in Equation 2.48.

This solution is more efficient than the classical SH product (Figure 4.2). Moreover, as most of the cosine energy is caught on 3 SH bands, we can limit the cosine projection to this number of bands ($l_2 < 3$) hence improving its evaluation. Nevertheless, there is a more efficient solution that does not imply a low frequency approximation of the cosine function. We can instead express the cosine in the world coordinate system and precompute it directly in the integral:

$$\mathbf{T}_l^m = \sum_{l_1,m_1} \mathbf{L}_{l_1}^{m_1} \left(\int_\Omega Y_{l_1}^{m_1}(\omega) \max(0, \omega \cdot z) Y_l^m(\omega) d\omega \right). \tag{4.6}$$

When the light function is expressed in the same coordinate system, this solution provides fewer arithmetic operations. If light is not in the same coordinate system, adding a rotation step, as the *zxzxz* method (Section 2.2.12), adds few operations and remains more efficient than other methods (Figure 4.2). The parts of the equations into brackets (Equation 4.5 and Equation 4.6) are precomputed to generate the code that will apply the SH product.

4.1.4 Use cases

As pointed out in the introduction, the goal of our methods is to compute the shading with SH using as few arithmetic operations and data storage as possible, and thus to reduce the computation time for rendering.

We compiled in Figure 4.3 the different algorithms one can implement to compute shading using SH, considering four functions that appear in the rendering equation: light (L),

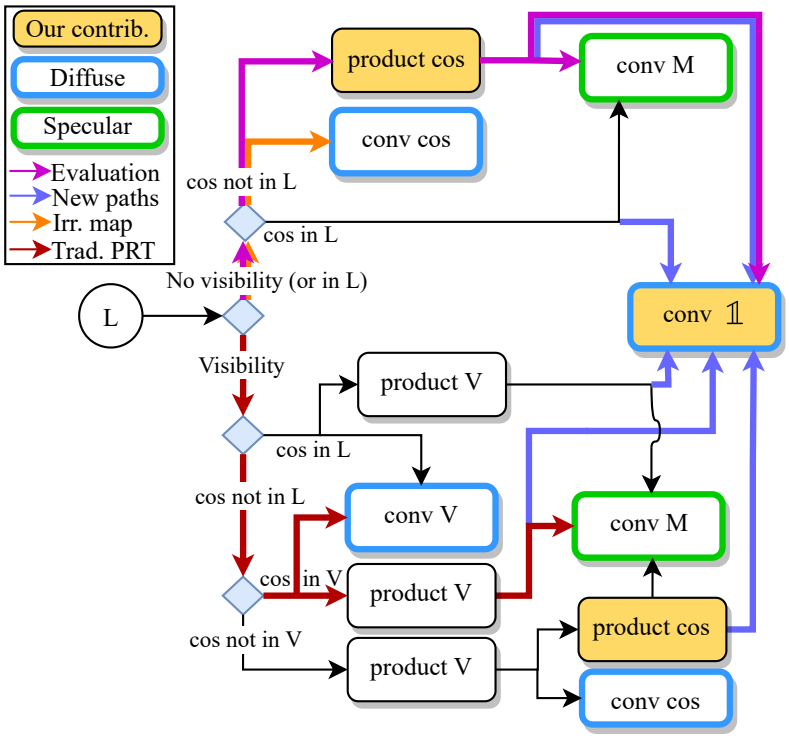


Figure 4.3: *Flowchart of typical SH rendering with decomposable BRDF. We annotated four paths: 1. paths we used for our evaluation, 2. new paths from our first contribution, 3. Irradiance map from [RH01], 4. Traditional PRT.*

visibility (V), brdf (M) and the clamped cosine ($\cos$). Paths that do not use visibility are particularly adapted to render fully dynamic scenes because shading do not depend on mesh-specific precomputations, unlike when visibility is precomputed at each vertex. Using traditional PRT, clamped cosine is packed with visibility in a precomputation step for each vertex. Similar but less common, cosine can be packed with light instead of visibility. This path can be used with dynamic visibility that do not take cosine into account, as it is included in the light function. Finally, path at the bottom, where L, V and cosine are projected separately on SH is the more general. This path can be taken if everything is dynamic, light and visibility, for instance when using many polygonal lights [WCZR20] and with dynamic visibility [RWS+06]. In such case, cosine cannot be packed with either function. Our contributions should be used for all of these algorithms to reduce their computational cost.

4.1.5 Evaluations and discussions of the proposals

For all our experiments, we use the magenta path with spherical lights sources where lighting is computed at each pixel. We evaluate our results on a RTX 2080 with all renderings done at resolution 1920×968. To compare, we compute the traditional SH product (Equation 2.47) using the same path for specular and Irr. map path for diffuse. To render dynamic scenes, cosine is precomputed on ZH and rotated on the fly using the efficient ZH rotation (Section 2.2.12) before computing the product.

Estimating diffuse from specular

Our method to compute double product from SH product allows to avoid the convolution needed to compute diffuse contribution, thus replacing N^2 multiplications, where N is the

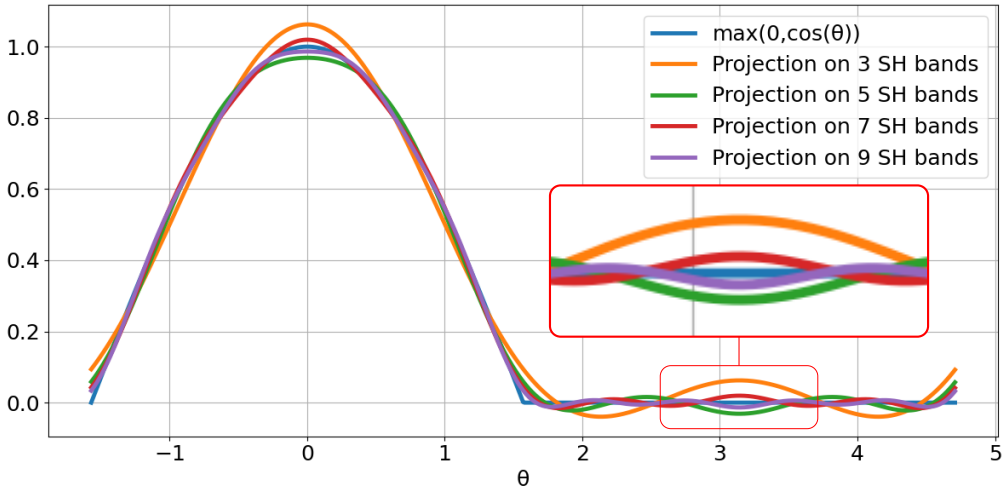


Figure 4.4: *Clamped cosine projected on different number of SH bands. Even though the majority of the cosine energy is caught with few bands, it produces non-negligible ringing artifacts as it is clamped.*

number of SH bands, by a single multiplication. This method creates a new derivation on each path (Figure 4.3). Moreover, diffuse is classically computed on 3 SH bands as the majority of the cosine energy is caught (Section 2.3.1) . This restriction produces non-negligible ringing artifacts, visible when light is only on one object's side (Figure 4.5), and is clearly explained by looking at the behavior of clamped cosine projection on SH (Figure 4.4). Our method removes these artifacts by computing diffuse lighting on the same number of bands as specular at no extra cost (Figure 4.5).

Efficient SH product with clamped cosine

The efficient SH product with cosine is well adapted to render dynamic scenes. It fits into paths where the SH product with cosine is a specific step. We have shown that with our method the number of arithmetic operations is greatly reduced (Figure 4.2), as well as computation time (Table 4.2). All our results were computed using Equation 4.6 with all SH coefficients expressed in the same frame. We also demonstrated (Table 4.2, last row) that when coefficients are not expressed in the same frame, thus requiring an additional SH

Eq. for SH product	Figure 4.2*	rmse ($\times 10^2$)		
		Figure 4.1a	Figure 4.1b	Figure 4.1c
Equation 2.47	1	2.4488	0.2609	5.1379
Equation 4.5	2	2.3652	0.2609	4.9844
Equation 4.5 ($l_j < 3$)	3	2.8814	0.5933	5.2027
Equation 4.6 (w. or wth. rotation)	6/4	2.1968	0.2156	4.9212

* Corresponding line in Figure 4.2

Table 4.1: *RMSE ($\times 10^2$) for the SH product. Our method always generates the lowest rmse because the cosine is not directly approximated by SH Equation 4.6. The important difference in magnitude between figures is due to the nature of materials used, mainly diffuse (Figure 4.1b) or glossy (Figure 4.1c). Specular is computed up to 9 SH bands while diffuse is computed with our method Equation 4.4.*

Eq. for SH product	Figure 4.2*	band limit			
		3	5	7	9
Equation 2.47	1	1.65	6.92	183.54	836.34
Equation 4.5	2	1.65	6.42	128.22	590.13
Equation 4.5 ($l_j < 3$)	3	1.63	3.78	41.54	289.88
Equation 4.6	6	1.46	1.78	2.92	6.09
Equation 4.6 (w. rotation)	4	1.59	2.19	5.08	15.78

* Corresponding line in Figure 4.2

Table 4.2: *Timings (ms) with the different solution to apply the SH product with a clamped cosine. Specular is computed up to the corresponding band while diffuse is computed using our proposal Equation 4.4.*

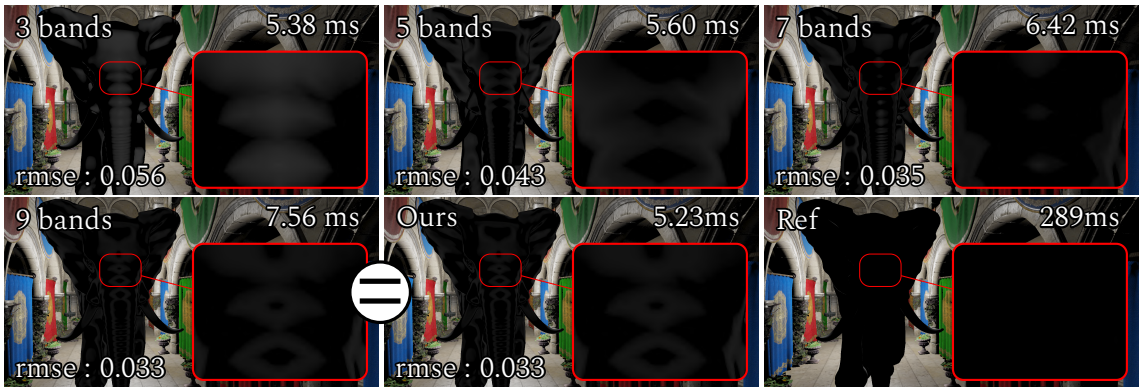


Figure 4.5: *Ringing due to clamped cosine (Figure 4.4). The scene contains only one light behind the elephant. Specular contribution is always computed on 9 SH bands. Diffuse contribution is computed with various number of SH bands, or evaluated from specular computation (our method). Negative color of shading result are clamped to zero, it explains why inset for 7 bands is darker than 9 while rmse is still higher. Ringing problem occurs too when using visibility with SH because of the frequency cut introduced by the clamping of the southern hemisphere of the unit sphere.*

rotation, our proposal still greatly improves the computation time. This last case is useful, e.g., when the frame of the SH coefficients (and their gradient) is fixed to compute them on a 3D grid (Section 3.3). Moreover, since the cosine is not directly approximated with SH (Equation 4.6), the rmse is lower than the other methods (Table 4.1).

4.2 Polygonal light approximation

We propose a method to approximate the radiance field produced by polygonal lights, more precisely, we seek a good approximation of the SH projection of the radiance field produced by polygonal lights. As any polygon can be divided into triangles, we focus our approximation on triangular lights. The method we propose approximates each triangular light by a corresponding spherical light. Approximating a triangular light by several spherical lights, or inversely approximate multiple triangular lights by one spherical light, are out of scope of this contribution and are left as future work.

4.2.1 Overview of the proposals

Let's quickly recap how we compute the SH projection at a point $\mathbf{x}$ of the radiance produced by a spherical light at position $\mathbf{y}$.

Reminder of the SH projection of the radiance field produced by spherical light.

In a first time, the ZH coefficients are computed in the coordinates frame aligned with the position of the spherical light

$$\tilde{\mathbf{L}}_l = \begin{cases} \sqrt{\frac{\pi}{2l+1}}(P_{l-1}(\cos(a(\mathbf{x}))) - P_{l+1}(\cos(a(\mathbf{x})))) & \text{if } l \neq 0 \\ \sqrt{\pi}(1 - \cos(a(\mathbf{x}))) & \text{otherwise} \end{cases} \quad \text{(3.10 revisited)}$$

where $a(\mathbf{x})$ is the half-angle subtended by the spherical light and defined by

$$a(\mathbf{x}) = \arcsin\left(\frac{r}{\|\mathbf{y} - \mathbf{x}\|}\right). \quad \text{(3.27 revisited)}$$

The SH coefficients are obtained by applying the ZH rotation (Section 2.2.12)

$$\mathbf{L}_l^m = \sqrt{\frac{4\pi}{2l+1}} Y_l^m(\omega_l)\tilde{\mathbf{L}}_l \,. \quad \text{(3.11 revisited)}$$

Our method to approximate the SH projection of the radiance field consists in positioning a spherical light on a triangle in order to obtain a similar geometrical configuration (Section 4.2.2). Then, the energy is corrected according to a term depending on the normal of the triangle and the direction to the point where the radiance field is evaluated (Section 4.2.3). However, the energy correction can be taken into account at different steps of the computation. We published a first method that apply this correction on the intensity of the radiance field by modifying the amplitude of the spherical harmonics coefficients (Equation 3.11 - Main method - [MMBP22]). Nevertheless, after developing this method (Section 4.2.3), we discuss an alternative (Section 4.2.5 - Alternative method) which consist in correcting the energy by modifying the solid angle of the sphere to match that of the triangle. Both methods keep a similar energy in the radiance field, the main method keeps the same solid angle and corrects the energy by changing the intensity while the alternative method corrects the energy by changing the solid angle without modifying the intensity (Figure 4.6).

4.2.2 Spherical light parameters

We look for the parameters of the spherical lights, in order to have a similar geometrical configuration with the triangular lights. The spherical light position and radius are deduced from the triangle vertices (Figure 4.7).

The position p_s of the sphere is defined as the centroid of the triangle:

$$p_s = \frac{p_1 + p_2 + p_3}{3}, \quad (4.7)$$

where p_1, p_2 and p_3 are the triangle vertices. The sphere radius is defined so that a disc with the sphere's radius and the triangle have the same area. Reminding that πr^2 measures the area of a disc, and $\|\vec{p_1p_2} \wedge \vec{p_1p_3}\|/2$ the triangle area, the radius of the sphere is:

$$r_s = \sqrt{\frac{1}{2\pi}\|\vec{p_1p_2} \wedge \vec{p_1p_3}\|}. \quad (4.8)$$

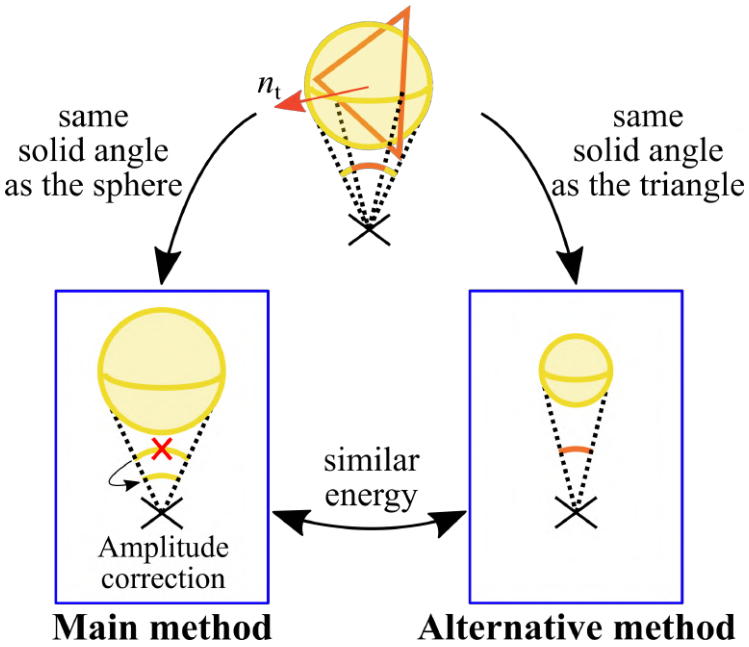


Figure 4.6: *Overview of the two approximation methods, where the main method fixes the parameters of the sphere (position and radius) once for all, while the alternative method changes the radius so that the solid angle of the sphere at each shaded point is equal to the solid angle of the triangle.*

4.2.3 Spherical harmonics coefficients and gradients

SH coefficients For each shading point, the flatness of the triangle is recovered from the spherical shape of the light by multiplying its energy by the factor $|\cos\gamma|$ (Figure 4.7), where γ is the angle between the triangle normal n_t and the direction $-\omega$ from the center of the spherical light to the shaded point. As this factor varies for each light at each shaded point, the equation computing the SH coefficients for the light at the shaded point (Equation 3.11) needs to be adapted, such that:

$$\mathbf{P}_l^m(\mathbf{x}) = \sqrt{\frac{4\pi}{2l+1}} Y_l^m(\omega)\tilde{\mathbf{L}}_l(\mathbf{x})\,|\cos\gamma| \tag{4.9}$$

(we use the absolute value for $|\cos\gamma|$ to consider the triangular light as double sided). This equation only multiplies by $|\cos\gamma|$ the SH projection of spherical light that we proposed in Section 3.1.2.

Impact on the gradient Thus, the gradient of this equation is deduced from the gradient of the original equation (Equation 3.26) as:

$$\begin{aligned}\nabla_{\mathbf{x}}\mathbf{P}_l^m(\mathbf{x}) = -\sqrt{\frac{4\pi}{2l+1}}\big(&\nabla_{\mathbf{y}} Y_l^m(\omega)\tilde{\mathbf{L}}_l(\mathbf{x})\,|\cos\gamma| + \\ &Y_l^m(\omega)\nabla_{\mathbf{y}}\tilde{\mathbf{L}}_l(\mathbf{x})\,|\cos\gamma| + Y_l^m(\omega)\tilde{\mathbf{L}}_l(\mathbf{x})\nabla_{\mathbf{y}}\,|\cos\gamma|\big)\,,\end{aligned} \tag{4.10}$$

with

$$\nabla_{\mathbf{y}}\,|\cos\gamma| = \Big(\frac{-n_t}{\|\mathbf{y}-\mathbf{x}\|} + (\mathbf{y}-\mathbf{x})\frac{n_t\cdot(\mathbf{y}-\mathbf{x})}{\|\mathbf{y}-\mathbf{x}\|^3}\Big)\frac{\cos\gamma}{|\cos\gamma|}\,. \tag{4.11}$$

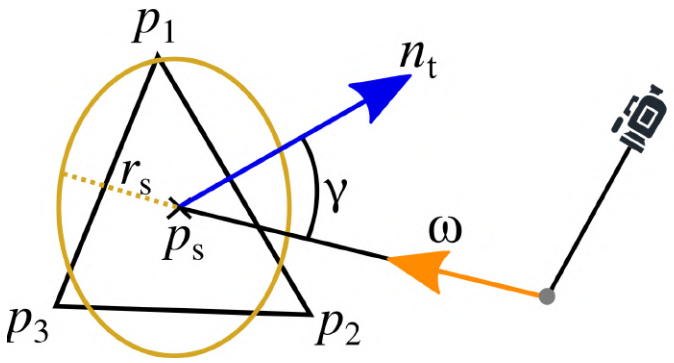


Figure 4.7: *Approximation of a triangular light with a sphere. The sphere position p_s is the centroid of the triangle (Equation 4.7) and the radius r_s is determined so that the area of the yellow disc is equal to the area of the triangle (Equation 4.8). γ is the angle between the triangle normal n_t and the direction ω. The shaded point is in grey.*

SH bands limit	Spherical lights (Ours)	Polygonal lights	
		Our approx.	Ref. [WCZR20]
5	2.22	2.53	7.21
6	2.96	3.39	13.08
7	3.77	4.42	16.21
8	4.76	5.35	29.07
9	5.18	5.74	164.01

Table 4.3: *Timings (in ms) for the computation of the SH and gradients coefficients in a 8^3 3D grid, using 648 lights (spheres or triangles). The computation for spherical lights and our approximation of polygonal lights are computed with the inversion of the loops (Section 3.3.3). Our approximation of polygonal lights exhibits a low overhead compared to spherical lights.*

4.2.4 Evaluation of the main proposal

In order to evaluate our approximation of polygonal lights (Figure 4.8), we propose to consider a square light source split in small triangles (as done in [WCZR20]), and then replace each triangle by our spherical approximation (Section 4.2.2). For our comparison, we use a slightly modified version of the code published by Wu *et al.* [WCZR20]. They used a formula from Arvo [Arv95] to compute the solid angle subtended by a triangle:

$$\Omega_t = \left(\sum_{i=1}^{3} \cos^{-1}((\vec{u_{i-1}} \times \vec{u_i}) \cdot (\vec{u_i} \times \vec{u_{i+1}})) \right) - \pi \,, \tag{4.12}$$

in which $\vec{u_i} = (p_i - \mathbf{x})/(\|p_i - \mathbf{x}\|)$ is the normalized direction from the point $\mathbf{x}$ to the vertex p_i of the triangle (Figure 4.13), and the index i is circular $\vec{u_4} = \vec{u_1}$ and $\vec{u_0} = \vec{u_3}$. This formula creates numerical inaccuracies when a triangular light is far away or when many polygonal lights have aligned edges. This strongly affects the comparison between triangular lights and our approximation. To avoid these problems, we compute the solid angle subtended by a triangle using the Van Oosterom's formula [VS83]:

$$\tan \frac{\Omega_t}{2} = \frac{(\vec{u_1} \times \vec{u_2}) \cdot \vec{u_3}}{1 + \vec{u_1} \cdot \vec{u_2} + \vec{u_2} \cdot \vec{u_3} + \vec{u_3} \cdot \vec{u_1}} \,. \tag{4.13}$$

We analyze three main criteria of scalability: the number of light sources, the 3D grid resolution, and the SH band limit. Our approximation of triangular lights creates only a small

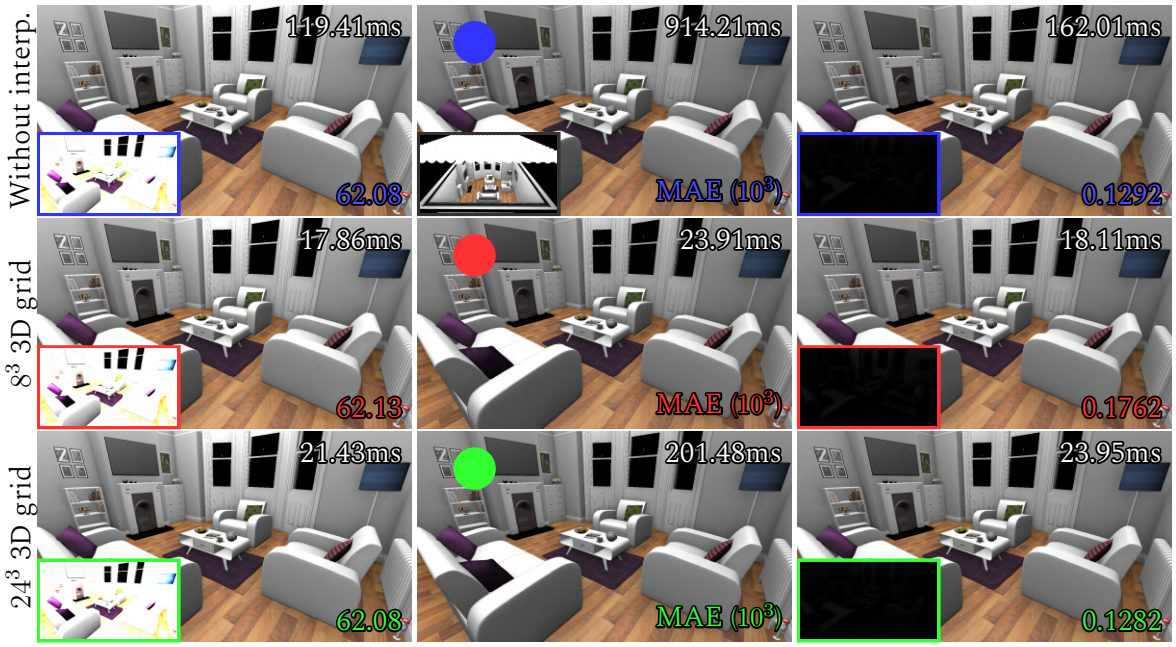


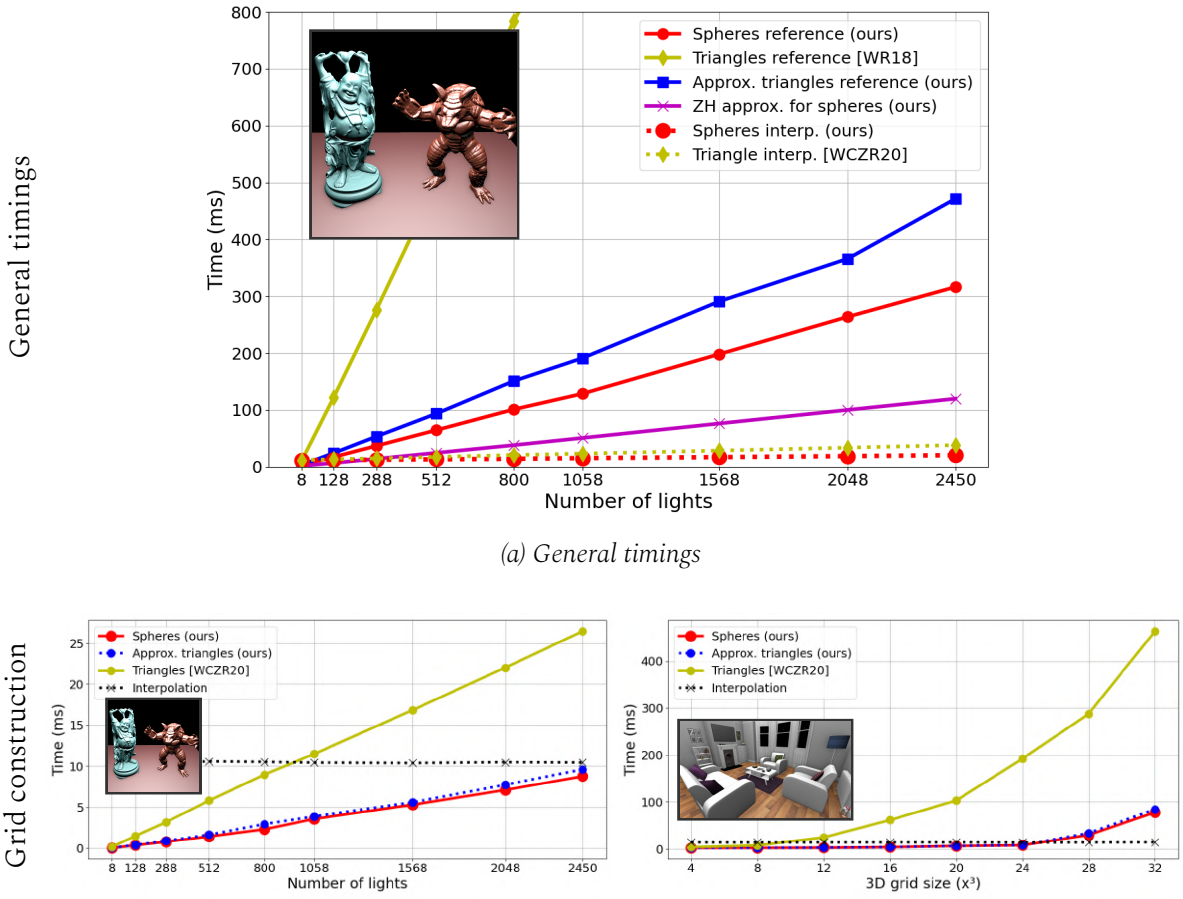
Figure 4.8: *Accuracy of our approximation of triangular lights [WR18, WCZR20] using interpolation. The insets show* 100× *absolute errors, insets and MAE are computed using the image with the corresponding color patch as reference. MAE is reduced by two orders compared to raw spherical lights and the* 100× *absolute error is only slightly visible, which illustrate the accuracy of our approximation of polygonal lights. This figure also highlights that triangular lights timings increase significantly with the size of the 3D grid (Figure 4.9c).*

overhead over spherical lights and it is more efficient to compute than the original triangular light (Figure 4.9b). We also observe a significantly better scalability of our approximation when increasing the number of SH bands (Table 4.3) or increasing the 3D grid resolution (Figure 4.9c). Our representation also provides an accurate approximation of the SH coefficients of triangular lights (Figure 4.10). The MAE is reduced by one order of magnitude compared to raw spherical lights (Table 3.4). The approximation error of the SH coefficients has a quadratic behavior according to the number of SH bands (Figure 4.10d). However, using the number of SH bands traditionally considered in real-time rendering, *i.e.* 5 to 9 SH bands, the error on the coefficients approximation remains very low, in an order of 10^{-8}, and the error in the final rendering is only slightly increased (Figure 4.11).

Limitations of the main proposal The triangles configuration plays a very important role in the error introduced by our approximation (Figure 4.12). This error lies in the difference of the shapes, sphere against triangle, projected on the unit hemisphere over a shading point. The higher this difference, the higher the approximation error. This difference is related to the solid angle subtended by the triangle, for well-shaped triangles, or to the narrowness of the triangle in the general case. The narrowness can be quantified by the inverse of the intersection area between the triangle and a co-planar disk centered at the center of gravity of the triangle having the same area as the triangle.

4.2.5 Alternative method

In this chapter, we approximate the lighting produced by polygonal lights by positioning a spherical light, whose position and radius are fixed once for all according to the triangle



(a) General timings

(b) Timings according to number of lights

(c) Timings according to grid size

Figure 4.9: *Timings (in ms) comparison between spherical light and triangular light (Section 3.3) extended with our approximation of triangular lights. (a) General timings to compute the final shading, on a 8^3 3D grid using different number of lights on Figure 3.12. (b) Timings to compute SH coefficients and their gradients in a 8^3 3D grid using different number of lights on Figure 3.12. (c) Timings to compute SH coefficients and their gradients for different 3D grid resolutions on Figure 4.8 with 722 lights.*

geometry. The energy is corrected by modifying the amplitude of the radiance function. This solution is very efficient and accurate at low-frequency (under 9 SH bands for the evaluation we proposed) as shown in the previous subsections. The alternative method consists in fixing also once for all the position of the sphere and its radius is determined at each shaded point so that the solid angle of the sphere is equal to the solid angle of the triangle approximated.

Specifically, there is no need to compute precisely the value of the new radius but only the new half-angle $\hat{a}(\mathbf{x})$ subtended by the sphere since the solid angle Ω_s of a sphere is only defined by the half-angle

$$\Omega_s = 2\pi(1 - \cos(\hat{a}(\mathbf{x}))) \quad \rightarrow \quad \hat{a}(\mathbf{x}) = \arccos(1 - \frac{\Omega_s}{2\pi}) \,. \tag{4.14}$$

The half-angle $\hat{a}(\mathbf{x})$ is computed at each shaded points where Ω_t is the solid angle of the triangle (Equation 4.12 or 4.13). Therefore, this alternative method only modify the gradient

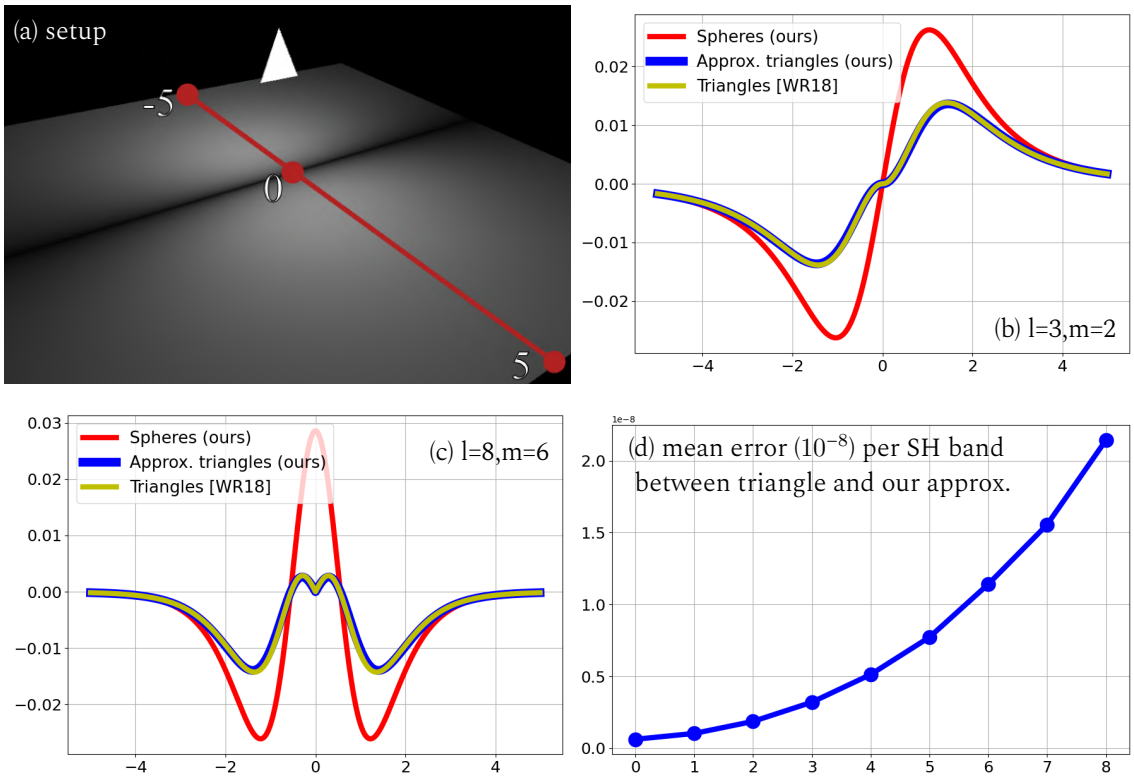


Figure 4.10: *Accuracy of our approximation of triangular lights SH projection [WR18]. We show the setup of the scene in (a). In (b) and (c) we evaluate one SH coefficient on the red line highlighted in (a). The plot in (d) represents the mean error (10^{-8}) between triangles and our approximation computed for each SH band on the whole plane lighted by the light.*

of the subtended half-angle in the computation of the gradient of the spherical harmonics coefficients for spherical lights. The gradient of the new half-angle is defined by

$$\nabla_{\mathbf{y}}\hat{a}(\mathbf{x}) = \frac{-\nabla_{\mathbf{x}}\Omega_t}{\sqrt{\Omega_t(4\pi - \Omega_t)}} \tag{4.15}$$

where $\nabla_{\mathbf{x}}\Omega_t$ is the gradient of the solid angle of the triangle. This gradient can be computed by a simple derivation of the work from Wu *et al.* [WCZR20] on the gradient of the SH coefficients for polygonal lights. They leverage the Reynolds transport theorem which originates in fluid mechanics [Lea07], *i.e.* the gradient of the SH coefficients is solved by an integral on the contours of the projection of the polygonal light on the unit sphere at the shaded point. As they consider the radiance as constant over the solid angle, the projection integral only involves the SH basis function. Thus, we only need to remove all the terms depending on the SH and simplifying to obtain a method for computing the gradient of the solid angle subtended by a polygonal light

$$\nabla_{\mathbf{x}}\Omega_t = \sum_{i=0}^{N} \left(\frac{\vec{u_i} \times \vec{u_{i+1}}}{\|\vec{u_i} \times \vec{u_{i+1}}\|} \right) \left(\frac{\cos(T_i) - 1}{\sin(T_i)} \right) \left(\frac{1}{\ell_i} + \frac{1}{\ell_{i+1}} \right) \tag{4.16}$$

where N is the number of vertices of the polygonal lights, $T_i = \arccos(\vec{u_i} \cdot \vec{u_{i+1}})$ and $\ell_i = \|p_i - \mathbf{x}\|$ (Figure 4.13). This equation is directly applicable for triangular lights. However, a careful differentiation of the equations to compute the solid angle of the triangle (Equation 4.12 and 4.13) could maybe lead to less computational equations but this is left as future work.

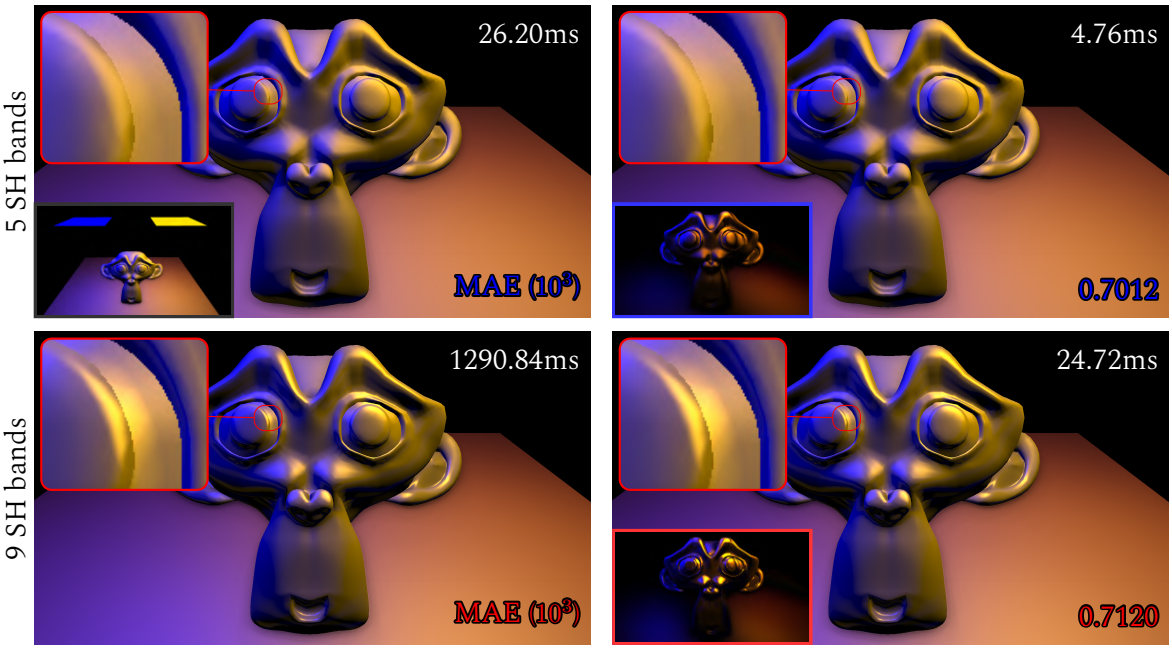


Figure 4.11: *The error of our approximation of triangular light [WR18] according to the number of SH bands. Results are generated with 36 lights, without interpolation and a glossy model. Bottom images inset in the left show the scene configuration and insets in the right show* 100× *absolute errors. The approximation error remains in the same magnitude order between 5 and 9 SH bands.*

Methods comparison

In terms of quality, the alternative method allow to reduce the approximation error of the radiance field produced by polygonal lights (Figure 4.14). The difference in error between the two methods lies in the approximation of the solid angle, this is particularly noticeable in the bright spot under the lights in the case where we use parallel triangular light to the shaded surface. In this case, the solid angle for the main method will be much larger than for the alternative method which will use the same solid angle value as for the triangle but not with the exact same shape.

Without interpolation, the main method for the approximation is the computation for each light of the factor $|\cos\gamma|$, that is a single scalar product, and then multiply it by all the SH coefficients. While the alternative method needs to compute for each light the solid angle of the triangle, so this means that the alternative method must keep in memory the 3 vertices of the triangle, which adds 2 floats (129% more data) compared to the main method which only keeps in memory the center, the normal of the triangle and the corresponding radius. The performance of the main method depends on the number of SH bands, since it multiplies the factor $|\cos\gamma|$ to each SH coefficients, which is not the case for the alternative method. As a result, on 5 SH bands (Figure 4.15), the main method is more efficient while on 9 SH bands, the alternative method becomes the most efficient.

With interpolation, the main method also becomes less efficient compared to the alternative method when the SH bands limit is increased for the same reason. However, the alternative method does not become more efficient on 9 SH bands. Indeed, for instance with 2048 lights, the main method is ≈ 1.27 times better than the alternative method on 5 SH bands and it is only ≈ 1.06 times better on 9 SH bands. One of the reasons for this

Compared method	Figure 3.12			Figure 4.8
	Number of lights			
	288	512	800	722
Comparison with **triangles** [WR18]				
Spheres (Equation 3.11)	15.86	15.76	15.67	62.08
Triangles approx. (Equation 4.9)	0.096	0.068	0.121	0.129
Comparison with triangles interp. [WCZR20]				
Spheres interp. (Equation 3.26)	15.93	15.83	15.75	62.13
Triangles approx. interp. (Equation 4.10)	0.215	0.165	0.174	0.176

Table 4.4: *MAE* (10^3) *with different scene configurations with an interpolation in a* 8^3 *3D grid.* ***Bold text*** *highlights rendering without interpolation. The comparisons show that our approximation of triangles significantly reduces the MAE compared to spherical lights.*

is that in order to lower the pressure of the registers, we have applied a loop inversion (Section 3.3.3). Thus, instead of computing the solid angle of the triangle and its gradient once per light for the alternative method, the inversion of the loops requires to recompute this information as many times as there are different degrees m. Thus, the computational weight of the operations to be computed for each light is greater because of this loop inversion, and so we would have to go up further in limit band for the alternative method to be more efficient than the main method.

However, there is a compromise between the two methods. Instead of computing exactly the solid angle of the triangle, it can be approximated by multiplying the radius of the sphere or the subtended angle

$$\hat{a_1}(\mathbf{x}) = \arcsin\left(\frac{r\,|\cos\gamma|}{\|\mathbf{y}-\mathbf{x}\|}\right) \qquad \hat{a_2}(\mathbf{x}) = a(\mathbf{x})\,|\cos\gamma| \ . \tag{4.17}$$

The approximation of the solid angle for $\hat{a_1}(\mathbf{x})$ and $\hat{a_2}(\mathbf{x})$ is very similar (Figure 4.16). However, this idea does not provide a satisfactory quality of the approximation which is due to a too large approximation of the solid angle (Figure 4.14). For you information, here is their respective gradients

$$\nabla_{\mathbf{y}}\hat{a_1}(\mathbf{x}) = r\frac{-\frac{(\mathbf{y}-\mathbf{x})}{\|\mathbf{y}-\mathbf{x}\|^2}\,|\cos\gamma| + \nabla_{\mathbf{y}}\,|\cos\gamma|}{\sqrt{\|\mathbf{y}-\mathbf{x}\|^2 - r^2\,|\cos\gamma|^2}} \tag{4.18}$$

$$\nabla_{\mathbf{y}}\hat{a_2}(\mathbf{x}) = \nabla_{\mathbf{y}}a(\mathbf{x})\,|\cos\gamma| + a(\mathbf{x})\nabla_{\mathbf{y}}\,|\cos\gamma| \ . \tag{4.19}$$

4.3 Conclusion

Motivated by an efficient SH projection of the radiance field produced by spherical light presented in chapter 3, the first part of this chapter proposes a framework for an efficient SH shading with *decomposable BRDFs* allowing to take advantage of this efficient SH projection with the use of dynamic scene without any pre-computation per scene and with no strong a priori on the scene mesh. This framework avoids redundant computations when computing both diffuse and specular contributions in SH shading. We have also shown that diffuse contribution can be easily computed with a single multiplication from the specular computation. We also shown that the SH product between light, or an arbitrary function,

and a clamped cosine can be done with only a few arithmetic operations. Our proposals then improve the computation time up to two orders of magnitude when using 9 SH bands. It could be envisioned to apply the same principle to other functions that we did for cosine, *i.e.* replace cosine by any other function in Equation 4.6. Nevertheless, while improving the computation time for the SH product, the final number of required operations will be hard to predict.

The second part of this chapter proposes an approximation of the SH projection of the radiance field produced by polygonal lights with spherical lights. This approach is motivated by two factors, the first being that the radiance field projection methods for spherical lights developed in this thesis are more efficient than the one developed in the state-of-the-art for polygonal lights. The second is that the use of polygonal lights is widespread in computer graphics and then several methods can benefit from our approach. However, the proposed method is limited to triangular lights, as any polygon can be divided by triangles, and distributes one spherical light per triangular light. The approximation of the SH coefficients have been clearly identified according to the geometric configuration of the lights. In these limiting cases, it appears that a distribution of several spherical lights per triangular light could push these limitations a little further. However, a careful investigation must be studied in order to guarantee optimal computation time and quality.

This chapter focused on direct lighting, the next chapter focuses on the use of SH for the estimation of global illumination and in particular by crossing the SH projection methods developed in chapter 3 with the state-of-the-art of virtual light based global illumination.

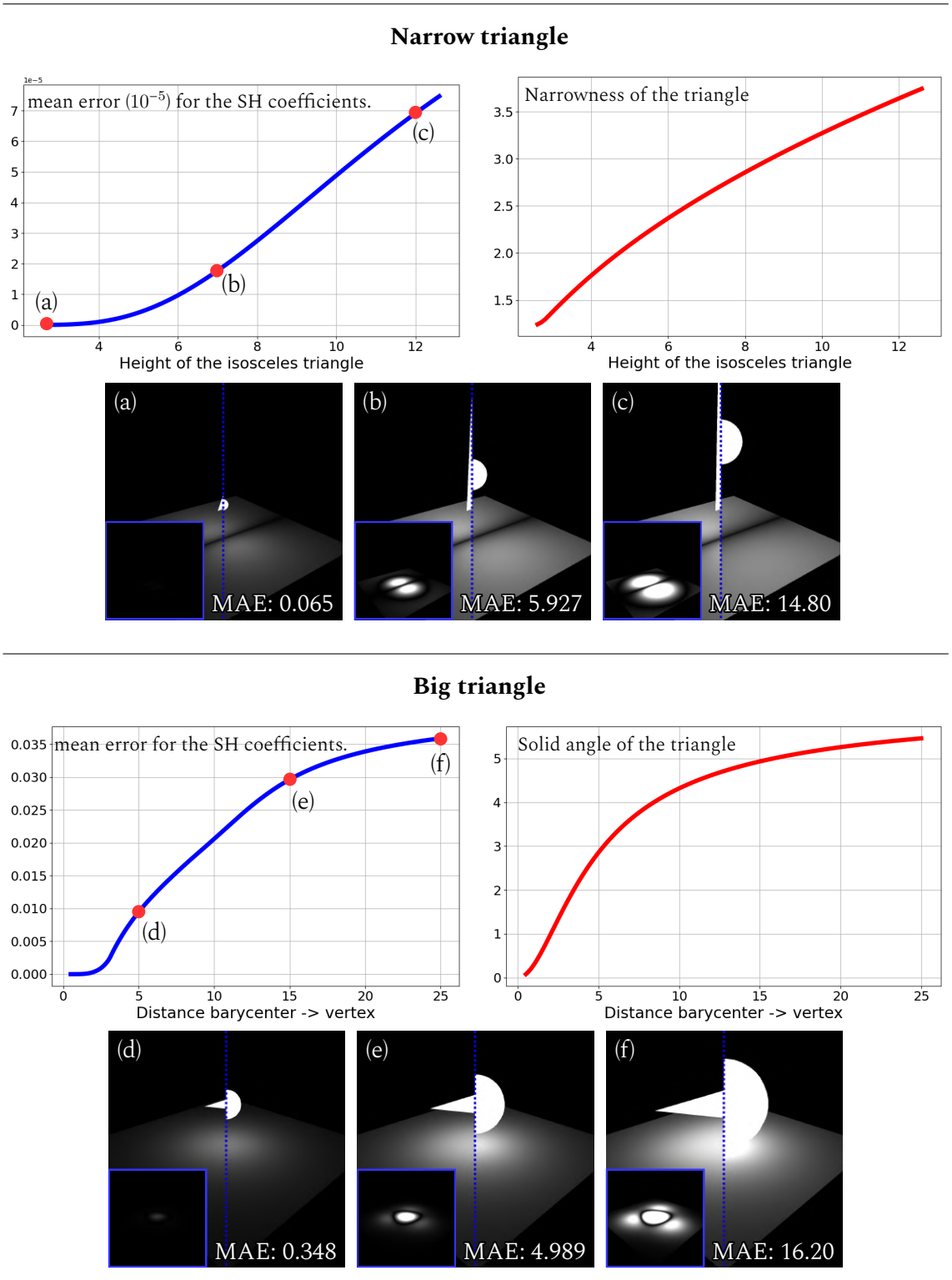


Figure 4.12: *Limitations and failure cases of our polygonal source approximation. Narrow triangles (first row) are known to be problematic in rendering. The approximation error is proportional to the* narrowness *of the triangle. Large triangles (second row) introduce an error related to their subtended solid angle. Insets show* 10× *absolute errors.*

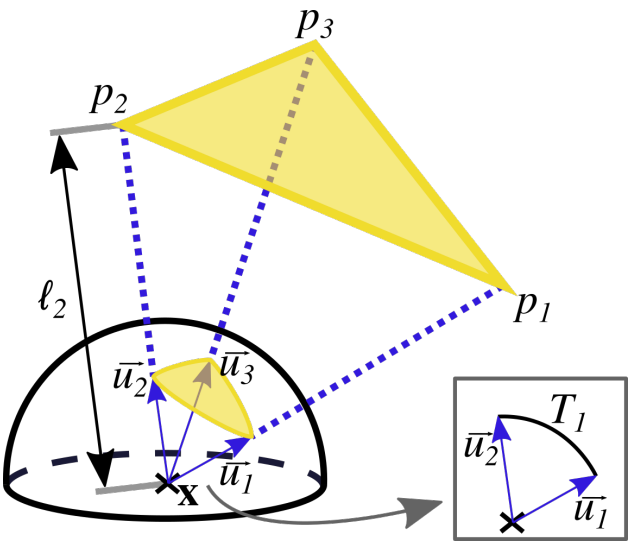


Figure 4.13: *Notations and projection of a triangle on the unit hemisphere.*

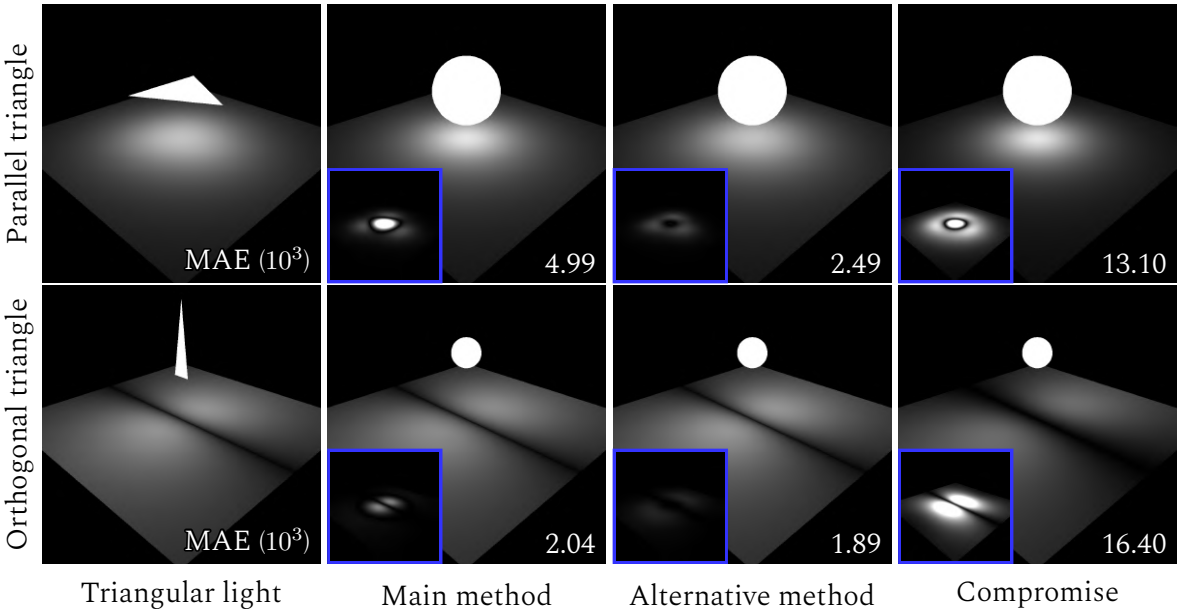
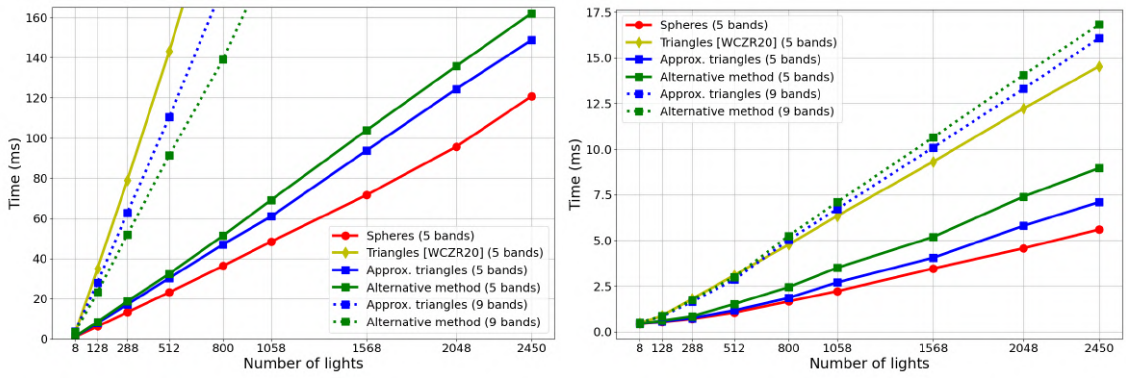
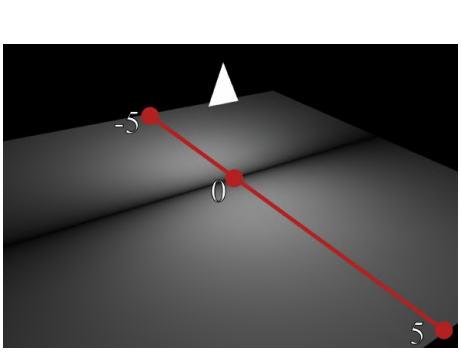


Figure 4.14: *Comparison of the different methods to approximate polygonal lights. Insets show* 10× *absolute errors. The alternative method, thanks to a good approximation of the solid angle, allows a better approximation of the lighting produced by polygonal lights than the main method. The alternative method looks for a similar solid angle value for the triangle and the sphere at each shaded point, to avoid some computation step the compromise method approximate the solid angle of the triangle. However, the solid angle approximation is too large and produces a poor quality result (Figure 4.16).*

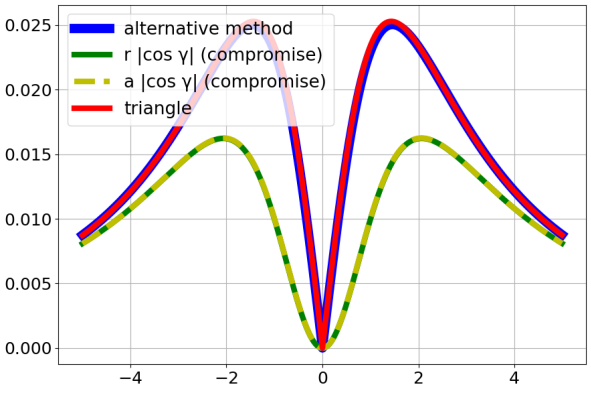


Without interpolation Grid construction

Figure 4.15: *Timings to compute the SH coefficients for each pixel (without interpolation) or for the 3D grid (grid construction) with their gradient for different number of SH bands. The main method (blue) correct the magnitude of each SH coefficients, thus, the greater the number of coefficients to be processed, the more expensive the method becomes. Indeed, the alternative method (green) tends to become less expensive than the main method when more SH bands are involved.*



(a) Figure (b) is computed along the red line.



(b) Solid angle for the different approximation methods.

Figure 4.16: *Comparison of the solid angle of the triangle with the different approximation method. The alternative method produces a good approximation of the solid angle of the triangle while the compromise method produce a poor approximation of the solid angle which explains the significant difference in error in the final rendering. (Figure 4.14)*

5 Global Illumination using Point-based Spherical Harmonics

PRT-based methods were initially proposed by the motivation of low-cost real-time global lighting evaluation. So far in this thesis, we have discussed the efficiency of direct spherical harmonic-based lighting. This chapter seeks to efficiently combine spherical harmonic-based lighting with global point-based lighting.

First, a quick overview of global point-based lighting methods is given (Section 5.1). It is also added a quick tour of the probe-based methods because the second contribution of this chapter is validated with a probe-based implementation.

The first contribution of this chapter is a new type of virtual light, named HVL (Section 5.2). These new virtual lights are based on the efficient SH projection of the radiance field produced by the spherical lights proposed in chapter 3. It is therefore discussed, here, the optimal distribution and configuration of the harmonics virtual lights as well as an in-depth comparison of the HVL with the state-of-the-art about virtual lights.

The second contribution proposes a point-based PRT method (Section 5.3). The PRT method precomputes traditionally the radiance transfers by path-tracing, which is independent for each shaded point. The method, proposed here, distributes virtual lights that are mutualized for all the shaded points. This allows to drastically reduce the precomputation time.

5.1 Point-based global illumination

Point-based global illumination converts the problem of global illumination where light rays can come from all directions into a problem of direct illumination produced by several hundreds or thousands (even millions) of virtual light sources. The idea is to distribute from the primary light, virtual sources that capture the incident lighting and redistribute it as direct lighting. This model is particularly efficient for simulating indirect lighting with a single bounce, but can also be generalized by distributing the virtual sources over as many bounces as necessary to simulate the global illumination. This section briefly reviews the main techniques and improvements of point-based global illumination with particular emphasis on the artifacts involved in these methods. Indeed, the contributions of this chapter aim to avoid these artifacts. For additional information on these methods, we refer the reader to the state of the art [RDGK12, DKH+14].

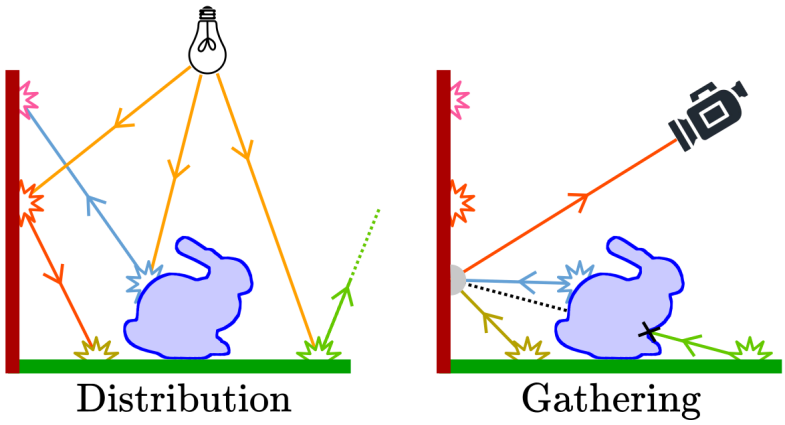


Figure 5.1: *The instant radiosity algorithm [Kel97] is based on two steps, the algorithm first distributes a set of VPLs from the primary source and then gather the lighting produced by the VPLs.*

5.1.1 Instant radiosity

The instant radiosity algorithm is composed of two steps [Kel97] (Figure 5.1), the distribution of the virtual sources and the gathering of the light their produce. The method derives its name from the radiosity, the radiometric quantity (Section 2.1.2), expressing the light emitted from a surface element. Indeed, distributing a VPL on each bounce of the light characterizes efficiently the radiosity of the scene.

The VPLs are distributed by a quasi-random walk [Kel97]. As in any quasi-Monte-Carlo method (Section 2.1.4), the generation of the light paths is done by following low discrepancy sequences. Thus, the VPL being distributed on the surface of the scene $\mathcal{S}$, it becomes more relevant to use the rendering equation formulated on the surface rather than in its directional form (Section 2.1.3):

$$L(\mathbf{x} \to \mathbf{z}) = \int_{\mathcal{S}} L(\mathbf{y} \to \mathbf{x}) F(\mathbf{y} \leftrightarrow \mathbf{x} \leftrightarrow \mathbf{z}) V(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathcal{S}_{\mathbf{y}} \ . \qquad \text{(2.16 revisited)}$$

Using M VPL, the rendering equation is approximated by a weighted sum of their contributions:

$$L(\mathbf{x} \to \mathbf{z}) \approx \frac{1}{M} \sum_{i=1}^{M} L_i(\mathbf{y}_i \to \mathbf{x}) F(\mathbf{y}_i \leftrightarrow \mathbf{x} \leftrightarrow \mathbf{z}) V(\mathbf{x}, \mathbf{y}_i) G(\mathbf{x}, \mathbf{y}_i) \qquad (5.1)$$

where $\mathbf{y}_i$ and L_i are respectively the position and radiance emitted toward $\mathbf{x}$ of the i-th VPL. For the visibility, since rasterization is synonymous with efficient parallelization, the most efficient way is to use a traditional rasterized shadow map [Wil78] at each $\mathbf{y}_i$ to compute the visibility with different shaded points $\mathbf{x}$.

Isotropic and anisotropic VPL The initial formulation of instant radiosity uses only isotropic VPL, *i.e.* emitting the same radiance in all directions, for any $\mathbf{x}$, $L_i(\mathbf{y}_i \to \mathbf{x})$ is constant. This makes it much easier to compute their contribution. However, the use of anisotropic VPL is more consistent when specular surfaces are involved. The use of anisotropic VPL is more annoying and notably a source of artifacts (Section 5.1.4).

The instant radiosity approach may be seen as distributing photons and each bounce is used as a virtual source. In this sense, it is similar to the photon mapping approach [Jen01, HHS05]. However, photon mapping relies on a density estimation to express the lighting. While the instant radiosity use a few hundred or even a thousand of photons, the photon map tends to use millions of photons. The simulation of photons is particularly

useful to generate light effects related to their convergence or divergence, such as caustics. In optics and mathematics, a caustic is the envelope of light rays undergoing reflection or refraction on a surface or curve. Nevertheless, caustics are a high frequency phenomenon requiring the simulation of many photons to be captured. Making photon maps more efficient to precisely simulate these phenomena, while the instant radiosity if more efficient to generate a low frequency global illumination. Wald *et al.* [WKB+02] combine both approaches to generate the two lighting effects efficiently.

5.1.2 Distribution

The initial formulation of instant radiosity [Kel97] distributes the VPL only according to the position of the primary light. Hence, depending on the position of the camera, VPLs may not contribute, or very little, to the final image. To avoid superfluous VPLs, importance sampling methods can be applied. Segovia *et al.* [SIMP06] distribute the VPLs by tracing paths from the camera and from the light and then connecting them explicitly. They keep the most relevant VPLs and then resample paths until they get the desired number of VPLs. Georgiev and Slusallek [GS10] reject the less relevant VPL with a probabilistic heuristic based on the contributed power of the VPLs. Segovia *et al.* [SIP07] create mutations of the most relevant path to create equally relevant paths that would be difficult to generate from scratch and thus position the VPLs efficiently.

To efficiently resolve the visibility, each VPL may have its own shadow map, this increases the cost to distribute the VPLs. Laine *et al.* [LLK07] alleviate this distribution cost in an interactive context using an incremental approach. By limiting their use to a single light bounce, a VPL is necessarily visible by the primary light to be considered as valid. A set of VPL is precomputed with their respective shadow map and at run-time, the validity of each VPL is computed according to the primary light position. The invalid VPLs are deleted and a new set of VPLs is sampled to replace them. VPLs are processed in subsets and processed separately during different frames to spread the distribution cost over several frames.

Shadow maps project the scene to a point $\mathbf{y}$ as an image and keep only the depth information at each pixel to compute the visibility of any point $\mathbf{x}$. Dachsbacher *et al.* [DS05] extend this principle by keeping reflective information, position, normal and color of the surface element, in the so-called *Reflective Shadow Map* (RSM). Each pixel of the RSM is considered as a potential VPL. Such approaches can benefit from an efficient rasterization, where approaches relying on ray tracing are more difficult to parallelize. Nevertheless, recent hardware offer massive ray-tracing capabilities allowing millions of VPLs to be distributed in a few milliseconds [LY19, YY17].

5.1.3 Gathering

The light produced by the VPL, must be gathered at each shaded point. The forward approach is to treat each shaded point independently summing the contribution of each VPL. However, this approach introduces unnecessary calculations. Indeed, any VPL will not necessarily contribute to the shaded point and therefore computing its contribution becomes useless. The works proposed in this thesis are not about improving the gathering of the VPL contributions, the approaches discussed in this section are orthogonal to our contributions.

Ritschel *et al.* [REG+09] ensure that the contribution of the virtual lights is never zero by clustering the virtual lights so that the solid angle of the cluster at the shaded point

is greater than a minimum solid angle. Dachsbacher and Stamminger [DS06] reverse the gathering, the contribution of the VPLs is splatted to each pixel. For each VPL, a splat in screen space is computed to select pixels where the VPL contributions are accumulated. Nichols and Wyman [NW09] note that a VPL have a similar contribution for two nearby pixels, in particular when the VPL is located on a diffuse material. Thus, the splats are refined into sub-splats according to geometry and light discontinuities and the VPL contribution is computed per sub-splat instead of per pixel. Forward light cut [LDdLRB16], not to be confused with the lightcut methods described later, propose a stochastic approach by distributing one VPL per triangle of the scene, each VPL having a random support function which define the spatial influence of the VPL and hence the splatting radius. While the splatting methods compute the spatial extent of the VPL contribution in screen space using bounding boxes, the culling methods compute this in world space using ellipsoids. This allows a shaded point to compute whether the contribution of a VPL will be zero or very small in order to early reject it. For these purposes, Yusuke and Tokuyoshy use a stochastic fall off of the VPL [TH16] and an ellipsoid fixed according to the GGX parameters of the material [TH17]. Fujieda *et al.* [FTH22] adapt this work to participating media.

Clustering the VPL reduces the number of contributions to compute to the number of clusters. The lighcut [WFA+05] represent the cluster by a tree according to the proximity of the VPL and their orientation. The root contains all the VPL and each leaf contains one VPL. A cut in the tree is computed so that any node of the cut satisfy an error criterion. Walter *et al.* refined this principle to multidimensional integration of the luminance field [WABG06] and bidirectional approach to also distribute VPL from the camera [WKB12]. Davidovič *et al.* [DKH+10a] propose an adaptive VPL stratification, the VPL are batched to reduce the global memory cost and then being scalable on GPU.

Computing the global illumination with VPL can be seen as a computation of a 2D matrix where the columns are the VPL and the rows are the shaded points. Each element of the matrix describes the contribution of one VPL to one pixel. Hašan *et al.* [HPB07] render random shaded points (rows) to select the representative VPL (column) and then render all the shaded points with only the representative VPL. Ou and Pellacini [OP11] use the same principle but on subsets that contain similar shading points. Hašan *et al.* [HVAPB08] generalize the principle with a third dimension, the time, to compress the VPL contributions for animations.

Visibility Visibility queries are often the bottleneck of point-based methods. The visibility of each VPL has to be computed for each shaded point, the number of queries can become very important. Rather than computing the visibility by ray casting, computing shadow maps [Wil78] at each VPL allows for efficient on-the-fly queries from any shaded point. The use of parabolic maps allows simplifying shadow maps, where a shadow map will directly characterize the visibility for the sphere, or hemisphere [OBM06,BAS02]. Moreover, the use of VPL clusters allows reducing the number of shadow maps to compute to the number of clusters [DGR+09]. Instead of lowering the number of shadow maps to be computed, the geometry can be modified to lower the computation complexity of each shadow map, *e.g.* Olsson *et al.* [OSK+14] use occluders clusters. Ritschel *et al.* [RGK+08], in their *Imperfect Shadow Maps*, use a point cloud of the scene where each point is randomly sent to a VPL to contribute to its shadow map. The shadow maps thus created have holes that are filled in by a surface reconstruction step [MKC07]. The method is extended to be adapted to the position of the camera [REH+11].

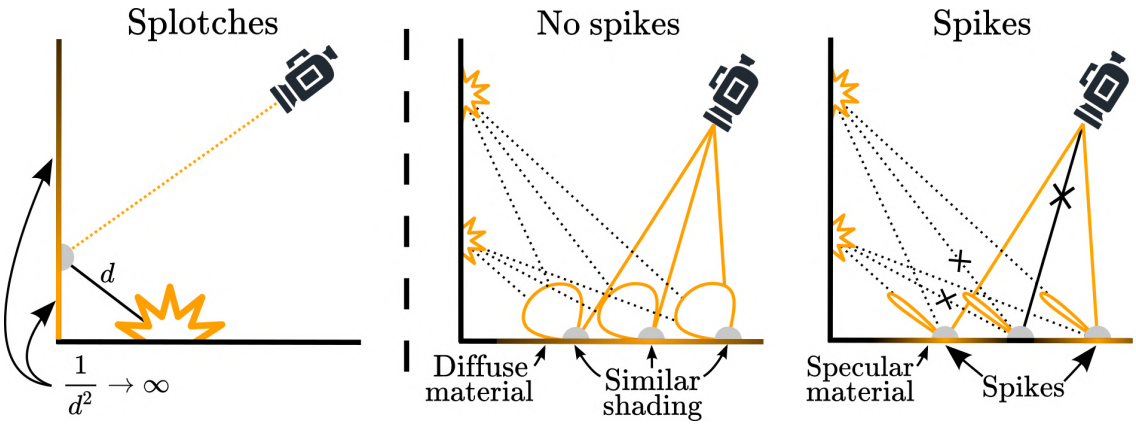


Figure 5.2: *Shading with VPL implies the inverse of the squared distance which tends to infinity when the shaded point approaches the VPL, thus creating splotches. When using highly specular materials, spikes appear if the density of VPL is not sufficient because the contributions of VPL will not be caught at consecutive shaded points.*

5.1.4 Artifacts

The VPLs are point lights, *i.e.* the radiance received at a shading point corresponds to a Dirac. This singularity produces rendering artifacts. In this thesis, we identify two types of artifacts related to two different causes that we call *splotches* and *spikes*. These artifacts are often confused in the literature. Banding artifacts can also occur if too few virtual sources are used compared to the size of the scene. They also appear if the quality of distribution of the virtual sources (or clustering) is poor.

Splotches The geometric factor of the rendering equation formulated on the surfaces is composed of a division by the squared distance between the shaded point $\mathbf{x}$ and the VPL at position $\mathbf{y}$ (Section 2.1.3). This division comes from the computation of the solid angle of the infinitesimal surface element of the VPL:

$$G(\mathbf{x},\mathbf{y}) = \frac{(\vec{\mathbf{xy}}\cdot n_{\mathbf{x}})(\vec{\mathbf{yx}}\cdot n_{\mathbf{y}})}{\|\mathbf{x}-\mathbf{y}\|^2}\,. \qquad \text{(2.17 revisited)}$$

Hence, G tends to infinity as the shaded point and the VPL get closer (Figure 5.2 - left). A common workaround is to clamp the geometric factor but this introduces a bias. Indeed, by bounding G there is an area around the shaded point whose light contribution is not fully taken into account. Some works propose to sample new paths with ray-casting in order to approximate a compensation term [KK06, DKH+10b, ENSD12]. Instead of ray casting, Novák *et al.* [NED11] approximate the compensation of the bounded region with a screen space method.

Spikes The spikes appear because of the specularity of the scene. Considering a shaded point capturing the radiance emitted by a VPL, a nearby shaded point will not necessarily capture the radiance of the VPL if the surface is too specular (Figure 5.2 - right). Thus, if the density of VPL is not sufficient in this kind of area, spikes appear on specular surfaces. Common workaround are to limit the glossiness of the scene. Therefore, more VPL must be used to capture the variations in lighting. In other words, the scene must be sufficiently covered with VPL. Instead of using more VPL, the *Virtual Spherical Light* (VSL) [HKWB09] inflate the VPL, avoiding the spikes by covering the scene better. Moreover, the solid angle in the geometric factor G becomes that of the sphere and avoids the division by the

squared distance, thus also avoiding splotches. However, the light contribution of a VSL is integrated over the area contained in the VSL and is computed by Monte-Carlo. In contrast, the *Harmonics Virtual Light* (HVL) model that is proposed in this thesis is a simplification of the VSL model analytically integrating the HVL contribution through spherical harmonics (Section 5.2). Yuksel [Yuk20] propose an attenuation for VPL based on the VSL model to avoid the splotches artifacts. Similar approaches as VPL/VSL has been adapted to participating media with the *Virtual ray/beam light* [NNDJ12a, NNDJ12b, VGS+19]. The *Virtual Spherical Gaussian Light* (VSGL) [Tok15b] avoid the spikes by approximating a dense set of VPL with a spherical Gaussian.

So far, we have explained the spikes when the shaded points are on specular surfaces Similarly, spikes appear also when the shading point is on a diffuse surface and the VPL are on a specular surface and the radiance they emit is anisotropic. VPLs are traditionally distributed at the bounce of a single light path, so the radiance they emit will be concentrated around the reflection direction of the path for specular material. This creates spikes when a shaded point is located in that direction. Rich-VPLs [SHD15] propose to route several paths through a VPL and store the outgoing radiance as an octahedral map. Thus, the radiance emitted by the Rich-VPL is the real radiance emitted at the VPL position and not a subsampling concentrated around an axis. This avoids spikes when VPLs are located on specular surfaces but not spikes for shading points located on specular surfaces. These artifacts are also minimized by better positioning strategies of VPL, *e.g.* Li *et al.* [LWLY22] distribute the VPL following a blue noise.

Flickering The flickering artifacts are related to splotches and spikes. When the scene is dynamic, the gathering of lighting is not necessarily consistent between each frame. As splotches and spikes, this problem does not exist if enough virtual lights can be gathered. Thus, efficient lighting gathering methods reduce the problem by taking into account many virtual lights, such as clustering methods [PKD12]. One of the best solutions relies on consistent temporal sampling of the virtual lights [BBH13].

5.1.5 Basis functions

SH allow storing spherical functions efficiently as a reduced set of coefficients instead of a discretization on the sphere, thus reducing the need of memory. For instance, Good and Taylor [GT05] use them in a photon map approach to store the photons contributions, avoiding an explicit storage of the photons. Similarly, Christensen [Chr08] projects the radiance emitted by VPLs/surface elements on the SH but also projects the radiance emitted by groups of VPLs. Thus, when VPLs are far from the shaded point the estimation of the radiance received by the group of VPL is simply a reconstruction from the SH projection. These two methods use SH as a simple storage medium, while the HVL model (Section 5.2) we propose takes advantage of the multiple SH properties, *e.g.* SH convolution (Section 2.2.10). The VSGL [Tok15b] method uses spherical Gaussians to estimate the radiance received from a group of VPL. In contrast, the HVL method only projects the radiance onto the SH received from one VPL (or VSL to be more exact). Wang *et al.* [WMB15] encode the radiance emitted by the VPL with wavelet, and use a wavelet-inspired tree structure similar to the lightcut structure (Section 5.1.3) allowing to compute the radiance of VPL cluster efficiently. However, their method is designed for offline rendering while the HVL are designed for interactive rendering.

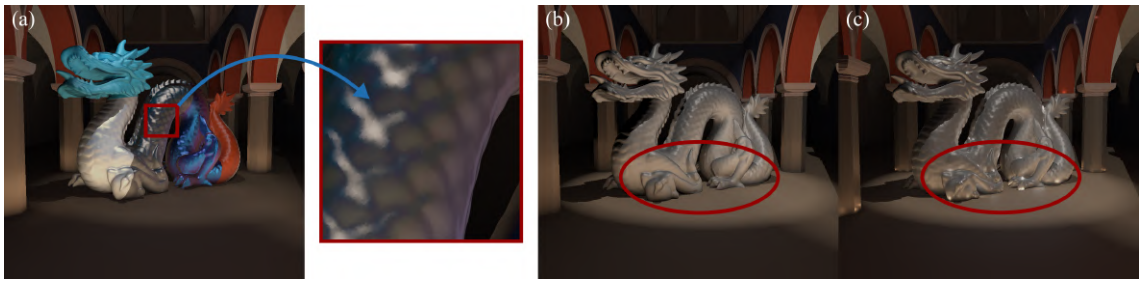


Figure 5.3: *(a) Harmonics Virtual Lights (HVLs) can render global illumination with arbitrary BRDF at no extra cost, while for example VPLs have an additional cost to use numerical material. (b) HVLs do not produce the classic artifacts of (c) the Virtual Point Lights methods.*

5.2 Harmonics Virtual Light

The *Harmonics Virtual Light* (HVL) model is a new type of virtual lights based on the efficient SH projection of the radiance field produced by spherical lights (Section 3.1). The HVL avoid the singularity of the VPL by using spherical lights (Figure 5.3). Thus, the HVL model can be seen as a simplification of the *Virtual Spherical Light* (VSL) model, we refer you to Section 5.1.4 for more details. This section focuses on the HVL model and its comparison with the VSL model and the *Virtual Spherical Gaussian Light* (VSGL) model.

5.2.1 Lighting pipeline

The HVLs rendering pipeline (Figure 5.4) is built over the SH projection of the BRDFs as unified material representation. This representation is precomputed once for all for

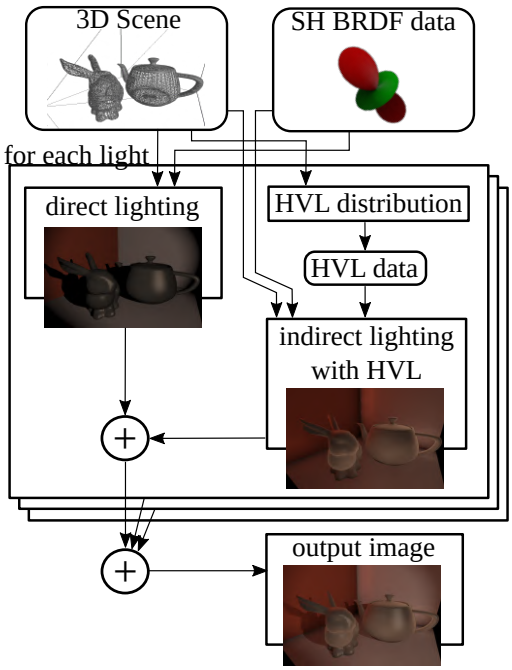


Figure 5.4: *Overview of the HVLs rendering pipeline. The inputs are a 3D scene with dynamic objects and lights, along with precomputed BRDF spherical harmonics coefficients. For each primary light of the scene, direct lighting is computed using SH framework. Indirect lighting starts with the distribution of HVLs from primary lights, followed by the gathering of HVLs contributions at each pixel. This pipeline allows using parametric like measured BRDF at the same cost.*

each material (Appendix B). Our two passes indirect lighting pipeline is similar to previous many-lights approaches (Section 5.1). For each frame, the first pass distributes HVLs according to primary lights; it stores the SH coefficients of the reflected luminance field of each HVL in an HVLs cache. The second pass gathers the contributions of the HVLs from the cache by computing the convolution between light emitted by HVLs and the material at the shaded point location. The light emitted by HVL is the result of the reconstruction of the HVL reflected luminance field in the direction of the shaded point. Thus, HVLs are anisotropic spherical lights whose emission intensity depends on the direction ω_j, the incident direction on the receiving shaded point. Figure 5.5 shows a typical one-bounce light path evaluated by our technique. The final image is then computed as the sum of direct and indirect contributions from each primary lights.

We formulate the indirect lighting $L_o(\mathbf{x}, \omega_o)$, leaving the shaded point x in the direction ω_o as

$$L_o(\mathbf{x}, \omega_o) = \sum_{s \in S} \Big(\sum_{j \in H_s} L_{j,o}(\mathbf{x}, \omega_o, \omega_j) \Big) \tag{5.2}$$

where S is the set of primary lights, H_s the set of HVLs generated from light s, ω_j the direction from x to the HVL j and $L_{j,o}(\mathbf{x}, \omega_o, \omega_j)$ its luminance contribution. According to the rendering equation (Section 2.1.3), this contribution is expressed as

$$L_{j,o}(\mathbf{x}, \omega_o, \omega_j) = \int_\Omega L_j(\mathbf{x}, \omega_i)\, F(\mathbf{x}, \omega_i, \omega_o) \max(0, \omega_i \cdot n_\mathbf{x})\, \mathrm{d}\omega_i \tag{5.3}$$

where $L_j(\mathbf{x}, \omega_i)$ is the incoming luminance at $\mathbf{x}$ emitted by the HVL j. As this luminance is only perceived in the solid angle subtended by the HVL, the integral domain is reduced to this solid angle Q_j. We approximate this equation by considering that the HVL emission is constant over the solid angle:

$$\begin{aligned} L_{j,o}(\mathbf{x}, \omega_o, \omega_j) &\approx L_j(\mathbf{x}, \omega_j) \int_{Q_j} F(\mathbf{x}, \omega_i, \omega_o) \max(0, \omega_i \cdot n_\mathbf{x})\, \mathrm{d}\omega_i \\ &\approx L_j(\mathbf{x}, \omega_j) \int_{Q_j} F'(\mathbf{x}, \omega_i, \omega_o)\, \mathrm{d}\omega_i \end{aligned} \tag{5.4}$$

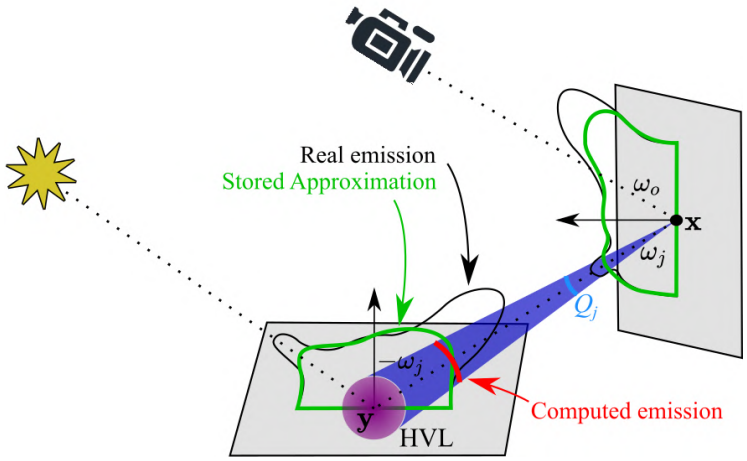


Figure 5.5: *Indirect lighting evaluation for one shaded point and one HVL. The HVL, in purple, uses a low-frequency approximation of the light emission corresponding to the first bounce of light emitted by the primary light. It's contribution to the shaded point at x is evaluated by projecting the HVL luminance onto SH and then convolving with the BRDF at the shaded point's location. The HVL emission is computed as constant over its subtended solid angle.*

where $F'(\mathbf{x}, \omega_i, \omega_o) = F(\mathbf{x}, \omega_i, \omega_o) \max(0, \omega_i \cdot n_\mathbf{x})$ and we note $\mathbf{F}$ the SH projection of F and $\mathbf{F}'$ the SH projection of F'. $\mathbf{F}$ will be needed to compute the HVL emission value (Section 5.2.2). As explained in Appendix B, $\mathbf{F}$ (and $\mathbf{F}'$) is a tabulation of the SH coefficients. Hence, $\mathbf{F}_\omega$ (or $\mathbf{F}'_\omega$) is the vector of SH coefficient corresponding to the direction ω.

We rely on the number of bands for spherical harmonics evaluation as a tradeoff parameter between accuracy and speed. To reach interactive performance, we typically use 1-5 bands to evaluate HVLs emission and 0-10 bands for the SH convolution on shaded points. This allows to represent low-frequency luminance field for HVLs and medium-frequency BRDF for the shaded point. Our approach scales well regarding high band limit and is able to render thousands of HVLs on 0-20 bands in a few seconds.

5.2.2 HVL luminance contribution

The luminance contribution of each HVL corresponds to the computation of Equation 5.4. This equation is expressed as

$$L_{j,o}(\mathbf{x}, \omega_o, \omega_j) \approx L_j(\mathbf{x}, \omega_j) \mathbf{L} \cdot \mathbf{F}'_{\omega_o, \mathbf{x}} \tag{5.5}$$

where $\mathbf{L}$ corresponds to the vector of SH coefficient representing the radiance field produced by the spherical light (Equation 3.11). The incoming luminance at $\mathbf{x}$ emitted by the j-th HVL is defined as

$$L_j(\mathbf{x}, \omega_j) = \Phi_j (\mathbf{F}_{\omega_l, \mathbf{y}} \cdot Y(-\omega_j)) G_j(-\omega_j) \tag{5.6}$$

where $\mathbf{y}$ and ω_l are respectively defined as the position of the HVL and its direction to primary light (Figure 5.6). Φ_j is the power of the HVL, corresponding to the power of the primary light divided by the number of HVL. $Y(-\omega_j)$ is the vector of the evaluation of the SH basis function in direction $-\omega_j$ and thus, the dot product corresponds to the reconstruction of the emission function towards the shaded point (Equation 2.37). Isotropic HVL reconstruction only use band 0 coefficient and skip SH evaluation.

The geometric factor $G_j(-\omega_j)$ corrects the energy taking into account the HVL light source spherical shape:

$$G_j(-\omega_j) = \frac{1}{\pi r_j^2} \max(0, n_j \cdot -\omega_j) H(n_\mathbf{x}, \omega_j) \tag{5.7}$$

where n_j is the normal of the j-th HVL, r_j its radius, πr_j^2 is an approximation of the surface area of the scene that lies in the HVL. The dot product corrects the energy according to

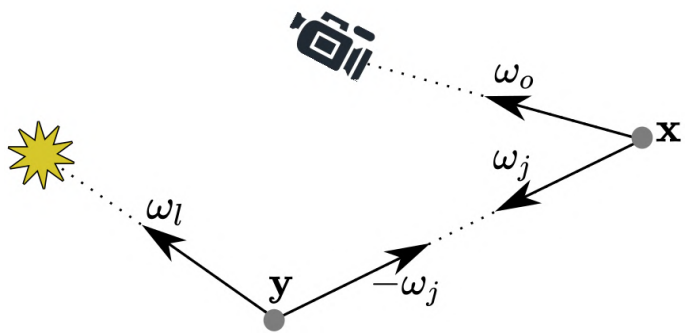


Figure 5.6: *Notations for a light path, $\mathbf{y}$ is an HVL and $\mathbf{x}$ a shaded point.*

the surface orientation. The factor H represents the proportion of the HVL that lies in the shading hemisphere, it corresponds to the intersection of two spherical caps where an approximation is given by Oat [OS07]. We derive the equation to our case, where the spherical cap is the hemisphere oriented by $n_{\mathbf{x}}$, the normal at point $\mathbf{x}$:

$$H(n_{\mathbf{x}}, \omega_j) = S\Big(\frac{\bar{a} - \min(\max(\cos^{-1}(n_{\mathbf{x}} \cdot \omega_j), \underline{a}), \bar{a})}{2a}\Big) \tag{5.8}$$

where a is the half-angle subtended by the spherical light source, $\bar{a} = a + \frac{\pi}{2}$, $\underline{a} = a - \frac{\pi}{2}$ and $S(x) = 3x^2 - 2x^3$ is the Smoothstep function. H could also be computed with the exact equation given in [OS07]. Since Monte-Carlo integration explicitly evaluates the clamped cosines, the VSL approach does not need to use such a corrective term.

When evaluating the HVL emission (Equation 5.6), we use the BRDF projection $\mathbf{F}$, without the cosine term $\max(0, \omega_i \cdot n_{\mathbf{x}})$ which is integrated into the geometric factor (Equation 5.7). This ensures that, when using low-frequency reconstruction (*i.e.* a few SH bands), the cosine is not over-smoothed and then, the energy is preserved.

The visibility term, although not present in the equations for the sake of clarity, is approximated by the visibility of the center of the HVL, as the VSL approach does.

5.2.3 Distribution and caching

We use a cache to store each HVL data: emission function, position, normal, BRDF index. To fill the cache, our implementation follows the principle of RSM (Section 5.1.2): the scene is rendered from each primary light and each rendered fragment stores HVL data in a set of 2D GPU textures. The BDRF index corresponds to the index in the precomputed material SH coefficients set. This set is stored as 2D textures, one per material. The memory requirement of the HVL cache spans from 16KB for 400 HVLs to 400KB for 10 thousands HVLs. This is negligible compared to the cost of storing the materials (Appendix B).

5.2.4 Radius heuristics

The radius of an HVL is an important parameter that impacts the rendered image quality. Intuitively, if HVLs do not sufficiently cover the scene, the rendered image exhibits the same kind of artifacts as VPLs (Section 5.1.4). But if HVL radius is too large, high frequency shading details will be lost as the HVLs contributions will overlap. Hence, we have to adapt HVLs radius so that there is the right amount of overlapping.

The HVL overlapping is directly linked to the HVL radius with regard to HVL density. As done in VSL [HKWB09], we define each HVL radius based on an estimation of local HVL density. Instead of searching for the closest neighbors, as proposed for VSL, we propose two dynamic heuristics relying on a regular HVL cache sampling to gather the HVL locations (Figure 5.7). Our first heuristic estimates the radius of an HVL from an approximation of the HVLs local density by sampling the HVL's 8-connected neighbors from the HVL cache:

$$r_1 = \frac{\sum_n w_n \, \|p - p_n\|}{\sum_n w_n}, \quad w_n = \frac{1}{|d - d_n| + \epsilon} \tag{5.9}$$

where r_1 is the radius of the HVL, d the HVL distance to its related primary light, p denotes the position in world space coordinates and $\sum_n$ sums over the 8-connected neighbors of the HVL, p_n and d_n being neighbor's position and distance from the primary light. This heuristic averages the distance of a HVL to its neighbors so that it overlap them and thus

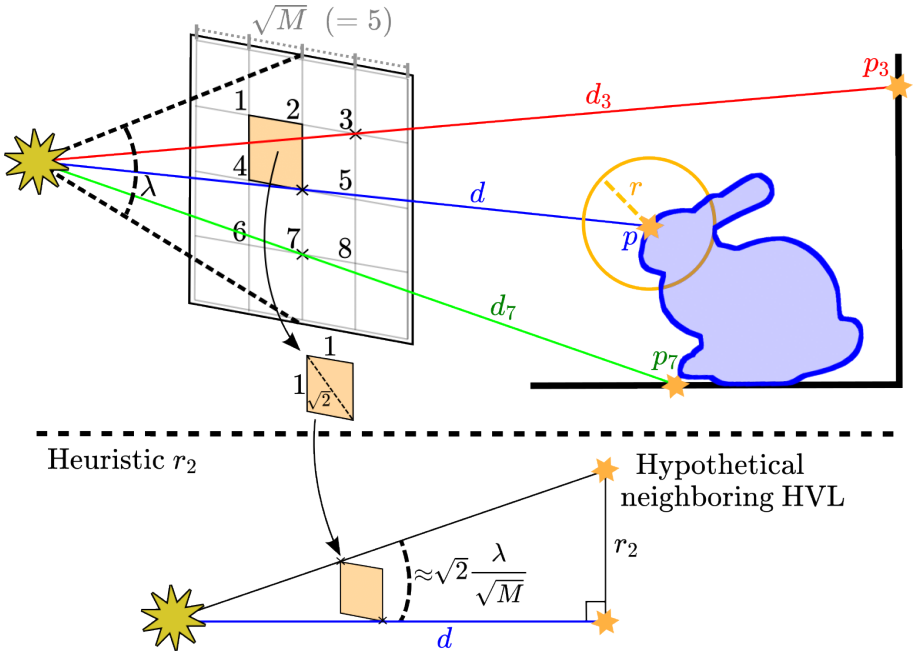


Figure 5.7: *Diagram of the proposed heuristics to compute the radius of the HVL.*

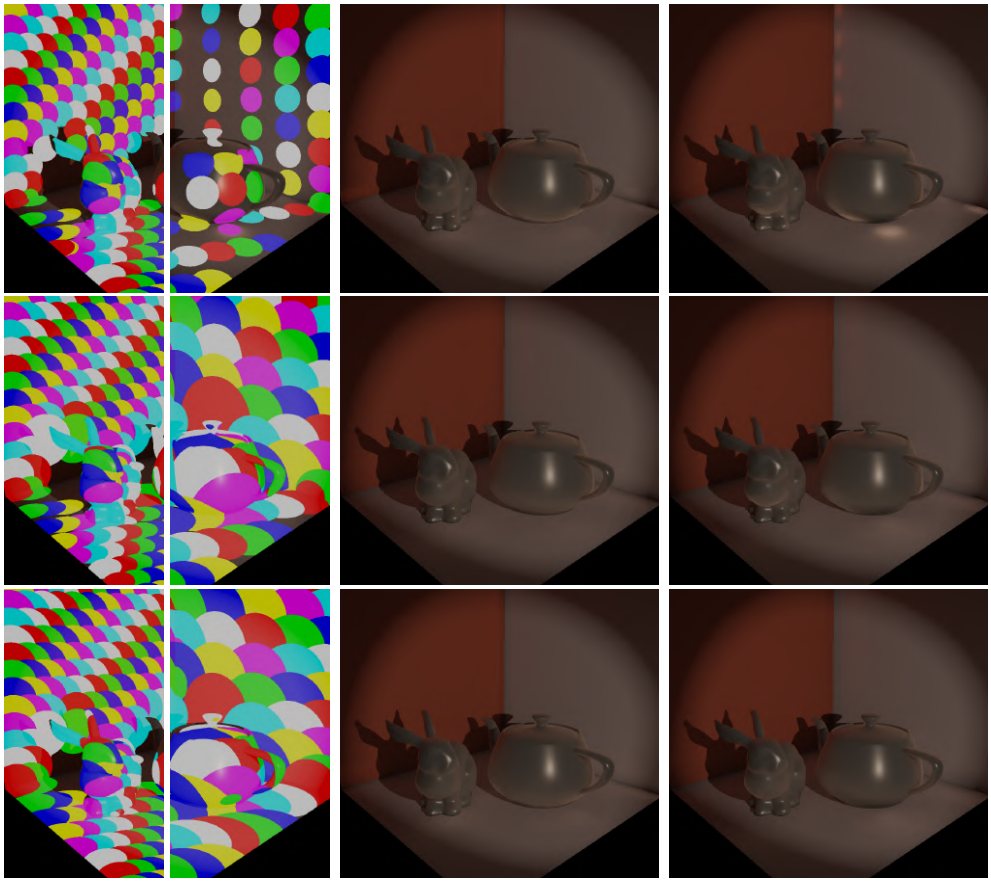


Figure 5.8: *HVLs coverage. Each colored circle corresponds to one HVL. From left to right: HVLs coverage with 400 and 100 HVLs; resulting image with 400 HVLs and then 100 HVLs. From top to bottom: fixed radius of 0.25, heuristic 1, heuristic 2 respectively.*

avoid spike artifacts. We weight the contribution of neighbors by the difference of distances from HVLs to light to avoid too large overlapping because of the scene geometry, such as the hole formed by the rabbit's ears in Figure 5.8. A small ϵ value is added to avoid a division by 0, should two HVLs distances to light be the same.

This first heuristic is based on the neighborhood of the HVL, making it expensive to compute. We propose a second heuristic (Figure 5.7), more suitable for real-time scenarios, independent of the HVL neighborhood by considering the FOV angle λ used to generate a square RSM instead. First, we approximate the angle γ formed by two diagonally adjacent HVLs by considering that the two HVLs are at the same distance from the light:

$$\gamma = \sqrt{2}\frac{\lambda}{\sqrt{M}} \tag{5.10}$$

where M is the total number of HVLs and $\sqrt{2}$ is the length of the diagonal. Starting from this, we approximate the distance between those HVLs, and hence the radius r_2 of an HVL, by considering a right triangle between the primary light and the two HVLs, where the right angle is at the HVL whose radius is sought:

$$r_2 = d\tan(\gamma) \tag{5.11}$$

where d is the distance of the light of the HVL and k is the same user-specified constant as for r_1. To avoid any trigonometric function evaluation, as γ is close from 0, we approximate the $\tan$ function by its Taylor series $\tan(x) \approx x + x^3/3$. The heuristic then becomes

$$r_2 \approx d\left(\gamma + \frac{\gamma^3}{3}\right) \tag{5.12}$$

The approximation of Equation 5.11 by Equation 5.12 is accurate. Indeed, in order to have an error of less than 0.01 for the evaluation of the function $\tan$ from its Taylor series, calculus hints that we must have $\gamma \lesssim 0.58$ and thus $\lambda/\sqrt{M} \lesssim 0.2$. So, for a spotlight of half-angle $45°$, $\sqrt{M} \gtrsim 3.92$, means that we need at least 16 HVLs for precise evaluation of the $\tan$ function. For a spotlight of half-angle $90°$, we need at least 64 HVLs. These requirements are far below the number of HVLs required in practice (*i.e.* at least a few hundreds). As shown in Figure 5.8, these heuristics allow having HVLs that overlap well, and thus covers the directly illuminated scene geometry, without having one HVL that swallow another one, which would reduce the impact of the number of HVLs. However, the balance remains fragile and artifacts may still appear, especially when the depth distribution of HVLs is heterogeneous because holes may appear in the coverage. To this end, and as in VSL, a user specified factor should be used to scale the computed radii. In our experiments, we obtained good results by setting this factor between 1 and 2.

5.2.5 Experimental results

Figure 5.9 illustrates how materials change the rendered image through the use of HVLs. Our method applies a low-pass filter on the lighting due to the use of SH, thus providing smooth and artifact-free results (Figure 5.10).

The HVL count has a linear impact on the computational cost, each HVL contribution being independent of the others. Increasing the SH band limit has a quadratic cost with respect to the band limit as the number of SH coefficients is equal to the square of the bands number. Table 5.1 confirms these trends. Thus, the computation time for HVLs

Figure 5.9: *Global illumination using HVLs with different measured materials. In our framework, materials can be changed dynamically without affecting rendering time. In the bottom row, only the materials applied to the chairs and the table are modified between the three pictures.*

is predictable, but is greatly impacted by the number of SH bands. Furthermore, if low-frequency BRDF are used, increasing the band limit will have no impact on the final picture quality. Using an adaptive band limit might keep the computational cost minimal but might also increase code divergence in the shader evaluation of the SH projection.

5.2.6 Comparison with VPL and VSL

To fairly compare HVL, VPL and VSL, we only use microfacet based BRDF (Figure 5.10 and 5.11). Indeed, if VPL/VSL computation time might be greatly impacted by the use of measured and SH encoded BRDF, our HVL method is agnostic to the used BRDF model and performs similarly on parametric or measured BRDF. In our comparison, we use the GGX microfacet distribution and evaluate the Fresnel reflectance using the exact term. We used material from MERL database, fitted on GGX by the work of Ribardière *et al.* [RBMS17]. The three methods are compared to path-traced references, rendered using pbrt-v3 [PJH16], using three metrics: RMSE, PSNR and SSIM. Table 5.2 shows timings and error metric for the three methods.

Scene	HVL number	Convolution band limit			
		3	5	7	9
Sponza Figure 5.3(a)	64	6.6	7.8	10.3	18.6
	196	19.8	23.5	30.4	56.1
	400	41.0	48.6	62.7	115.3

Table 5.1: *Timings (ms) for indirect lighting gathering from various HVLs numbers and SH bands for convolution on shaded points. The emission function for each HVL is computed with 3 bands.*

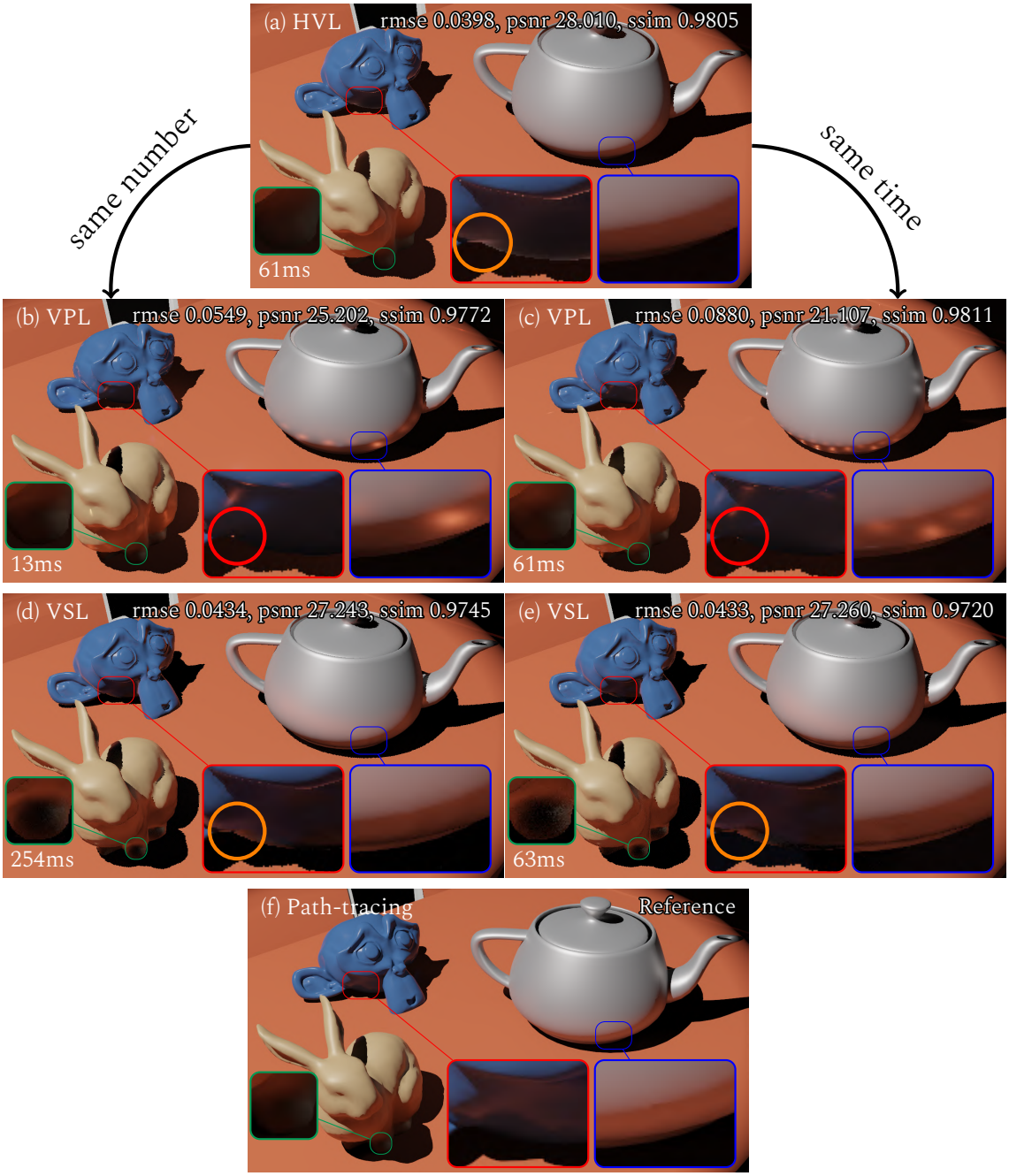


Figure 5.10: *Comparison between HVL, VPL, VSL and path-tracing. (a) uses 5 SH bands for convolution on shaded point, 3 bands for the emission function and 400 HVLs, (b) and (c) are resp. render with 400 and 1928 VPLs. (d) and (e) are render resp. with 400 and 256 VSLs and 25 and 9 importance samples of the shaded point's BRDF. HVLs and VSLs use heuristic r_2. (f) is a path-traced reference. Images are rendered at 1280x1024. The red circles show an artifact due to VPLs on the background and resulting from the lack of visibility management for secondary lights in our implementation. The orange circles highlight an overestimation of the indirect lighting due to the spherical nature of the HVL and VSL.*

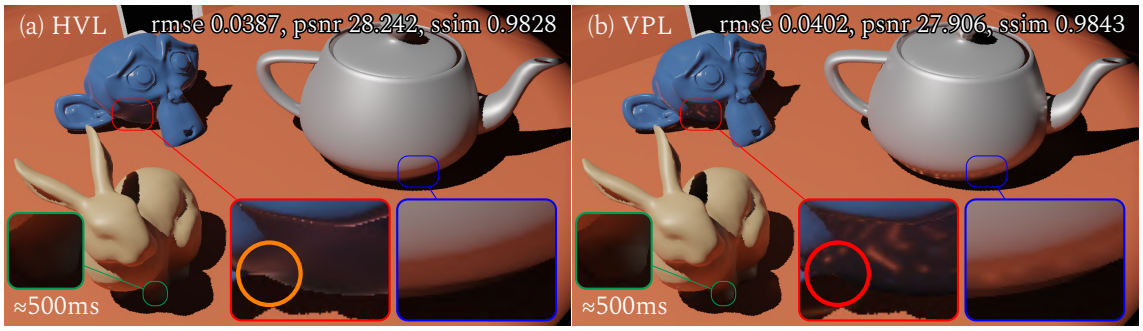


Figure 5.11: *Comparison of HVL and VPL at same time using same resolution and same heuristic for HVL as the figure above. (a) is rendered with 1500 HVLs and uses 9 SH bands for convolution and 3 SH bands for the emission of HVLs. (b) is rendered with 10000 VPLs.*

Figure 5.10a, 5.10b and 5.10d are computed using the same budget of virtual lights. To avoid a visual comparison bias, virtual lights are placed at the exact same locations for all three methods. Our method avoids the classical VPLs artifacts but is more computationally expensive. Using 25 samples per virtual light, VSL is even more expensive while still exhibiting noise. Aiming at interactive rendering, VSLs must use low sample count and results on noisy pictures (Figure 5.10e). To remove this noise, particularly visible on the bunny's feet, the number of samples must be at least tripled, with a similar increase on the computation time (Figure 5.10d).

With the same computation time budget, VPLs still exhibit artifacts despite using more virtual lights, whereas HVLs produce a smoother but artifact-free result (Figure 5.10a and 5.10c). VSLs produce a noisy result as only a low number of samples might be used (Figure 5.10e). Thus, HVLs are an interesting alternative to produce a fast low-frequency global lighting and gradually raise the final image in frequency. Figure 5.11a and 5.11b are computed given a time budget of 500ms. VPLs still exhibit artifacts despite the big number of virtual lights, while HVLs give a smooth result.

Light leaks artifacts may appear in some situations due to a combination of several factors (orange and red circles in Figure 5.10 and 5.11). Inaccurate visibility estimation is one of these factors when using virtual lights. The red circles show a light leak due to this visibility problem for VPLs due to sources located on the background. These artifacts are not related to the VPLs themselves but to the limitations of our implementation. For HVL and VSL (orange circles), the spherical nature of the sources is the main cause of artifacts, due here to sources located near Suzanne. The spherical nature of the sources generate an overestimation of the actual lit area. This is especially problematic when using a few sources, since smaller is the number of sources, larger is their radius. Finally, the frequency band limit resulting from the SH order for HVL smooths out these artifacts.

As shown by these experiments, HVLs always give a smooth but biased result, whatever the number of virtual lights or the time budget given. This robustness of our method is its biggest advantage for interactive lighting design. Note that while VPLs are less appropriate for interactive usages due to the high-variance estimation of indirect lighting with small number of VPLs, they define an unbiased estimator of indirect lighting. Hence, running simulation using VPL for a long enough time will always give a better result than HVL.

Figures		Parameters			Timings		Error metrics		
		lights	VSL samp.	SH bands	distrib.*	indirect light	rmse	psnr	ssim
VPL	5.10b	400	-	-	0.59	13	0.054	25.20	0.977
	5.10c	1928	-	-	0.59	61	0.088	21.10	0.981
	5.11b	10k	-	-	**2.44**	500	0.040	27.91	0.984
VSL	5.10d	400	25	-	0.59	254	0.043	27.24	0.974
	5.10e	256	9	-	0.59	63	0.043	27.26	0.972
HVL	5.10a	400	-	5	0.38	61	0.039	28.01	0.980
	5.11a	1.5k	-	9	**1.6**	500	0.039	28.24	0.983

* cache size is 1024^2 pixels (**bold**=$\mathbf{2048^2}$)

Table 5.2: *Timings (ms) and error metrics (path-tracing used as reference) detail for Figure 5.10 and 5.11. RSM for HVLs are cheaper because fewer data are manipulated (see Section 5.2.3 for details).*

5.2.7 Optimized zonal harmonics shading

HVLs are a general model that handle any kind of material, we derive an efficient shading when using a *decomposable BRDF* where the specular lobe is a circularly symmetric kernel, *i.e.* only representable with ZH. This method also relies on the circularly symmetric radiance produced by the spherical lights. Thus, the convolution has to be applied for each spherical light independently instead of once at the end with the accumulated radiance for all the lights. However, this improvement makes the method more competitive.

Diffuse contribution Ramamoorthi and Hanrahan use a 4×4 matrix to compute the diffuse contribution (Section 2.3.2) where the radiance is represented by a full set of SH [RH01]. This matrix can be truncated by a 2×2 matrix when using a circularly symmetric emission function which can be projected on ZH, instead of SH, by orienting the projection according to the direction of emission function

$$M = \begin{pmatrix} c_3\mathbf{L}_2^0 & c_2\mathbf{L}_1^0 \\ c_2\mathbf{L}_1^0 & c_4\mathbf{L}_0^0 - c_5\mathbf{L}_2^0 \end{pmatrix}$$
$$c_2 = 0.511664 \quad c_3 = 0.743125$$
$$c_4 = 0.886227 \quad c_5 = 0.247708\ .$$

The diffuse contribution can be evaluated according to the direction of the emission lobe ω_l. If we note $z = \omega_l \cdot \mathrm{n}$, the illuminance is computed as

$$E(\mathrm{o}) = \mathrm{o}^t M \mathrm{o} \quad \text{with} \quad \mathrm{o}^t = \begin{pmatrix} z & 1 \end{pmatrix}\ .$$

Hence, we can write

$$E(z) = c_3\mathbf{L}_2^0 z^2 + 2c_2\mathbf{L}_1^0 z + c_4\mathbf{L}_0^0 - c_5\mathbf{L}_2^0\ .$$

The diffuse contribution can be modulated by the visibility of the HVL center.

Specular contribution First, other function O other than the reflectance F or radiance L are taken out of the integral and approximated by the direction of the HVL center ω_l. The function O is the cosine of the rendering equation, but can also contain the visibility. Note that the same process can be applied to the visibility, where the visibility is approximated by

Figure 5.12: *Comparison with 900 VPL, 529 VSGL and 400 HVL, generated at equal time. Images are rendered at 1600×900. The HVL cache resolution is 322^2 pixels, and 14^2 pixels for one VSGL. The contribution of HVLs is on 15 SH bands. In this rendering, HVL are limited to circularly symmetric specular lobe to compute lighting only on ZH. Insets show VPL spike artifacts and VSGL or HVL robustness to those artifacts.*

the visibility of the HVL center. The integral is then computed by the dot product between the vector of SH coefficients, where the ZH radiance $\tilde{\mathbf{L}}$ is rotated in the SH frame coordinate of $\mathbf{F}$, where ω_l is the direction to the HVL in the frame coordinate of $\mathbf{F}$:

$$\int_\Omega F(\omega)L(\omega)O(\omega)\,\mathrm{d}\omega \approx O(\omega_l)\sum_{l,m}\mathbf{F}_l^m\mathbf{L}_l^m \approx O(\omega_l)\sum_{l,m}\mathbf{F}_l^m\sqrt{\frac{4\pi}{2l+1}}Y_l^m(\omega_l)\tilde{\mathbf{L}}_l\,. \tag{5.13}$$

However, as noted in [DSJN19] if $\mathbf{F}$ is a circularly symmetric kernel oriented toward the ω_r direction, thus only represented with ZH in the frame coordinate aligned with ω_r. $\mathbf{F}$ and $\tilde{\mathbf{L}}$ can be aligned in the same frame coordinate by aligning the z axis of each frame coordinate without caring for the phase alignment (x and y axis) since the two functions are circularly symmetric. The alignment of the z axis corresponds to a rotation of angle $\arccos(\omega_l\cdot\omega_r)$:

$$O(\omega_l)\sum_{l,m}\mathbf{F}_l^m\mathbf{L}_l^m \approx O(\omega_l)\sum_{l}\mathbf{F}_l\sqrt{\frac{4\pi}{2l+1}}Y_l^0(\omega_l\cdot\omega_r)\tilde{\mathbf{L}}_l \tag{5.14}$$

where $Y_l^0(\theta) = Y_l^0(\omega)$. Hence, the complexity to compute the specular contribution becomes $\mathcal{O}(N)$ instead of $\mathcal{O}(N^2)$, where N is the number of SH bands and only the ZH basis functions are evaluated instead of all the SH basis function.

5.2.8 Comparison with VSGL

We use our own implementation of a simplified version of VSGL [Tok15b], based on publicly available faster VSGL code [Tok15a]. This allows us to produce a fair comparison between both methods, but without importance or interleaved sampling. Our method benefits from faster computation time thanks to the ZH expansion, thereby limiting convolution to materials with a circularly symmetric specular lobe which is fair compared to VSGL, being itself limited to BRDF with a single lobe.

Method	Parameters		Timing (ms)	rmse	psnr	ssim
	number of VL	number of SH bands				
VPL	400	-	17	0.2428	12.296	0.8765
	900	-	39	0.2603	11.691	0.8852
	1156	-	52	0.2425	12.308	0.8831
HVL	400	5	23	0.2417	12.336	0.8592
		10	32	0.2336	12.629	0.8733
		15	40	0.2317	12.703	0.8785
		20	47	0.2314	12.713	0.8804
	1156	5	68	0.2406	12.374	0.8643
		10	93	0.2321	12.687	0.8806
		15	117	0.2297	12.776	0.8851
		20	140	0.2292	12.796	0.8859
VSGL	400	-	32	0.2617	11.645	0.8756
	529	-	40	0.2539	11.907	0.8740
	1156	-	87	0.2336	12.632	0.8762

Table 5.3: *Timings (ms) and metrics for the figures of comparison between HVL and VSGL.*

To summarize our implementation:

- The ZH projection for the light is done using our method for spherical light (Equation 3.10).
- The BRDF convolution is computed with the fast shading with zonal harmonics proposed in Section 5.2.7.
- The HVLs directional emission is computed by evaluating a GGX BRDF.
- Visibility of the HVLs (and VSGL) is not taken into account.

VSGL method approximates a set of nearby virtual light relying on RSM, as it is the most efficient method to quickly access a VPL and its neighbors, and taking advantage of RSM mipmaping. Our HVL method extends VSL and only relies on RSM to sample the position of the virtual lights. HVL are then less dependent on RSM, thus more flexible than VSGL. As shown in Figure 5.12, HVL and VSGL exhibit similar results without any spikes.

For a fair comparison, both methods are restricted to single-lobe BRDF only. Our VSGL implementation defines one VSGL per pixel group; HVL are restricted to single-lobe BRDF when using the ZH acceleration method. Moreover, this ZH acceleration makes the HVL method much more competitive than the full SH method (40ms for 15 SH bands).

Implementation details of VSGL

We implemented a simpler version of VSGL from the code available online of a paper on fast VSGL [Tok15b], because there is no code available for the full VSGL method [Tok15a]. We specify that one VSGL use two spherical Gaussians, one for diffuse and the other one for specular contribution. What changes compared to the full version is that it is only one VSGL without extras, no importance sampling and no interleaved sampling.

For a fair comparison between our method and VSGL, we slightly modified the code pub-

Figure 5.13: *Observation of the effect of the number of zonal harmonics using 400 HVLs. We clearly see the specular reflection on the ground becoming higher with higher frequency. However, we can also see that the rendering does not need to go beyond 15 ZH. Indeed, between 15 and 20 ZH, we do not observe any visible difference. The error values show it as well.*

lished by [Tok15b] to generate several VSGLs, while the author generates only one VSGL on the whole RSM. We distribute VSGLs and HVLs uniformly on the RSM. We typically use 14^2 pixels for each VSGL, and an RSM of 280^2 pixels for 400 VSGLs and 322^2 pixels for 529 VSGLs. We place HVLs or VPLs at the center of the pixel group used to define a VSGL when comparing with the same number of virtual lights.

Results and comparisons

Results are generated with a RTX 2080 on 1600×900 resolution. The strong specular reflection on the ground of Sponza is the most visible effect which varies depending on the rendering technique (Figure 5.12). We consider the result generated with path-tracing as a reference to compute the error metrics. We compiled in Table 5.3 the timings to generate these results and the resulting scores on RMSE, PSNR and SSIM.

The Sponza scene needs to use 15 ZH bands to catch the strong specular reflection on the ground (Figure 5.13). Figure 5.12 shows that at equal time, HVL and VSGL are able to render images with similar quality. With the same amount of virtual light and considering the same quality on this scene, VSGL are faster (Figure 5.14). However, on mainly diffuse scenes, we could lower the number of ZH bands while keeping a similar quality. And so, HVL would become faster than VSGL. Hence, the number of ZH bands can be used as a slider to manage the quality according to the scene used, being a trade-off between quality and rendering time. Also, the VPL method is obviously faster than HVL or VSGL but generates the traditional artifacts as shown in the (Figure 5.15) generated with the same budget of virtual lights.

Figure 5.14: *Comparison between VSGL and HVL. HVL need use enough ZH bands to match the rendering generated with VSGL.*

5.2.9 Limitations and future work

Primary light As shown in Section 5.2.2, our HVLs framework only considers primary light primitives from which incident direction is a Dirac. In order to manage area lights, the process of computing HVLs luminance contribution should be adapted. The SH coefficients of the incident luminance field due to the area light are to be stored on each HVL. Once the receiving shaded point is known, the SH coefficients for the emission of each HVL are computed according to the shaded point. The convolution of this emission function with the HVL incident luminance field will then result in the incident luminance field at the shaded point. Due to this additional convolution, the computation time of the method will increase.

HVLs visiblity The HVLs visibility could be approximated by taking into account only their center and the traditional VPL method to compute the visibility (Section 5.1.3). This approximation is only valid when HVLs radii are small. For large HVLs, visibility management is more complex. It might be interesting to project the visibility on SH, as done in PRT technique for dynamic scenes (Section 2.3.3), but this will require to evaluate a triple product ($\mathbf{V} \cdot \mathbf{L} \cdot \mathbf{F}$) instead of only a double product. The extra cost of visibility projection and triple product would prevent to reach interactive rendering time. Indeed, unlike like the double product, the triple product does not reduce to a dot product of vectors.

Efficient shading Our prototype implementation only uses a simple light gathering to compute HVLs contributions at each pixel, in which each pixel gets the contribution of all HVLs. This has a strong impact on the computation time when using numerous HVLs. For a better efficiency, a shaded point must ideally consider only HVLs that will have an influence on itself. Our proposal is orthogonal to any efficient virtual light gathering method (Section 5.1.3). As shown for the VSGL method [Tok15b], the shading time can be reduced by using interleaved sampling [KH01]. HVLs are band limited by construction, thus, they do not generate arbitrary high frequencies. This property is particularly useful in interleaved sampling methods to define a proper sampling strategy.

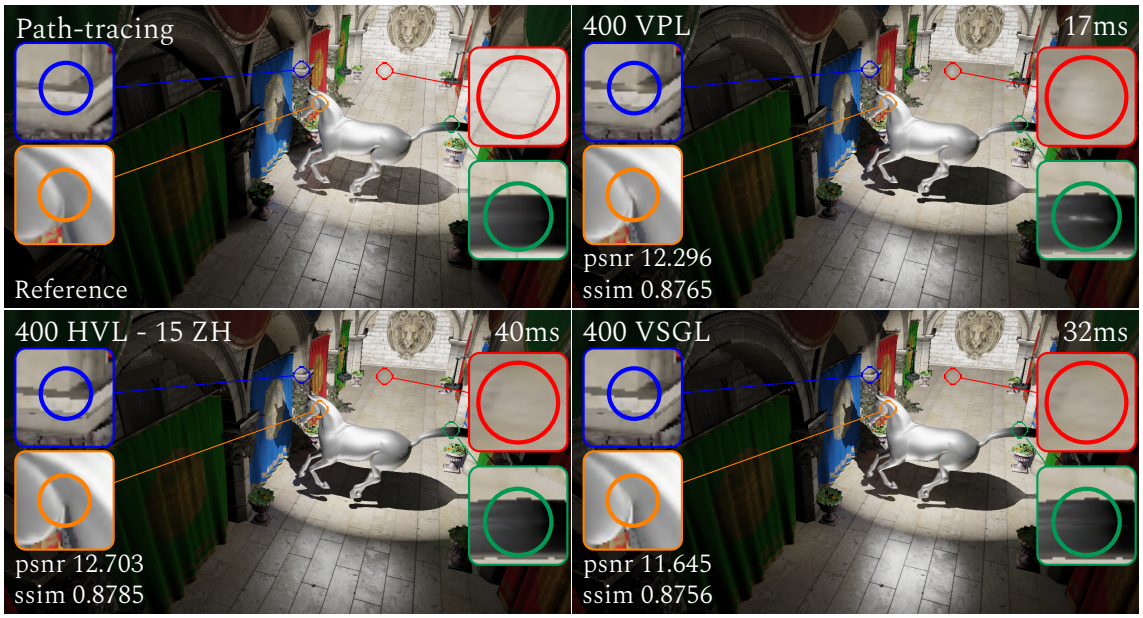


Figure 5.15: *Comparison between VPL, HVL and VSGL. Compared to VPL, HVL and VSGL do not produce artifacts with a low number of virtual sources, HVLs reduce artifacts through frequency smoothing and the spherical nature of the virtual sources.*

Hemispherical bases Although spherical in shape, HVLs only emit light in a hemisphere. While the current implementation accounts for this using the geometric term, we thought about using the hemispherical basis proposed by Gautron *et al.* [GKPB04]. Nevertheless, this requires to correctly adapt several computations to this basis, such as the integration of Legendre polynomials or the efficient evaluation of harmonics. Unfortunately, the tricks proposed by Sloan [Slo13] to efficiently evaluate spherical harmonics do not apply directly to hemispherical harmonics. Alternatively, the derivations could be adapted to a more recent basis, such as the $\mathcal{H}$-basis of Habel *et al.* [HW10].

Soft angular kernels In our proposal, as for VSLs, we consider that the emission is constant on the solid angle subtended by the spherical light. If VSLs rely on this hypothesis for efficient uniform sampling for Monte-Carlo integration, we use this to derive an analytical solution for HVL (Section 3.1). However, the solid angle boundary corresponds to an infinite frequency in the emission function that produces ringing artifacts in the SH approximation. To limit these ringing artifacts, one can use a soft angular kernel in the ZH projection of spherical lights (Equation 3.4). However, finding an analytical and efficient solution with an arbitrary kernel is challenging but should improve the final result.

Interpolation of the spherical light projection HVLs do not benefit from the interpolation and gradient computation of spherical lights proposed in Section 3.3. Indeed, the SH coefficients representing the radiance field produced by spherical lights have large amplitude variations when the light sources are close to the projection points, *i.e.* when the spherical lights are included in the interpolation grid, which breaks the interpolation when a sparse grid is used. By construction, the HVL are at the surface of the scene, therefore in the interpolation grid. A careful implementation must therefore be implemented to take advantage of the acceleration of the interpolation.

5.3 Point-based precomputed radiance transfer

This section presents preliminary results where the idea is to efficiently project any function onto the SH using a point-based approximation. Thus, in a first step, we quickly study the differences in convergence that exist between the two SH projection methods (Section 5.3.1): *the scalar product* (Section 2.2.3) and *the least squares* (Section 2.2.4). Then in a second step, we discuss the application of a point based SH projection for probes (Section 2.3.6) and image based lighting to propose a VPL based SH projection (Section 5.3.2).

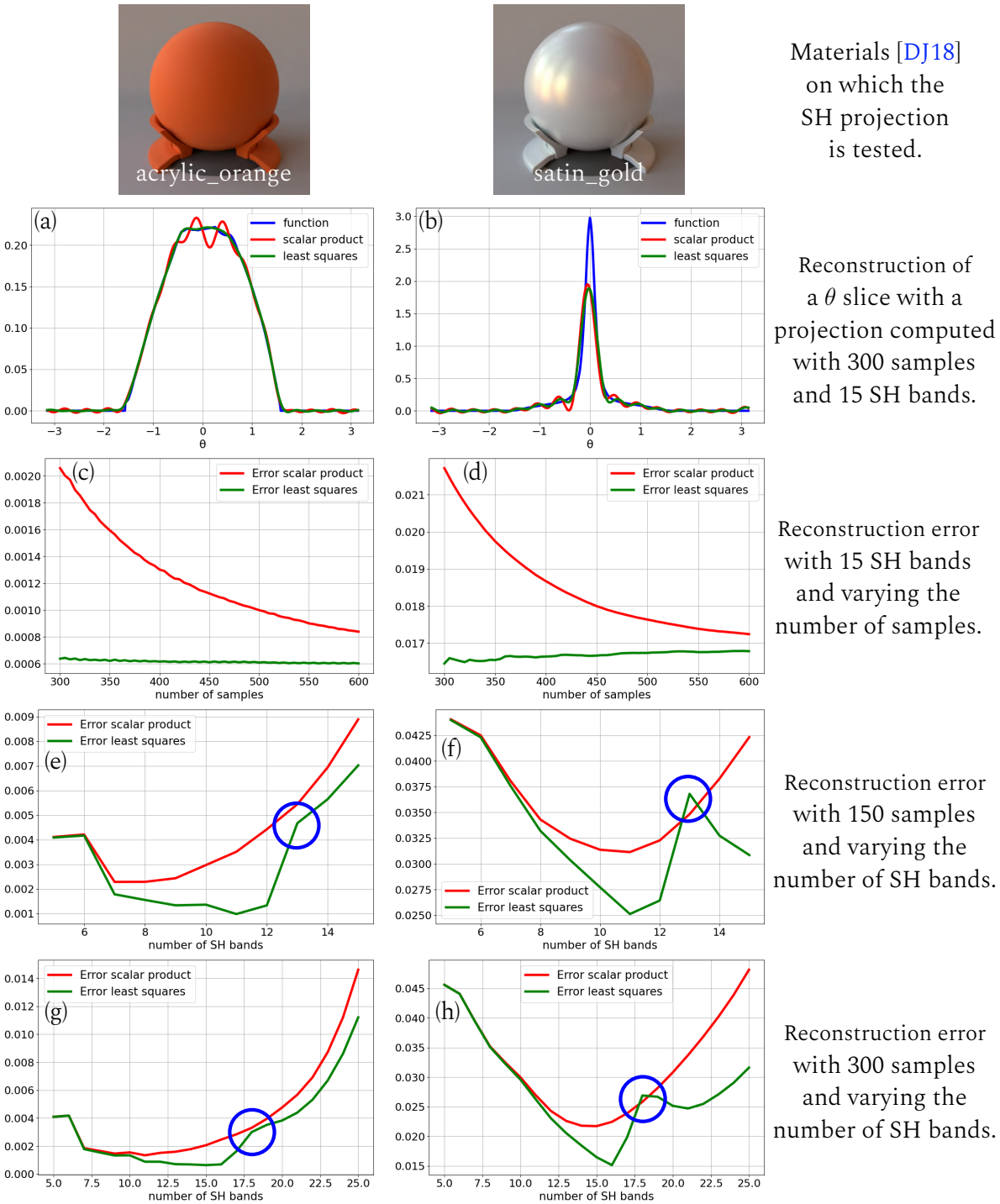
5.3.1 Efficient spherical harmonics projection

We compare the two main SH projection methods, with on the one hand *the scalar product* based on the resolution of an integral (Section 2.2.3) and on the other side, the resolution with *the least squares* (Section 2.2.4). Using *the scalar product*, the function can be projected by solving analytically the integral, as done in this thesis for spherical lights (Chapter 3). The integral can also be solved numerically with Monte-Carlo methods (Section 2.1.4). By construction, the least squares are a numerical method and thus similar to Monte-Carlo methods. The matrices constructed to compute the SH projection for both methods are very similar and clearly outlined in Section 2.2.3 and 2.2.4. In this section, we compare the convergence of the two methods.

The scalar product versus the least squares Let's call N the number of SH coefficients (number of unknowns) we are looking for and M the number of samples we use to resolve the integral. For both methods, the samples are placed at exactly the same locations according to the *spherical Fibonacci sampling* [KISS15]. In a first step, we have taken two functions with different frequencies (Figure 5.16). We observed significant differences in convergence on a range of 100 to 600 samples and up to about 20 SH bands. On these ranges, the SH projection with the scalar product converges slowly to the final value of the SH coefficients where the least squares converges very quickly (Figure 5.16c and 5.16d). In addition, we can note that the projection tends to create oscillations in the reconstructed signal as long as it has not converged (Figure 5.16a). In the higher frequency function, there are still oscillations which are not related to the convergence of the coefficients but related to the traditional ringing artifact of the SH (Section 2.2.5) indicating that 15 SH bands are not sufficient to capture the signal (Figure 5.16b).

By varying the number of bands, we can see that with a small number of bands the two methods perform in the same way. When the number of bands increases, the least squares tend to create less error in the reconstruction of the signal. We observe 3 configurations in the reconstruction error (Figure 5.16e, 5.16f, 5.16g and 5.16h):

- *Overdetermined system* ($N < M$, before the blue circle). When there are more samples than coefficients to compute. The least squares tend to converge faster than the scalar product. This is even more true when the number of coefficients to compute become closer to the number of samples.
- *Underdetermined system* ($N > M$, after the blue circle). When there are more coefficients to compute than samples. Least squares tend to produce a smaller error than the scalar product. However, the error of both methods becomes large because of the frequential nature of the SH. This means that when the problem is overdetermined, there are not enough samples to capture the variations of the SH basis functions resulting in a large error.



Materials [DJ18] on which the SH projection is tested.

Reconstruction of a θ slice with a projection computed with 300 samples and 15 SH bands.

Reconstruction error with 15 SH bands and varying the number of samples.

Reconstruction error with 150 samples and varying the number of SH bands.

Reconstruction error with 300 samples and varying the number of SH bands.

Figure 5.16: *Comparison of the convergence of the SH projection between the quadrature scalar product resolution (red) and the least squares method (green). The comparison is computed on a low frequency function (left) and a higher frequency function (right) and the samples for both methods are positioned at exactly the same locations following the spherical Fibonacci sampling [KISS15]. On these sample ranges (between 300 and 600) the least squares method allows a faster convergence but the method suffers from a stability problem when the method tends towards a determined system. Indeed, when the number of samples approaches the number of coefficients, the conditioning of the matrix becomes huge (Figure 5.18) and increases of the error (blue circle). To minimize the problem, we inverted the matrix using a truncated SVD.*

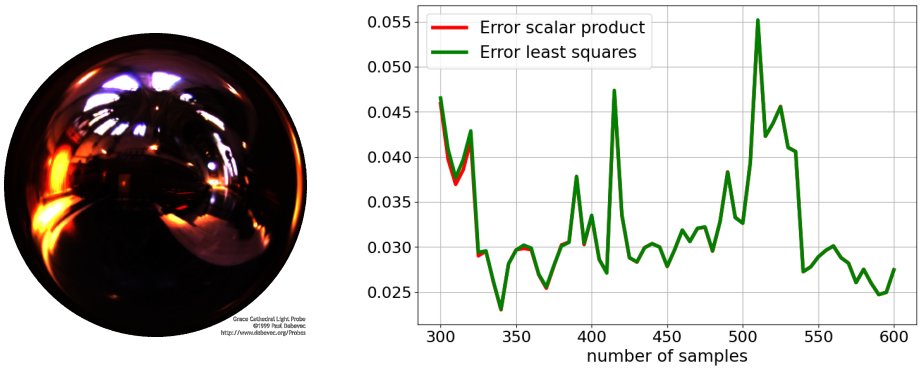


Figure 5.17: *On the comparison scale used (300-600 samples and 15 SH bands) both projection methods perform similarly on a very high frequency signal. The environment probe (left) is projected on 15 SH bands and with different number of samples (right).*

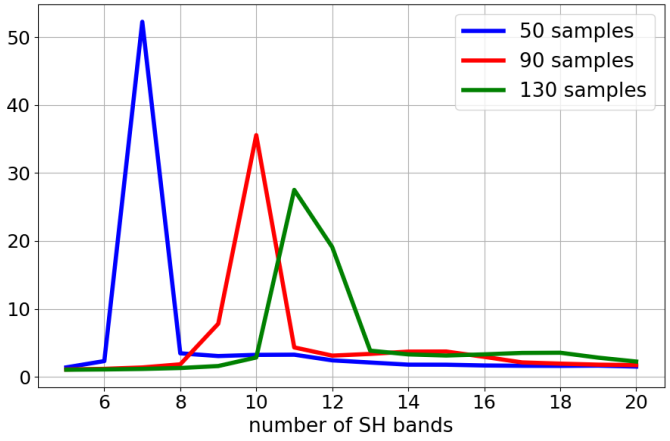


Figure 5.18: *Conditioning of the least squares matrix with different number of samples and SH bands. The conditioning becomes huge when the number of SH coefficients tends towards the number of samples. (The number of SH coefficients is equal to the square of the number of bands.)*

- *Determined system* ($M \approx N$, blue circle). When there are as many coefficients to compute as there are samples, there is no particular problem to solve the scalar product. However, the matrix to be inverted in the least squares method suffers from a conditioning problem (Figure 5.18). The conditioning increases when M tends to N and is maximal when the problem is exactly determined ($M = N$).

However, these configurations (and thus the conditioning problem) only have an impact on the evaluation for low/medium frequency functions. On the sample range considered in this study and on higher frequency signals (Figure 5.17), it is mandatory to use an overdetermined system in order to ensure the convergence. Indeed, the variations of the error of the projection indicate that the projection requires much more than 600 samples before converging, which is much higher than the 225 SH coefficients used. We also observe that on these ranges, the projection error of the scalar product and the least squares have a very similar behavior. On this example, we observed similar behaviors for both methods up to at least 50 bands. Moreover, in computer graphics, the rendering methods rarely use more than 20 SH bands because the number of coefficients is too large for the computations to be efficient. This means that for traditional rendering, the two methods will perform in

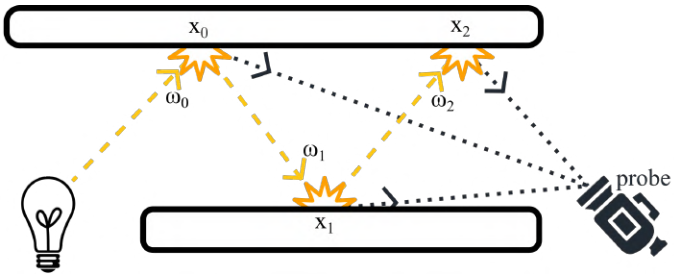


Figure 5.19: *A VPL is distributed at each bounce of the paths launched from the primary light. The probes gather the contribution of each VPL.*

the same way. Nevertheless, if the signal to be projected is band limited, the least squares may compute more efficiently the SH projection but the link between the band limit of the signal and efficiency gain has to be established precisely and is left as future work.

We have not mentioned the differences in the execution time of the two methods because we have experimented in a framework where the projection matrices of the two methods are precomputed and thus giving the same execution time as explained in Section 2.2.4.

5.3.2 SH projection of the radiance field with VPL

Probe-based global illumination is a well-suited framework to apply the SH projection by numerical resolution (Section 2.3.6) . Notably, because the global lighting acts as a low pass filter, which is already interesting for the SH. This property is also interesting for the least squares SH projection because of the assumptions made in the previous subsection. Hence, methods projecting the radiance coming at each probe can benefit from the advantages of the least squares convergences [MK20, IKA+21]. However, as detailed in Section 5.3.3, the least squares are inappropriate to compute the SH projection of the radiance field using VPL. Hence, we propose a simple way to greatly accelerate the SH projection of the probes using VPL. In order to compute the radiance arriving at each probe, luminous paths must be launched from each probe in all directions. This quickly becomes very expensive, so to reduce the computation cost of each probe, we experimented with a projection using VPL (Figure 5.19), which the SH projection of the radiance field is straightforward.

The HVL (Section 5.2) and the probe based approach proposed here are similar, both seek to compute the SH projection of the radiance field. Nevertheless, the HVL approach is specially designed to generate artifact-free one bounce indirect lighting in real-time with few virtual sources and without precomputation. Here, the probe approach proposes to precompute the indirect radiance field generated by VPLs distributed according to any number of luminous bounces, where the number of VPLs and the number of bounces are chosen appropriately to the scene used.

SH projection of probes

We seek to project on SH the radiance field arriving at each probe. The work presented below is thus specialized for the use of probes but can be generalized to any shaded point.

Traditional approach Each probe is traditionally computed by path-tracing with an environment camera. The resulting image is then projected on SH by scanning all the pixels weighted by their solid angle [RH01]. It is an easy solution to implement from an existing render engine but that suffer from degeneration depending on the parameterization of

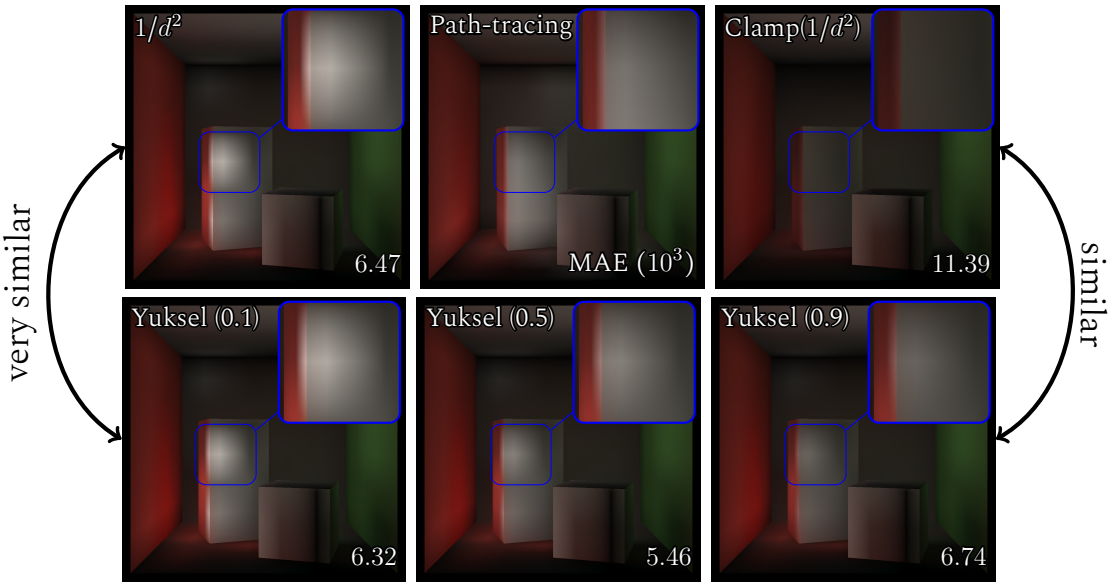


Figure 5.20: *Indirect illumination obtained with the probes using different attenuation model for the VPL. Yuksel's attenuation must set a radius for the VPLs, instead of choosing a radius for each VPL independently according to the density of the virtual sources, we obtain good results by setting a radius for all the VPLs according to the positioning of the probes. A parameter of 0 indicates that the VPLs have a radius of 0 and a parameter of 1 gives a radius equal to the minimum distance that exists between the probes (in a regular probe grid). A parameter of 0.5 yields a good result.*

the environment camera, usually at the poles. A more efficient method is to compute directly the SH projection by launching paths on the desired directions, using, for instance, uniform or quasi-uniform distribution:

$$\mathbf{S}_l^m = \int_\Omega S(\omega)\, Y_l^m(\omega)\, \mathrm{d}\omega \approx \frac{4\pi}{N} \sum_i^N S(\omega_i) Y_l^m(\omega_i) \tag{5.15}$$

where $S(\omega_i)$ is the result of the path-tracing converged in direction ω_i. Hence, the SH projection can be computed by the resolution of the scalar product or by the least squares.

VPL based approach The radiance field emitted by a VPL and received by a shaded point is a Dirac (Section 3.1.1). Thus, the projection of the radiance field at the shaded point is directly the evaluation of the SH basis functions in the direction of the Dirac multiplied by the radiance emitted by the VPL in the direction of the shaded point:

$$\mathbf{S}_l^m \approx \frac{1}{M} \sum_i^M L_i(\mathbf{y}_i \to \mathbf{p}) Y_l^m(\omega_i) \tag{5.16}$$

where p is the probe position, $\mathbf{y}_i$ is the position of the i-th VPL, $\omega_i = (\mathbf{y}_i - \mathbf{p}) / \|\mathbf{y}_i - \mathbf{p}\|$ and L_i also gives the visibility and geometric term between the probe and the VPL (Equation 5.1). This means that instead of computing expensive path-tracing at each probe, VPLs are distributed once to compute the SH projection for all the probes (or shading points).

However, VPL produce the *spikes* and *splotches* artifacts identified earlier (Section 5.1.4). The HVL model avoids them by combining SH and spherical lights (Section 5.2). These artifacts are smoothed out when few SH bands are used, since SH acts as a low pass filter, and the spherical lights allow a better coverage of the indirect radiance emitted by the scene, thus avoiding these artifacts (Section 5.1.4). The SH projection of the radiance field

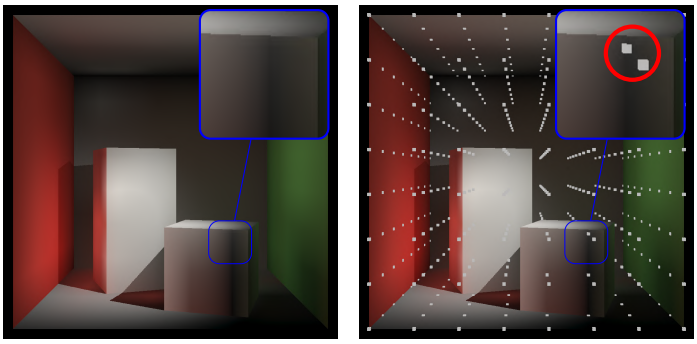


Figure 5.21: *Typical artifacts related to probe positioning. They are due to probes inside objects that do not capture illumination but are still interpolated to obtain the final shading. Thus, a more cautious interpolation avoids this problem but this thesis is interested in computing the probes efficiently and does not seek to solve this problem.*

produced by VPLs does not escape these artifacts. However, the use of probes apply a spatial filter when the probes are distributed parsimoniously in the volume of the scene. Thus, we did not find any really annoying spikes artifacts in the results. In addition, we use a number of VPLs appropriate to the underlying scene, which naturally reduces these artifacts. Nevertheless, splotches still appeared. Their appearance is related to the division by the square distance present in the geometric term. For simplicity, we call this division the attenuation term:

$$A(\mathbf{x},\mathbf{y}) = \frac{1}{\|\mathbf{y}-\mathbf{x}\|^2}\,. \tag{5.17}$$

Instead, we use an attenuation model based on spherical lights proposed by Yuksel [Yuk20]

$$A_{Yuksel}(\mathbf{x},\mathbf{y}) = \frac{2}{r^2}\left(1 - \frac{\|\mathbf{y}-\mathbf{x}\|}{\sqrt{\|\mathbf{y}-\mathbf{x}\|^2 + r^2}}\right) \tag{5.18}$$

where r is a radius affected to the VPL. The model of the *Virtual Spherical Light* [HKWB09] needs to compute a radius per virtual light according to their density. This step can be costly because it requires access to nearby virtual lights. We observed that in order to avoid *splotches* artifacts with the probes, it was not necessary to compute the density of the virtual lights but choosing a radius according to the density of the probes was sufficient. We obtained good results by choosing a radius equal to the minimum distance between two probes divided by two (Figure 5.20). This metric is easy to compute on a regular grid but simple equivalent metrics could be computed to adapt the VPL radius according to the grid model.

Results

We implemented our prototype using an Intel Xeon 2.10 GHz, 32 threads processor and a RTX 3080 graphics card. We implemented the computation of the probes as a precomputation step in pbrt-v3 [PJH16]. The probes are either computed by path-tracing or by VPL, both methods compute only the indirect lighting. Using path-tracing, the paths are independent for each probe and start at the position of each probe distributed according to the *spherical Fibonacci mapping* [KISS15]. The VPL are distributed using P path launched from the primary lights with a depth of D (number of bounces for each path). Hence, the VPL are distributed at each bounces of the paths yielding $P \times D$ maximum VPL in the scene. Each VPL is taken into account for the SH projection of the radiance field for each probe

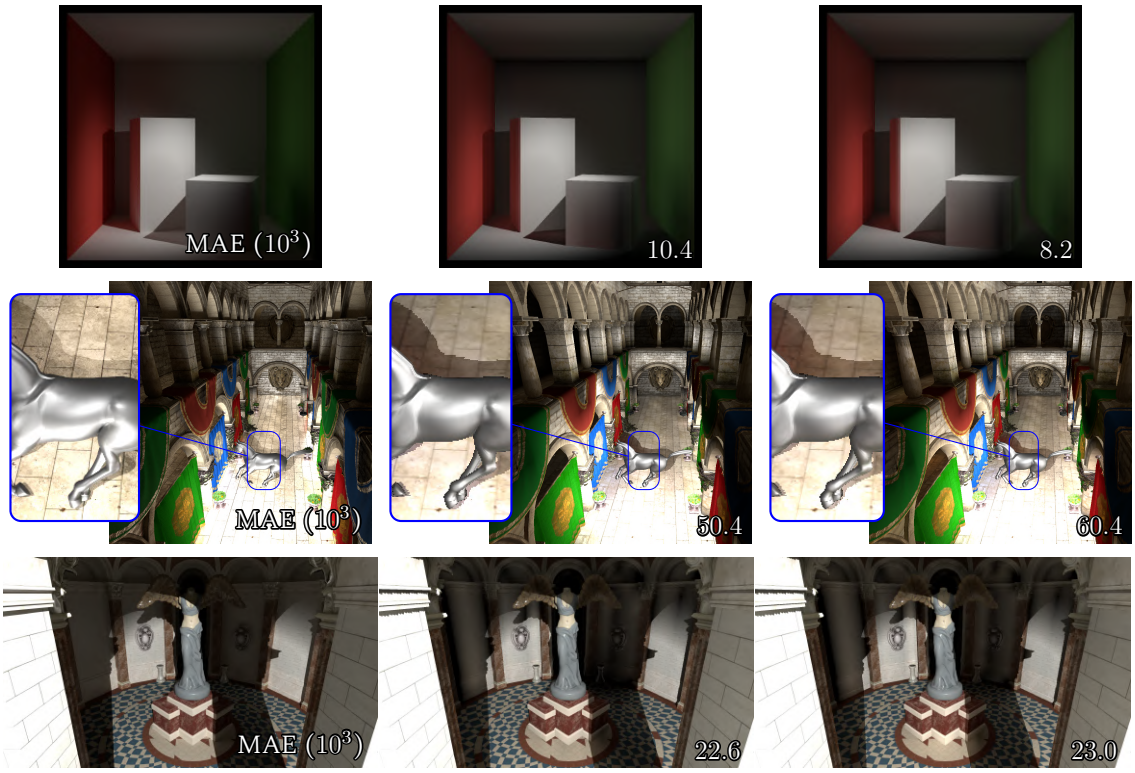


Figure 5.22: *Indirect lighting computed with probes. The computation of the probes with VPLs produces a final lighting very close to the one computed with path-traced probes at a much lower cost (Table 5.4). Probes have difficulty capturing high frequency illumination if their density is not sufficient (horse inset).*

and visibility is computed by ray casting. For both methods, the SH basis functions are evaluated using the efficient Sloan's method (Section 3.2.1).

The interpolation of the probes is implemented in a prototype real-time engine using OpenGL / GLSL 4.5. The core of the method is to speed up the computation of the SH probes, so the probe interpolation implemented here is a simple trilinear interpolation that can produce artifacts related to the positioning of the probes, *e.g.* if a probe is inside an object and therefore does not receive light (Figure 5.21). Methods taking into account these problems are orthogonal to the method proposed here [SL17]. The final shading is computed with the new framework proposed in Section 4.1. It would be interesting to take into account the radiance gradient in order to improve its interpolation. However, its computation is not straightforward (Section 5.3.3).

The results were generated on 5 SH bands. We obtain a similar lighting quality between the path-traced and VPL-computed probes, while achieving an indirect lighting quality close to the result path-traced for each pixel (Figure 5.22 and 5.23). The low pass filter induced by the use of SH allows reducing the number of VPLs needed while keeping an acceptable quality. These first results show that VPLs allow to obtain a quality of projection of the probes at a lower cost than the traditional path-tracing (Table 5.4), in particular when many probes are involved. Indeed, the probes computed by path-tracing use independent light paths while the VPL are mutualized for all the probes. Moreover, all the VPLs are taken into account independently in the computation of the SH projection. Thus, all the methods that accelerate the gathering of the VPLs can be directly used in this application

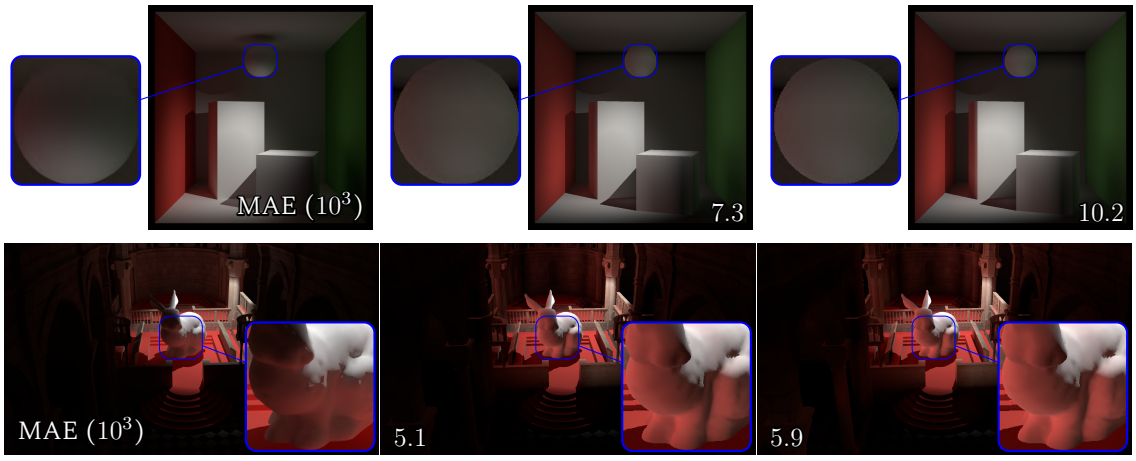


Figure 5.23: *Relighting result for dynamic objects. The sphere and the bunny are not present in the scene during the computation of the probes (path-traced or with VPL). However, they are present for the path-traced result at each pixel (left). Thus, the probes can simulate the lighting on dynamic objects with a coherent lighting quality sufficient for many applications.*

(Section 5.1.3). An in-depth analysis is left as future work, such as an analysis of the accuracy and time according to the number of SH bands and number of VPL.

5.3.3 Discussions and future work

This section presents two different experimental works. The observations on the convergence of the SH projection using the scalar product or the least squares (Section 5.3.1) and the SH projection of the radiance field produced by VPLs (Section 5.3.2). These works can be greatly improved and raise interesting research avenues that we expose here.

Least squares projection (Section 5.3.1) We have observed that the least squares can be particularly interesting when the number of SH coefficients is smaller than the number of samples and when it tends towards it, but also according to the frequencies that compose the signal to be projected. Several methods [HRKM03, Sch13, MW11] compute exactly the scalar product with a minimal number of samples on condition that the frequencies of the signal respect certain conditions. Thus, similar research applied to the least squares method is left as future work and seems to be an interesting opening of our preliminary works.

The initial aim idea for the SH projection of probes using VPL was to take benefit of the advantages of the least squares method. Indeed, in the context of SH rendering, it has been shown that this method has good noise resistance properties as well as robustness with sparse data [SHHS03, LLW06, VSG+13], properties that seem to fit perfectly with VPL rendering. Defining the radiance function received in all directions (by a point) as the radiance surface, the least squares must sample this surface to compute its SH projection. However, the VPLs do not sample this surface but reconstruct it by summing the virtual lights, this is due to the fact that VPL are a subsampling of the light paths. We believe that virtual light approaches that accurately sample the surface of the radiance field at any point of the scene could take advantage of the benefits of the least squares projection exposed in Section 5.3.1. Hence, approaches such as Rich-VPL [SHD15] could be used for this purpose.

Scene*	#Probes	Path-traced probes			VPL probes					real-time indirect pass (ms)
		#paths	depth	time (s)	VPL distribution			SH proj. (s)	total time (s)	
					#paths	#VPL	time (s)			
Cornell	512	10000	8	**7.8**	2000	7894	0.2	0.2	**0.4**	2.3
Sun-temple	27000	6000	5	**942.6**	10000	27794	0.2	67.5	**67.7**	6.7
Sibenik	14400	10000	5	**371.5**	10000	49668	0.2	48.5	**48.7**	5.4
Sponza	8192	7000	4	**467.6**	30000	108177	0.8	90.5	**91.3**	4.3

*Cornell (Fig. 5.22 and 5.23 - top) , Sun-temple (Fig. 5.22 - bottom)
Sibenik (Fig. 5.23 - bottom) , Sponza (Fig. 5.22 - middle)

Table 5.4: *Details of the parameters for the computations of the probes. The number of paths corresponds to the number of paths launched from each probe, this corresponds to the number of directions to evaluate the SH projection integral. Each direction relaunch the path 8 times and with a certain depth. This depth is the same for the VPL distribution paths. Thus, the VPL uses a number of paths starting from each light source and a VPL is positioned at each bounce on a surface. Their distribution is extremely fast and their gathering to compute the SH projection is slower but still faster than the path-tracing method for each probe. However, the gathering implemented here is very naive, it consists of a for loop on each VPL, so the many gathering improvements proposed in the state of the art can be used to strongly reduce this time (Section 5.1).*

Perspectives on the VPL approach (Section 5.3.2)

Interactivity Interactivity means the refinement of the shading computation as well as its display at different convergence steps. The refinement of the lighting is more straightforward with VPL than with path-tracing. It is common to see interactive rendering of path-tracing slowly converging for a single image, but not suitable to compute the SH projection of several thousand probes at different convergence steps of the path-tracer. Indeed, in order to propose an interactive approach with path-traced probes, the SH projection (Equation 5.15) needs to be recomputed at each convergence steps which means that all radiance from all directions ω_i for each probe must be stored in memory. For instance, using a grid of 30^3 probes with 10000 directions for each probe requires 3.24GB. It should be noted that such a grid is not very dense for a large scene and this number of directions corresponds to approximately only an image of 100×100. This implies that the memory requirements in typical use cases will be much higher. Our method only necessitates to add the contribution of VPL multiplied by the SH evaluation. Thus, no complete recomputation of the SH projection is required and no significant storage is needed, since it is only necessary to store the VPLs for all the probes.

Future applications All systems that need an efficient or interactive SH projection of the incident radiance field can benefit from our proposal. For example, NeRF systems that project the incident radiance and whose cost is reflected in the tens of hours of training required of the neural network [YLT+21, STC+22]. In the context of differentiable rendering, recent work differentiates the SH projection of the direct radiance field produced by polygonal lights [WCZR20]. We have proposed similar work but adapted to spherical lights (Section 3.3). Generalizing these works for the full radiance fields, direct and indirect using a point-based model, seems to be a promising and exciting avenue of research.

Radiance gradient Before discussing the differentiation of the radiance field generated by VPL let's discuss the one produced by a point light. The SH projection of the radiance

emitted by such light source is simply an evaluation of the SH basis function in the direction of the point light scaled by the intensity of the light, that can be also modulated by an attenuation factor:

$$C_l^m = A(\mathbf{x}, \mathbf{y}) Y_l^m(\omega) I \tag{5.19}$$

where $\mathbf{x}$ is the shaded point, $\mathbf{y}$ is the point light position, $\omega = (\mathbf{y} - \mathbf{x}) / \|\mathbf{y} - \mathbf{x}\|$ and I is the intensity of the light. Hence, using differentiation rules, the spatial gradient of the SH coefficients is given by

$$\nabla_{\mathbf{x}} C_l^m = \Big(\nabla_{\mathbf{x}} A(\mathbf{x}, \mathbf{y}) Y_l^m(\omega) + A(\mathbf{x}, \mathbf{y}) \nabla_{\mathbf{x}} Y_l^m(\omega) \Big) I \, . \tag{5.20}$$

An efficient method to compute the gradient of the SH basis function $\nabla_{\mathbf{x}} Y_l^m(\omega)$ is given in Section 3.2.2. The spatial gradient of the attenuation factor is straightforward to compute, using the inverse of the squared distance (Equation 5.17) or the Yuksel method (Equation 5.18) [Yuk20], it is given by

$$\nabla_{\mathbf{x}} A(\mathbf{x}, \mathbf{y}) = -\frac{2(\mathbf{y} - \mathbf{x})}{\|\mathbf{y} - \mathbf{x}\|^4} \tag{5.21}$$

$$\nabla_{\mathbf{x}} A_{Yuksel}(\mathbf{x}, \mathbf{y}) = -\frac{2(\mathbf{y} - \mathbf{x})}{\|\mathbf{y} - \mathbf{x}\| \left(\|\mathbf{y} - \mathbf{x}\|^2 + r^2 \right)^{3/2}} \, . \tag{5.22}$$

While the intensity I is considered as constant for point light, it's not longer the case for VPL and hence needs to be also differentiate, involving the complex differentiation of materials as well as visibility (Equation 5.1). Many methods address this kind of problem in differentiable rendering approaches (Section 2.3.8). An additional difficulty is that SH handle a binary visibility, and are not suited to handle full visibility with depth information (Section 2.3.3). This makes SH particularly difficult to handle complex visibility or reflections on fully dynamic scenes, especially when lights are inside the volume of the scene. This is typically the case with VPLs, since by construction they are placed directly on the surface of the scene. To the best of our knowledge, we don't know method that use VPL to differentiate the radiance field but it seems an interesting avenue of research to improve the efficiency of the existing methods.

5.4 Conclusion

In this chapter, we have proposed to combine the spherical harmonics with the point-based global illumination approaches. The first motivation was to propose an efficient model to avoid the traditional artifacts of *Virtual Point Lights* (VPL) that appear due to their singularity (Section 5.1.4). The *Virtual Spherical Light* (VSL) model minimizes these artifacts by using spherical lights. However, the illumination produced by these virtual lights is integrated by Monte-Carlo, thus generating noise. We proposed a new model of virtual lights called *Harmonic Virtual Lights* (HVL) which computes the global illumination by taking advantage of the efficient SH projection of the radiance field produced by spherical lights proposed in Chapter 3. Hence, HVL produce an analytical, and noise-free, approximation of VSL lighting.

The point based lighting methods allow to quickly approximate the integral of the rendering equation. Thus, the SH projection, based on the resolution of an integral, can be computed efficiently. As shown in Section 2.2, the SH projection can be computed by solving the integral or by a least square system. Thus, in a point-based lighting context, we studied the differences in convergence between the two projection methods. However, if

the VPL methods fit well with the resolution of the integral they do not fit directly with the least squares method. We have designed a prototype to efficiently compute the SH projection of the radiance field produced by VPL compared to the traditional path-tracing method. However, a thorough comparison between the two approaches needs to be done and is left as future work.

6 Conclusion

6.1 Summary of the contributions

We have presented several contributions to improve the SH projection of the local and global radiance field as well as to efficiently compute the final lighting at each pixel.

Spherical Light In Chapter 3, we proposed an efficient method to project on the SH, at any point of the scene, the radiance field produced by spherical lights. Thus, several tens of spherical lights can be used in a real time context where the projection is computed at each pixel. We then extended this work to compute the gradient of the spherical harmonic coefficients always representing the radiance field produced by spherical lights. The application demonstrated in this thesis computes the projection and the gradient of the SH coefficients only on the vertices of a 3D volumetric grid in which the 3D scene is immersed. The SH coefficients are then interpolated for each shaded point of the scene. The interpolation based on the gradient allows an accurate approximation of the SH coefficients while using hundreds of spherical light in real time. The gradient of the SH basis functions is used in the computation of the gradient of the SH coefficients. We have extended Sloan's work of efficient evaluation of SH basis functions to compute their gradient efficiently.

The proposed work on spherical lights is inspired by recent work for polygonal lights [WR18, WCZR20]. Although similar in the final application, the methods of the SH projection for the two types of lights are greatly different and notably more expensive in number of operation for the polygonal lights. So, in order to take advantage of the efficiency of the methods, we suggested in Chapter 4 a method to approximate the SH coefficients representing the radiance field produced by polygonal lights with spherical lights. The shading obtained on some SH bands (5-10) with our approximation is particularly close to the one obtained with polygonal lights.

Spherical Harmonics framework with decomposable BRDF The use of dynamic scene with SH rendering methods needs in a first step an efficient computation of the SH coefficients of the radiance field at each shaded points. This is possible with the proposed methods for spherical lights. In a second step, the light and the material must be convolved efficiently. We have proposed in Chapter 4 a new framework to efficiently compute this convolution with decomposable materials (whose diffuse and specular contribution are computed separately). Thus, this new framework allows real-time performances, with demonstrated performances beyond 60 frames per second and is also easy to implement.

Harmonics Virtual Light *Virtual point lights* are commonly used to generate global lighting efficiently. However, their singularity produces well known artifacts and several solutions have been proposed to minimize them (Section 5.1.4). Thus, we have proposed a new type of virtual lights, the *harmonic virtual light* (Chapter 5) which are based on the use of spherical lights and thus on the efficient method of SH projection of the radiance field they produce proposed in Chapter 3. This new type of virtual lights corresponds to an extension and approximation of the *virtual spherical light*, which avoids the singularity of point lights. However, their integration is computed by Monte-Carlo thus generating the noise associated with these methods where our method computes analytically the illumination, thus avoiding this kind of noise.

Point-based Precomputed Radiance Transfer Finally, in Chapter 5 we have proposed experimental works to combine *precomputed radiance transfer* methods with point-based lighting. There are two main methods to compute the SH projection of an arbitrary function (Section 2.2); the resolution of the scalar product corresponding to a computation of an integral, and a resolution with the least squares method. Thus, in a point-based context, we have produced a comparative study of the convergence of these two methods. Then, we used the *virtual point lights* to project on SH the radiance field efficiently at any point. We demonstrated that this method worked better than the traditional projection methods based on path-tracing. However, a thorough comparison needs to be made and is left as future work.

6.2 Perspective and future work

The work presented in this thesis can be improved by several improvement levers. Here are two possible improvements that we think are particularly interesting for future development.

Hemispherical Harmonics The work proposed in this thesis focuses on the use of spherical harmonics to manage lighting in a generic way. Even if some cases consider the lighting only on a hemisphere, algorithms consider the radiance that can come from all directions, *e.g.* with the probes distributed in the volume of the scene. However, considering the lighting on an opaque surface only requires an efficient characterization of the radiance received on the hemisphere above the surface. In this case, hemispherical harmonics are particularly useful (Section 2.2.2). It may therefore be particularly interesting to convert the work proposed in this thesis to the hemispherical harmonics.

Efficient Spherical Harmonics projection The proposed work on the efficient projection on spherical harmonics is experimental (Section 5.3). The comparison between the two projection methods (scalar product and least square) highlights the advantages of the least square method allowing a faster convergence of the SH coefficients. This observation can be exploited for future applications. Also, we have shown that projecting the radiance field using virtual sources on SH was efficient compared to the traditional path-tracing approach but requires an in-depth comparison. Indeed, it is necessary to quantify more precisely the error in the SH coefficients that appears with the use of VPLs, for instance by generating comparisons with equal time or number of samples compared to path-tracing.

Appendix

A Résumé de la thèse

A.1 Introduction et contexte

L'efficacité est une exigence pour de nombreuses applications en informatique graphique. L'efficacité du rendu se décline en 3 catégories : temps réel, temps interactif et hors ligne. Le temps réel correspond à un temps de calcul de l'ordre de quelques millisecondes, permettant ainsi une fréquence d'affichage rapide, telle que requise dans les jeux vidéo visant généralement plus de 60 *Images Par Secondes* (IPS). Le temps d'interaction est de l'ordre de moins de quelques secondes permettant une qualité de rendu plus fine tandis que la fréquence de rafraîchissement est suffisante pour interagir efficacement, la *Conception Assistée par Ordinateur* (CAO) vise généralement cette catégorie. Au-delà, on parle de rendu hors ligne, permettant une qualité maximale comme celle souhaitée pour les images finales d'un film. Un rendu efficace est synonyme de performance, en termes de nombre d'opérations mais aussi en termes d'organisation de ces opérations pour optimiser leur exécution en fonction du matériel utilisé. Le besoin d'efficacité est toujours plus grand en raison, notamment, de l'utilisation de scènes 3D de plus en plus complexes, de par leur géométrie mais aussi de par la complexité des transports de lumière présents dans la scène. La clé de l'efficacité du rendu est une caractérisation efficace du transport de la lumière dans une scène 3D. C'est la variété des configurations possibles de la scène qui rend ce problème particulièrement complexe (Figure A.1).

Figure A.1: *Les besoins de simulation de l'éclairage sont très divers, allant par exemple du rendu de grands paysages lumineux comme dans Red Dead Redemption II [Roc18] (gauche) à des milieux urbains de nuit utilisant des centaines de lumières surfaciques comme dans Stray [Blu22] (droite).*

C'est dans ce contexte d'efficacité que les harmoniques sphériques (HS) ont pris tout leur sens en informatique graphique. Les HS sont des fonctions de base orthogonales permettant d'approcher toute fonction sphérique par un vecteur de coefficients où chaque coefficient est attaché à une HS. L'orthogonalité des fonctions de base rend leur utilisation particulièrement intéressante pour le rendu en informatique graphique. Grâce à ces propriétés, les harmoniques sphériques ont été utilisées très tôt dans le domaine de l'informatique graphique [CMS87]. Cependant, leur application la plus notable est venue un peu plus tard avec la méthode du *Précalcul de Transfert de Radiance* [SKS02]. Dans cette méthode, les harmoniques sphériques encodent le transport de la lumière à chaque point de l'environnement 3D.

A.2 Organisation de la thèse et aperçu des contributions

L'une des principales difficultés de l'utilisation des harmoniques sphériques pour les scènes dynamiques est la projection sur les HS à la volée du champ de radiance en tout point de la scène. Le chapitre 3 présente trois contributions distinctes améliorant l'efficacité de la projection sur les harmoniques sphériques pour les lumières sphériques. La première contribution consiste en un calcul analytique et récursif de la projection du champ de radiance produit par une lumière sphérique, basée sur une évaluation efficace des harmoniques sphériques. La deuxième contribution est une dérivation analytique de l'évaluation des fonctions de base des harmoniques sphériques permettant de proposer dans une troisième contribution un calcul analytique et récursif du gradient de la projection du champ de radiance produit par une lumière sphérique. Nous démontrons l'efficacité de cette troisième contribution en calculant la projection sur les HS du champ de radiance, avec le gradient des coefficients d'HS, uniquement sur un sous-ensemble parcimonieux de points distribués volumétriquement dans la scène. Le gradient permet d'interpoler avec précision les coefficients d'HS représentant le champ de radiance à chaque point de la scène. Ces contributions sont motivées par des travaux récents proposant des contributions similaires, mais adaptées à des lumières polygonales : [WR18, WCZR20].

Cependant, l'éclairage que nous percevons en un point n'est pas simplement le champ de radiance que le point reçoit, mais le résultat de la convolution entre le champ de radiance et le matériau sur lequel le point est situé. Nous avons conçu un nouveau cadre de travail pour calculer efficacement la convolution pour les scènes dynamiques et avec l'utilisation de matériaux couramment utilisés en informatique graphique séparant la contribution diffuse et spéculaire (Chapitre 4). De plus, nos méthodes de projection du champ de radiance produit par les lumières sphériques étant plus efficaces que les travaux proposés pour les lumières polygonales, nous proposons une méthode pour approcher la projection du champ de radiance produit par les lumières polygonales en utilisant nos méthodes présentées dans le chapitre précédent.

Enfin, le chapitre 5 se concentre sur l'éclairage global en combinant l'utilisation des harmoniques sphériques avec les méthodes d'éclairage global basé point. Nous proposons un nouveau type de lumière virtuelle, appelé *Harmonics Virtual Light* (HVL), basé sur les méthodes de projection efficaces du champ de radiance produit par les lumières sphériques proposées dans le chapitre 3. Nous montrons également que la combinaison des VPL et des harmoniques sphériques peut être particulièrement bénéfique pour calculer efficacement le champ de radiance global reçu par des sondes distribuées volumétriquement dans la scène.

A.3 Contenu du mémoire

Ce mémoire est divisé en 6 chapitres, dont le premier introduit la thèse et le dernier vient le conclure. Le chapitre 2 vient poser l'état de l'art de l'éclairage utilisant les harmoniques sphériques ainsi que détailler les mathématiques de ces dernières. Les 3 chapitres restants décrivent les contributions proposées. Le chapitre 2 et les 3 suivants représentent les chapitres d'intérêt de ce mémoire que nous venons résumer ici.

A.3.1 Chapitre 2 - Éclairage et harmoniques sphériques

Ce chapitre introduit les bases de la synthèse d'image, les mathématiques des harmoniques sphériques, qui les rendent notamment particulièrement bien adaptées à la synthèse d'image et les applications des HS dans l'histoire de l'informatique graphique. Ce chapitre permet donc de poser les bases nécessaires à la compréhension des contributions présentées dans les chapitres suivants. Ainsi, afin de bien comprendre le résumé des prochains chapitres nous résumons ici les clés de compréhension du chapitre 2.

Synthèse d'image

L'équation du rendu [Kaj86] est un bilan énergétique à la surface d'un objet, elle relie la radiance entrante à la radiance sortante en un point de la scène. L'équation du rendu repose principalement sur deux fonctions, la radiance incidente et une fonction de transfert correspondant aux propriétés du matériau sur lequel est posé le point de la scène. Pour une direction de sortie (direction de vue), l'équation du rendu intègre la radiance incidente multipliée par la fonction de transfert pour l'ensemble des directions incidente. C'est-à-dire l'ensemble des directions correspondant à la sphère unité. Les méthodes Monte-Carlo sont classiquement utilisée pour résoudre numériquement l'équation du rendu en chaque point de la scène. Les bases de fonctions sont un outil mathématique permettant de représenter une fonction par un vecteur de coefficients, où chaque coefficient est rattaché à une seule fonction de base. Les coefficients sont calculés en projetant la fonction sur chacune des bases de fonctions. La projection correspond à un produit scalaire fonctionnel. Les bases de fonctions orthogonales, telles que les harmoniques sphériques, simplifient le calcul d'une intégrale du produit de deux fonctions par le produit scalaire des coefficients des fonctions de base. Ainsi, cela s'applique parfaitement pour résoudre l'équation du rendu efficacement.

Harmoniques sphériques

Les Harmoniques Sphériques (Figure A.2) forment une fonction de base orthogonale où chaque fonction est indexée par un ordre l et un degré m. Les HS sont analogues aux séries de Fourier, décomposition en sinus et cosinus, mais défini sur une sphère. Les HS sont ordonnées de la manière suivante : plus l'ordre l est grand, plus les fréquences angulaires sont élevées. Plus précisément, l'ordre l définit le nombre d'oscillations le long des longitudes et le degré m le long des latitudes. L'ordre l peut aller à l'infini et le degré m varie entre $-l$ et l, formant ainsi une pyramide de fonctions.

La projection, le vecteur de coefficients, peut-être calculé par deux grandes méthodes. D'un côté la projection par résolution du produit scalaire, comme évoqué précédemment, et de l'autre la projection par moindres carrées. Analogiquement aux méthodes des moindres carrées, la résolution du produit scalaire peut-être fait numériquement. Nous démontrons que la méthode des moindres carrées permet de converger plus rapidement selon la

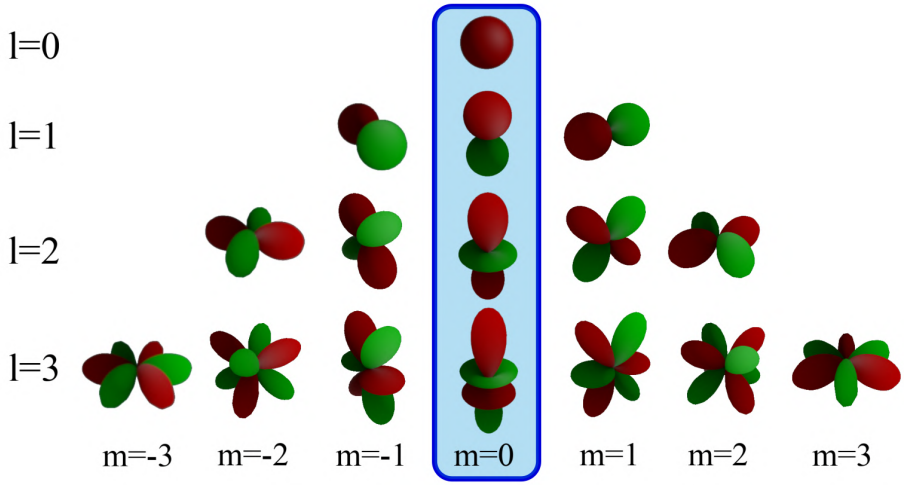


Figure A.2: *Représentations des premières fonctions de base des harmoniques sphériques réelles. Les lobes sont colorés selon leur signe, les valeurs positives en rouge et négative en vert. La sous-base des harmoniques zonales pour lesquelles $m = 0$ est encadrée au milieu.*

nature de la fonction qui est projetée (Section A.3.4). Les harmoniques sphériques agissent comme un filtre passe-bas, où la fréquence de coupure est fixée par l'ordre l utilisé. Ainsi, les représentations fréquentielles coupant brutalement les fréquences sont connues en traitement de signal pour produire des oscillations dans la reconstruction du signal. Des solutions classiquement utilisées en traitement de signal peuvent être mis en place pour atténuer ces oscillations.

Le cas où $m = 0$ est une sous-base des HS, appelé les Harmoniques Zonales (HZ). Cette sous-base est particulièrement utile pour représenter des signaux circulairement symétriques. Ainsi, cela permet de réduire drastiquement le nombre de coefficients à prendre en compte pour ce type de signal. En conséquent, des méthodes cherchent à factoriser des signaux arbitraires par une combinaison d'HZ [NSF12, SLS05]. De plus, les propriétés de convolution et rotation sont beaucoup plus efficaces lorsque le signal est réduit à un vecteur de coefficients sur les HZ, comparé aux mêmes propriétés posées sur les HS.

Application à l'informatique graphique

Les premières utilisations des HS en informatique graphique avaient pour but d'utiliser des cartes d'environnement efficacement [CMS87, RH01]. En effet, une carte d'environnement est valable pour l'ensemble des points de la scène et peut donc être compressée par un vecteur de coefficients. La méthode de *Précalcul de Transfert de Radiance* [SKS02] est la méthode la plus notable d'utilisation des HS. Celle-ci vient projeter sur les HS la fonction caractérisant les transferts de radiance, et ceux en chaque point de la scène. Cela permet de simuler en temps réel l'éclairage global avec des conditions d'éclairage quelconque.

Les techniques de mise en cache de la radiance [KGBP05] et éclairage par sondes [IKA+21] sont des exemples typiques de l'utilisation des HS dans un cadre général. Des techniques cherchent à tirer profit des méthodes de factorisation sur les HZ évoqués précédemment pour proposer des méthodes d'éclairage toujours plus efficace [BXH+18, WR18, SBN15]. De manière générale, les HS peuvent être mis à contribution lorsque des fonctions sphériques doivent être représentées, pour représenter par exemple l'occlusion, ou améliorer les méthodes de rendu différentiable ainsi que les champs de radiance neuronaux (NeRF).

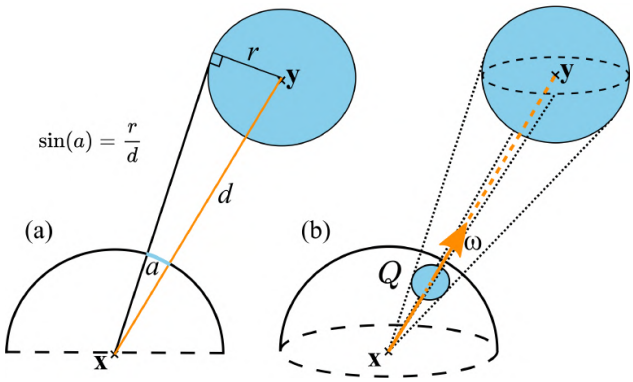


Figure A.3: *Configuration géométrique de la contribution de luminance d'une lumière sphérique. (a) Découpage 2D montrant le rayon r de la lumière sphérique, la distance d avec la lumière sphérique, et le demi-angle a de la calotte. (b) La lumière sphérique en bleu se projette sur une calotte sphérique Q sur la sphère unitaire centrée sur le point d'ombrage* $\mathbf{x}$.

Enfin, il est intéressant de noter que les HS sont surtout utilisés pour encoder l'éclairage localement, en un point de la scène. D'autres fonctions, comme les ondelettes [NRH03] ou les gaussiennes sphériques [WRG+09] sont utilisés dans le même intérêt et disposent de propriétés différentes comparées aux HS. Cependant, des travaux récents changent de paradigme en encodant l'éclairage globalement [RBRD22, BDBS22].

A.3.2 Chapitre 3 - Lumière sphérique

Ce chapitre cherche à calculer efficacement en tout point de la scène les coefficients d'HS représentant le champ de radiance produit par des lumières sphériques. Pour cela, ce chapitre propose une projection efficace du champ de radiance sur les HS en tout point de la scène. Le champ de radiance peut aussi être calculé seulement sur un ensemble parcimonieux de point pour être par la suite interpolé en tout point de la scène. Ce chapitre propose une méthode de calcul efficace du gradient des coefficients d'HS permettant notamment d'améliorer la qualité d'interpolation. De plus, le calcul du gradient des coefficients repose sur le calcul du gradient des fonctions de base des HS aussi proposé dans ce chapitre. Ces contributions sont notamment basées sur des travaux similaires récents, mais pour calculer efficacement les coefficients d'HS représentant le champ de radiance produit par des lumières polygonales [WR18, WCZR20].

Lumière sphérique

Une primitive de lumière simple émet de la lumière telle qu'un point $\mathbf{x}$ la recevant ne la reçoit qu'en une seule direction ω. Dans ce cas, la projection sur les HS du champ de radiance en $\mathbf{x}$ correspond à une simple évaluation des harmoniques sphériques en ω. Dans le cas des sources surfaciques, la lumière est reçue dans un ensemble de directions. Pour les lumières sphériques, cet ensemble de directions correspond à une calotte sphérique (Figure A.3), c'est-à-dire une fonction circulairement symétrique qui est donc représentable uniquement à l'aide des HZ. Ainsi, la contribution est une nouvelle formule permettant de calculer efficacement la projection du champ de radiance sur les HZ. Cependant, la projection est dans un repère orienté selon la direction du centre de la sphère au point $\mathbf{x}$, la projection de plusieurs lumières sphériques ne sont donc pas directement cumulable. Cependant, les HZ disposent d'une propriété de rotation efficace permettant de mettre toutes les projections dans le même repère (Section A.3.1).

Gradient des harmoniques sphériques

La méthode de rotation des HS implique une évaluation des fonctions de base des harmoniques sphériques. Sloan propose une méthode d'évaluation efficace des harmoniques sphériques [Slo13]. Cette méthode repose sur la factorisation des fonctions trigonométriques impliquées dans les HS. Cette factorisation fait disparaître l'évaluation explicite de fonctions trigonométriques remplacée par des relations récursives sur l'ensemble des HS. Nous étendons la méthode de Sloan afin de calculer efficacement le gradient des harmoniques sphériques. Nous identifions une factorisation similaire à celle proposée par Sloan dans le calcul du gradient des HS et nous adaptons les relations récursives pour obtenir le gradient. Cette nouvelle méthode est mise à profit dans le calcul du gradient des coefficients d'HS représentant le champ de radiance produit par des lumières sphériques.

Gradient de la projection des lumières sphériques

Nous cherchons ici à obtenir un calcul efficace du gradient des coefficients d'HS représentant le champ de radiance produit par des lumières sphériques. La projection du champ de radiance repose sur deux évaluations, d'un côté les coefficients d'HZ représentant le champ de radiance dans le repère orienté selon la lumière sphérique et de l'autre côté, l'évaluation des HS. Ces deux étapes nécessitent ainsi d'être dérivées pour pouvoir calculer le gradient des coefficients d'HS. Ainsi, une méthode efficace du calcul du gradient des HS a été proposée dans le précédent paragraphe. Le calcul du gradient des coefficients d'HZ est simplement une dérivation analytique directe. La nature récursive sur l'ensemble des HS de l'algorithme proposé nécessite de garder en mémoire un ensemble important de coefficient et de gradient. Cela impacte grandement les performances dans un contexte d'implémentation parallèle sur carte graphique. Afin de pallier ce problème, nous proposons une réorganisation des calculs permettant de ne pas rompre les relations récursives tout en minimisant le nombre de coefficients et de gradients à stocker en mémoire.

Le gradient des coefficients d'HS peut être utilisé à plusieurs fins. L'application utilisée dans cette thèse et la même que celle proposée par Wu *et al.* [WCZR20]. Les coefficients des HS représentant le champ de radiance produit par des lumières sphériques (et polygonales dans le cas de Wu *et al.*) sont calculés sur une grille volumique qui contient la scène 3D. Le gradient des coefficients est aussi calculé seulement sur cette grille. Les coefficients sont ensuite interpolés pour chaque point de la scène, le gradient permet d'interpoler précisément les coefficients d'HS.

Ainsi, lorsque la projection sur les HS est calculée pour chaque point de la scène, une dizaine de lumières sphérique peut-être utilisé en temps réel. Lorsque l'interpolation est utilisée, plusieurs centaines de lumières sphériques peuvent être utilisées. Cela permet, par exemple, de simuler des lumières texturées en temps réel. Le gradient permet d'obtenir une qualité d'interpolation telle que la différence entre les deux méthodes est invisible à l'œil nu.

A.3.3 Chapitre 4 - Éclairage direct

Ce chapitre est divisé en deux sous-sections distinctes. Dans un premier temps, nous proposons un nouveau cadre de travail permettant de calculer efficacement l'éclairage final avec des matériaux décomposables, c'est-à-dire dont la contribution diffuse et spéculaire sont calculées séparément. Ensuite, il se trouve que les méthodes que nous avons développées pour des lumières sphériques sont plus efficaces que celles proposées pour des lumières polygonales. Ainsi, nous proposons dans un second temps d'utiliser les lumières

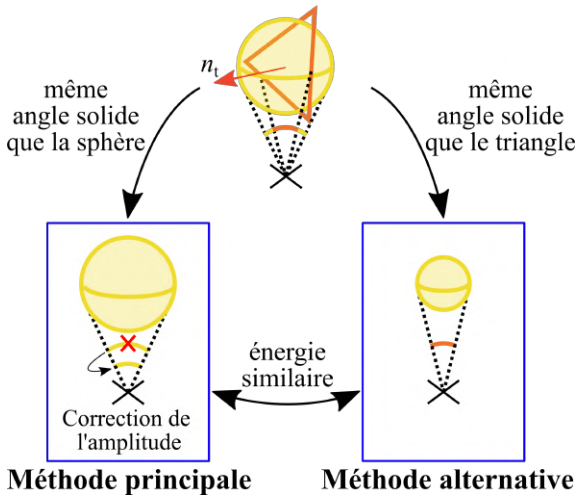


Figure A.4: *Vue d'ensemble des deux méthodes d'approximation, où la méthode principale fixe les paramètres de la sphère (position et rayon) une fois pour toutes, tandis que la méthode alternative modifie le rayon de sorte que l'angle solide de la sphère à chaque point éclairé soit égal à l'angle solide du triangle.*

sphériques afin d'approcher les coefficients d'HS représentant le champ de radiance produit par des lumières polygonales.

Éclairage avec des matériaux décomposable

Un matériau peut être majoritairement diffus, comme une feuille de papier, ou majoritairement spéculaire comme une piste de bowling. Ces deux aspects correspondent à deux façons de réfléchir la lumière, deux lobes de réflexions. Un matériau diffus réémet la lumière reçue dans toutes les directions de manière uniforme tandis qu'un matériau spéculaire réémet la lumière dans un ensemble plus ou moins restreint de directions. La gestion de ces deux lobes est suffisante pour simuler une grande variété de matériaux et la contribution lumineuse de ces deux lobes sont généralement calculées séparément. Cependant, lors du calcul de l'éclairage par le biais des HS, nous avons identifié des redondances de calcul que nous avons factorisé. Cette factorisation permet de calculer la contribution diffuse à partir du calcul de la contribution spéculaire avec une seule multiplication.

De plus, le calcul de la contribution spéculaire implique un produit avec un cosinus clampé. Lors de l'utilisation de matériau capturé projeté à l'avance sur les HS, il est courant d'y inclure le cosinus clampé de l'équation du rendu. Cependant, cela complexifie énormément le stockage des coefficients d'HS. L'alternative est de calculer un produit sur les HS avec le cosinus clampé. Ainsi, nous proposons une méthode efficace de calculer ce produit pouvant être calculé à la volée pour chaque pixel en temps réel.

Approximation des lumières polygonales

Les méthodes de calcul des coefficients d'HS représentant le champ de radiance produit par des lumières sphériques (Section A.3.2) sont plus efficace que les méthodes calculant les coefficients d'HS représentant le champ de radiance produit par de lumières polygonales. Cela s'applique aussi pour le calcul du gradient des coefficients d'HS [WR18, WCZR20]. Ainsi, afin de tirer profit de nos méthodes de calculs moins coûteuses, nous proposons deux méthodes afin d'approcher les coefficients d'HS représentant le champ de radiance

produit par des lumières polygonales (Figure A.4). La méthode se focalise sur les lumières triangulaires, comme n'importe quel polygone peut-être divisé en triangles et une lumière sphérique correspond à une lumière triangulaire. Les deux méthodes positionnent le centre de la sphère au barycentre du triangle. La méthode principale calcule le rayon de la sphère tel que le disque sur le même plan du triangle ait la même aire que celui-ci. L'amplitude de chaque coefficient d'HS est corrigée selon la normale du triangle est la direction du point éclairé. La méthode alternative calcule le rayon de la sphère à chaque point éclairé de manière que l'angle solide de la sphère soit le même que celui du triangle. Ainsi, la méthode principale applique une correction rapide à calculer, mais dépendante du nombre de coefficients à traiter. Tandis que la méthode alternative applique une correction moins rapide à calculer, mais qui ne dépend pas du nombre de coefficients. En conséquence, la méthode principale est plus rapide à calculer lorsque peu de coefficients d'HS sont impliqués et la tendance s'inverse lorsque beaucoup d'HS sont utilisées. Néanmoins, la méthode secondaire permet une meilleure qualité d'approximation des coefficients d'HS.

A.3.4 Chapitre 5 - Éclairage global basé point

Les chapitres précédents s'intéressent à une intégration locale de l'éclairage. En contraste, ce chapitre se focalise sur une génération de l'éclairage global en utilisant notamment un système basé point. L'éclairage global basé point est une méthode communément utilisée pour la synthèse d'image en temps réel où l'environnement 3D est échantillonné sous forme d'un nuage de point capturant l'éclairage incident et le redistribuant dans la scène afin de générer l'éclairage indirect. Les harmoniques sphériques sont combinés dans ce chapitre avec ces modèles d'éclairage basé point pour notamment éviter les écueils de ces derniers.

Traduction française des acronymes anglais

Ce chapitre utilise quelques acronymes en anglais. Nous utilisons dans ce résumé les mêmes acronymes anglais pour simplifier le lien avec le contenu du mémoire. Les acronymes anglais sont traduits dans la table suivante.

Acronyme	Anglais	Français
RSM	Reflective Shadow Map	Carte d'ombre réflective
VPL	Virtual Point Light	Lumière virtuelle ponctuelle
VSL	Virtual Spherical Light	Lumière virtuelle sphérique
VSGL	Virtual Spherical Gaussian Light	Lumière virtuelle sphérique gaussienne
HVL	Harmonics Virtual Light	Lumière virtuelle harmoniques

Radiosité instantanée

L'algorithme de radiosité instantanée repose sur deux étapes, la distribution des lumières virtuelles et la collecte de leur éclairage (Figure A.5). Les lumières virtuelles sont positionnées généralement à chaque rebond d'un chemin lumineux lancé depuis une source de lumière [SIMP06, SIP07]. Ainsi, la lumière qu'elles émettent dépend du chemin lumineux de distribution. Les cartes d'ombres permettent de générer les ombres dans une scène 3D en gardant seulement une information de profondeur depuis la source primaire de lumière. Cette approche est particulièrement intéressante pour le temps réel car hautement parallélisable. Les RSM [DS05] étendent le principe des cartes d'ombres en gardant en plus de

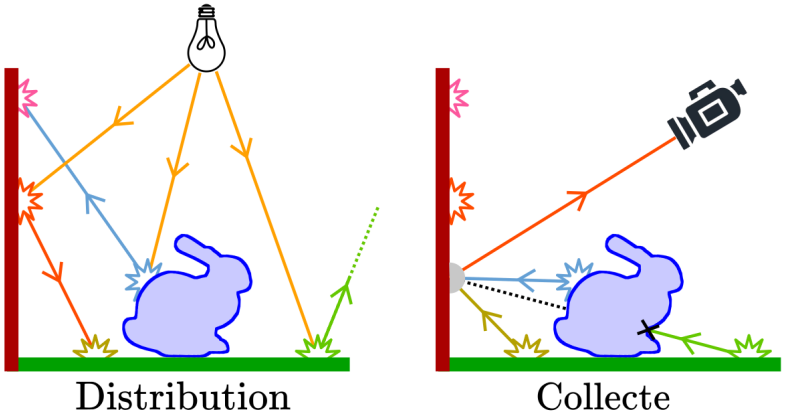


Figure A.5: *L'algorithme de radiosité instantanée [Kel97] est basé sur deux étapes, l'algorithme distribue d'abord un ensemble de VPLs à partir de la source primaire et rassemble ensuite l'éclairage produit par les VPLs.*

l'information de profondeur des informations de réflexions. Ainsi, les RSM sont particulièrement efficaces pour distribuer les lumières virtuelles. La collecte de l'éclairage des lumières virtuelles est délicate à calculer efficacement, en particulier lorsque des milliers, voire millions, de lumières virtuelles sont impliquées. Des approches regroupent les lumières virtuelles similaires en une seule lumière [DGR+09] ou les organisent dans des structures arborescentes [WFA+05] afin de calculer leur éclairage plus facilement. Cependant, cette thèse ne s'intéresse pas à simplifier la collecte de l'éclairage, mais à plutôt corriger des défauts liés à la singularité des VPL.

Les VPL sont des lumières singulières car ponctuelles. Ainsi, l'éclairage est capturé et redistribué en un seul point sur la scène 3D. Cela tend à créer des artefacts lorsque la densité des VPL n'est pas suffisante pour capturer la complexité de l'éclairage. Les artefacts sont liés au vide qui existe entre les lumières virtuelles, les VSL utilisent des lumières sphériques afin d'éviter des zones de vide dans l'éclairage [HKWB09]. Cependant, elles introduisent deux difficultés. Dans un premier temps, il est nécessaire de donner à chaque source un rayon permettant d'éviter les zone de vide sans générer de recouvrement trop important des VSL entre-elles et dans un second temps, l'éclairage généré par les VSL est plus complexe à collecter. En contraste, les VSGL approchent l'éclairage d'un ensemble de VPL avec des gaussiennes sphériques, permettant aussi de combler les zones de vides [Tok15b].

Sources virtuelles basées harmoniques sphériques

Les HVL sont un nouveau modèle de lumière virtuelle proposée dans cette thèse. Elles peuvent être vues comme une simplification des VSL où l'éclairage produit par des lumières sphériques est projeté sur les HS. Ainsi, ces travaux mettent à profit les travaux proposés sur les lumières sphériques (Section A.3.2). Ce qui représente une accélération significative par rapport au VSL où l'intégration de l'éclairage est calculée par les méthodes Monte-Carlo. Les HVL sont initialement conçues pour supporter n'importe quel matériau sans surcoût, les matériaux étant projetés sur les HS dans une étape de précalcul. Afin de les distribuer le plus efficacement possible, nous proposons un modèle de distribution des HVL basé sur les RSM. Nous proposons aussi deux heuristiques de calcul des rayons des lumières sphériques virtuelles qui estiment la densité des lumières virtuelles lorsque celles-ci sont distribuées à l'aide d'une RSM. Ainsi, les VSL peuvent directement bénéficier de ce modèle de distribution. Nous démontrons que les HVL permettent d'obtenir un com-

promis entre qualité et temps de calcul obtenu pour le modèle des VPL et des VSL. Le modèle des VSGL est limité à des matériaux avec un seul lobe spéculaire simple. Les HVL peuvent être optimisées lorsque les matériaux sont limités à un lobe spéculaire circulairement symétriquement. Cela permet d'accélérer grandement l'éclairage en limitant le calcul aux HZ. Ainsi, les HVL deviennent compétitives par rapport aux VSGL en produisant une qualité d'éclairage similaire.

Précalcul de transfert de radiance basé point

Cette section présente des travaux expérimentaux afin d'accélérer les modèles de précalcul de transfert de radiance avec les modèles basé point. Dans un premier temps, nous voulions obtenir un modèle de projection efficace sur les HS d'une fonction arbitraire. Ainsi, nous avons comparé les deux grandes méthodes de projection sur les HS, à savoir le calcul du produit scalaire (lorsque celui-ci est calculé par des méthodes Monte-Carlo) et l'estimation aux moindres carrées (Section A.3.1). Dans l'état de l'art, l'estimation aux moindres carrés démontre des qualités de résistance au bruit ainsi qu'une robustesse lorsque la projection est calculée avec peu d'échantillons [SHHS03, LLW06, VSG+13]. Cependant, aucune comparaison de convergence n'est réalisée entre les deux méthodes de projection dans l'état de l'art. Nous avons observé que les moindres carrés apportent une convergence plus rapide selon la nature de la fonction projetée ainsi que le nombre de bandes d'HS utilisées.

Dans un second temps, l'idée est de projeter le champ de radiance sur les HS en utilisant le modèle des VPL. Les méthodes de précalcul de transfert de radiance estiment les transferts à chaque point de la scène en lançant des chemins lumineux indépendant de chaque point. En contraste, les VPL sont mutualisés pour l'ensemble des points de la scène. La projection sur les HS du champ de radiance arrivant en chaque point de la scène devient alors moins coûteuse à calculer tout en générant une qualité d'éclairage final similaire.

A.4 Conclusion

En conclusion, nous avons présenté plusieurs contributions pour améliorer la projection sur les harmoniques sphériques de la radiance locale et globale ainsi que pour calculer efficacement l'éclairage final à chaque pixel. Nous avons proposé des méthodes efficaces pour calculer les coefficients d'HS représentant le champ de radiance produit par des lumières sphériques en tout point de la scène. Mais aussi, un nouveau cadre de travail pour calculer efficacement l'éclairage en chaque pixel avec des matériaux décomposable. Ainsi qu'un nouveau modèle de lumière virtuelle, les HVL, permettant de générer efficacement l'éclairage global sans artefacts.

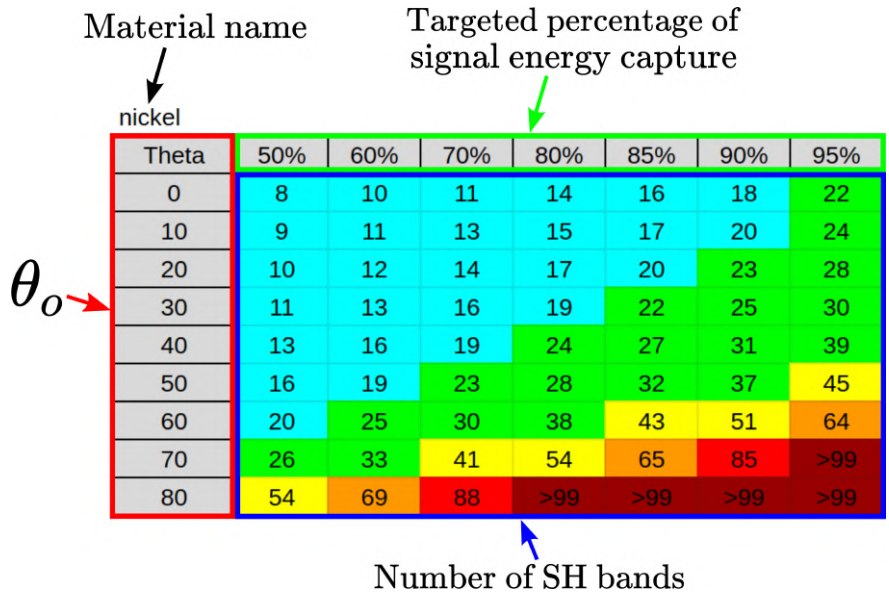
Le travail présenté dans cette thèse peut être amélioré par plusieurs leviers d'amélioration. Gautron *et al.* [GKPB04] dérivent à partir des HS une nouvelle base appelée les harmoniques hémisphériques, permettant donc de représenter seulement des signaux hémisphériques. Cela signifie que les harmoniques hémisphériques auront besoin d'impliquées moins de fonction de base pour représenter un signal à qualité égale. Ainsi, les travaux proposés dans cette thèse pourraient être adaptés à cette base. Enfin, les derniers travaux proposés dans cette thèse (Section A.3.4), notamment la projection efficace du champ de radiance sur les HS, manquent d'une comparaison approfondie afin de quantifier plus précisément le gain de temps ainsi que l'erreur d'approximation.

B Spherical harmonics material database

B.1 Brief presentation

In this thesis, we used different material models including the captured materials proposed by the *Mitsubishi Electric Research Laboratories* (MERL) [MPBM03] as well as the *Realistic Graphics Lab* (RGL) [DJ18]. We share on a git repository[1] the files storing the SH projection of the materials of the two databases as well as the code computing the SH projection and reading the files storing the SH coefficients. The next section (Section B.2) gives more detail on the SH projection and Table B.1 references the materials used in this thesis.

We also produced a study[2] to analyze the number of SH bands needed to capture the material according to a percentage of signal energy capture. In order to simplify the data visualization, this study is limited to isotropic material defined by $F(\omega_o, \omega_i) = F(\theta_o, \theta_i, |\phi_i - \phi_o|)$ Hence, when θ_o is fixed, the reflectance F is a spherical function that can be projected on the SH. We have computed SH projection tables according to θ_o and a target energy to capture the signal organized in the following form.



Theta	50%	60%	70%	80%	85%	90%	95%
0	8	10	11	14	16	18	22
10	9	11	13	15	17	20	24
20	10	12	14	17	20	23	28
30	11	13	16	19	22	25	30
40	13	16	19	24	27	31	39
50	16	19	23	28	32	37	45
60	20	25	30	38	43	51	64
70	26	33	41	54	65	85	>99
80	54	69	88	>99	>99	>99	>99

[1] https://gitlab.com/PierreMezieres/shmaterials
[2] Redirection link in the git repository.

For instance, this material requires 24 SH bands to capture 80% of the signal energy when $\theta_o = 40°$. As a result, we observe that mostly diffuse materials require at most about ten SH bands to be captured accurately. While specular materials may require about 20 bands up to more than 100 for the most specular materials. An order of magnitude that becomes particularly difficult to use for rendering. The study was conducted on spherical harmonics but also on hemispherical harmonics which allow us to observe that hemispherical harmonics require fewer coefficients to achieve a similar result obtained with spherical harmonics.

B.2 SH projection details

To compute BRDF coefficients, we use the approach proposed by Kautz *et al.* [KSS02]. A BRDF is a function of the two directions ω_i and ω_o, yielding a 4-dimensional function. When evaluating F, the direction ω_o is fixed; hence, the BRDF evaluation corresponds to a 2-dimensional function projected on SH. The SH coefficients are tabulated according to discrete samples of the direction $\omega_o = (\theta_o, \phi_o)$.

$$\mathrm{Tab}[\theta_o, \phi_o] = \mathbf{F}_{\omega_o} \tag{B.1}$$

where $\mathbf{F}_{\omega_o}$ is the SH coefficients vector of $\mathbf{F}$ computed for the output direction ω_o. The dabase is created with a sampling step of one degree for θ and ϕ. With such uniform sampling, a material has 90×360 samples of $\mathbf{F}_{\omega_o}$ and accessing the coefficients requires a constant time array lookup. The one degree step allows capturing subtle material variations. Other sampling and reconstruction strategies might be used but are out of scope of this thesis. As an optimization compared to the approach of Kautz *et al.* [KSS02], we store isotropic material in a one-dimensional array. Isotropic materials depend on the $\Delta\phi = |\phi_i - \phi_o|$ rather than the actual values of ϕ_i and ϕ_o. Hence, by using the isotropic parametrization of F:

$$F_{iso}(\theta_o, \theta_i, \Delta\phi) = F(\theta_o, 0, \theta_i, \Delta\phi) \ , \tag{B.2}$$

we drop the explicit dependence on ϕ_o and only discretize θ_o. This way, the storage of isotropic material drops down to 90 samples

$$\mathrm{Tab}_{iso}[\theta_o] = \mathbf{F}_{\omega_o} \text{ with } \omega_o = (\theta_o, 0) \ . \tag{B.3}$$

Limitation We use an equiangular parametrization of the input direction to project materials on SH. If this memory efficient parametrization scales well for isotropic materials, the memory cost for anisotropic ones quickly become prohibitive. Indeed, for an isotropic material, 90 directions are sampled; using 10 SH bands and 32-bit float RGB coefficients, one material represents 108kB. For instance, the required memory to use 200 different materials is 21MB. On the other hand, in the presence of anisotropic materials, the storage for one material is 360 times larger and raise to 388MB for one material. Investigating the parabolic parametrization proposed by Kautz *et al.* [KSS02] should help to reduce this memory cost. Indeed, using their 128×128 parabolic map would reduce the cost of one anisotropic material on 10 SH bands from 388MB to 20MB. Doing so, it must be first demonstrated that the resolution will be sufficient to capture the full behavior of the material according to the number of SH bands.

Fig.	Object	materials
Figure 3.6	Horse	23, 28, 32, 33, 38,
Figure 3.16	Fertility	29, 30, 35, 39, 40
Figure 5.3	Dragon	12, 18, 34, 35, 39
	Atrium	1, 2, 6, 11, 18
Figure 5.8	-	1, 4, 11
Figure 5.9 (Sponza)	Sponza	10, 11, 18, 19, 20, 22, 23 24, 25, 26, 31, 36, 37, 40
Figure 5.9 (Breakfast)	Table	7, 16, 36
	Chairs	16, 23, 32
	Other	3, 8, 15, 17, 18, 21, 22, 27, 33
Figure 5.10, 5.11	-	5, 9, 13, 14
Figure 5.16	-	23, 39

MERL [MPBM03] (1) alum-bronze, (2) alumina-oxide, (3) aluminium, (4) beige-fabric, (5) blue-acrylic, (6) blue-fabric, (7) blue-rubber, (8) gold-metallic-paint, (9) gold-paint, (10) green-plastic, (11) red-fabric2, (12) red-phenolic, (13) red-specular-plastic, (14) silver-metallic-paint, (15) silver-metallic-paint2, (16) teflon, (17) two-layer-gold, (18) white-fabric, (19) white-marble, (20) white-paint, (21) yellow-phenolic.

RGL [DJ18] (22) acrylic-felt-green, (23) acrylic-felt-orange, (24) acrylic-felt-pink, (25) acrylic-felt-purple, (26) acrylic-felt-white, (27) cardboard, (28) cc-amber-citrine, (29) cc-nothern-aurora (30) cm-toxic-green, (31) ilm-l3-37-metallic, (32) ilm-solo-m-68, (33) ilm-solo-illenium-falcon, (34) irid-flake-paint2, (35) paper-blue, (36) paper-red, (37) paper-yellow, (38) paper-white, (39) sating-gold, (40) vch-frozen-amethyst.

Table B.1: *Materials used from MERL/RGL database*

Bibliography

[AKDS04] Thomas Annen, Jan Kautz, Frédo Durand, and Hans-Peter Seidel. Spherical Harmonic Gradients for Mid-Range Illumination. In Alexander Keller and Henrik Wann Jensen, editors, *Eurographics Workshop on Rendering*. Eurographics Association, 2004. doi:10.2312/EGWR/EGSR04/331-336. ↑ 35, 54

[AP21] Jan Allmenröder and Christoph Peters. Linearly transformed spherical harmonics. *Interactive 3D graphics and games poster*, 2021. ↑ 35

[Arv95] James Arvo. Applications of irradiance tensors to the simulation of non-lambertian phenomena. In *Computer Graphics and Interactive Technique*, page 335–342. Association for Computing Machinery, 1995. doi:10.1145/218380.218467. ↑ 72

[BAS02] Stefan Brabec, Thomas Annen, and Hans-Peter Seidel. Shadow mapping for hemispherical and omnidirectional light sources. *Advances in Modelling, Animation and Rendering*, pages 397–407, 2002. doi:10.1007/978-1-4471-0103-1_25. ↑ 86

[BBH13] T. Barák, J. Bittner, and V. Havran. Temporally coherent adaptive sampling for imperfect shadow maps. In *Eurographics Symposium on Rendering*, EGSR '13, page 87–96. Eurographics Association, 2013. doi:10.1111/cgf.12154. ↑ 88

[BDBS22] Laurent Belcour, Thomas Deliot, Wilhem Barbier, and Cyril Soler. A data-driven paradigm for precomputed radiance transfer. *Computer Graphics and Interactive Technique*, 5(3):1–15, 2022. doi:10.1145/3543864. ↑ 41, 123

[Bli22] Blizzard Entertainment. *Overwatch 2*, 2022. ↑ 4, 5

[Blu22] BlueTwelve Studio. *Stray*. Annapurna Interactive, 2022. ↑ 4, 119

[BS08] Louis Bavoil and Miguel Sainz. Screen space ambient occlusion. *NVIDIA developer information*, 6, 2008. ↑ 38, 54

[Bun07] Bungie Studios. *Halo 3*. Microsoft Games Studios, 2007. ↑ 3

[BXH+18] Laurent Belcour, Guofu Xie, Christophe Hery, Mark Meyer, Wojciech Jarosz, and Derek Nowrouzezahrai. Integrating clipped spherical harmonics expansions. *ACM Trans. Graph.*, 37(2), March 2018. doi:10/gd52pf. ↑ 34, 36, 122

[Cam09] Cameron James. *Avatar*. 20th Centure Fox, Dune Entertainment, Giant Studios, Lightstorm Entertainment, Ingenious Film Partners, 2009. ↑ 6

[CD 20] CD Projekt RED. *Cyberpunk 2077*. CD Projekt, 2020. ↑ 4

[CDAS20] R R Currius, D Dolonius, U Assarsson, and E Sintorn. Spherical gaussian light-field textures for fast precomputed global illumination. *Computer Graphics Forum*, 39:133–146, 2020. doi:10.1111/cgf.13918. ↑ 41

[Chr08] Per Christensen. Point-based approximate color bleeding. *Pixar Technical Notes*, 2(5):6, 2008. ↑ 6, 88

[CIGR99] Cheol Ho Choi, Joseph Ivanic, Mark S. Gordon, and Klaus Ruedenberg. Rapid and stable determination of rotation matrices between spherical harmonics by direct recursion. *J. Chem. Phys.*, 111:8825–8831, 1999. doi:10.1063/1.480229. ↑ 21

[CJAMJ05] Petrik Clarberg, Wojciech Jarosz, Tomas Akenine-Möller, and Henrik Wann Jensen. Wavelet importance sampling: Efficiently evaluating products of complex functions. *ACM Trans. Graph.*, 24:1166–1175, 8 2005. doi:10/c79g6q. ↑ 40

[CL08] Hao Chen and Xinguo Liu. Lighting and material of halo 3. *SIGGRAPH Classes*, pages 1–22, 01 2008. doi:10.1145/1404435.1404437. ↑ 3

[CMS87] Brian Cabral, Nelson Max, and Rebecca Springmeyer. Bidirectional reflection functions from surface bump maps. *SIGGRAPH Comput. Graph.*, 21(4):273–281, aug 1987. doi:10.1145/37402.37434. ↑ 3, 29, 120, 122

[Cup12] Robert Cupisz. Light probe interpolation using tetrahedral tessellations. In *Game Developers Conference*, 2012. ↑ 37

[DGR+09] Zhao Dong, Thorsten Grosch, Tobias Ritschel, Jan Kautz, and Hans-Peter Seidel. Real-time indirect illumination with clustered visibility. *Vision, Modeling and Visualization Workshop*, 2009. ↑ 86, 127

[DH94] James R Driscoll and Dennis M Healy. Computing fourier transforms and convolutions on the 2-sphere. *Advances in Applied Mathematics*, 15(2):202–250, 1994. doi:10.1006/aama.1994.1008. ↑ 26

[DHB17] Jonathan Dupuy, Eric Heitz, and Laurent Belcour. A spherical cap preserving parameterization for spherical distributions. *ACM Trans. Graph.*, 36, 2017. doi:10.1145/3072959.3073694. ↑ 12, 36, 60

[DIC16] DICE. *Mirror's Edge Catalyst*. Electronic Arts, 2016. ↑ 1

[DJ18] Jonathan Dupuy and Wenzel Jakob. An adaptive parameterization for efficient material acquisition and rendering. *ACM Trans. Graph.*, 37(6), 2018. doi:10.1145/3272127.3275059. ↑ 30, 47, 61, 105, 129, 131

[DKH+10a] Tomáš Davidovič, Jaroslav Křivánek, Miloš Hašan, Philipp Slusallek, and Kavita Bala. Combining global and local virtual lights for detailed glossy illumination. *ACM Trans. Graph.*, 29(6), dec 2010. doi:10.1145/1882261.1866169. ↑ 86

[DKH+10b] Tomáš Davidovič, Jaroslav Křivánek, Miloš Hašan, Philipp Slusallek, and Kavita Bala. Combining global and local virtual lights for detailed glossy illumination. *ACM Trans. Graph.*, 29:1–8, 2010. ↑ 87

[DKH+14] Carsten Dachsbacher, Jaroslav Křivánek, Miloš Hašan, Adam Arbree, Bruce Walter, and Jan Novák. Scalable realistic rendering with many-light methods. *Computer Graphics Forum*, 33(1):88–104, feb 2014. doi:10.1111/cgf.12256. ↑ 83

[Dog20] Hawar Doghramachi. Lighting technology of the last of us part II. In *SIGGRAPH Talks*, SIGGRAPH '20. Association for Computing Machinery, 2020. doi:10.1145/3388767.3407358. ↑ 4

[DS05] Carsten Dachsbacher and Marc Stamminger. Reflective shadow maps. In *Interactive 3D graphics and games*, I3D '05, page 203–231. Association for Computing Machinery, 2005. doi:10.1145/1053427.1053460. ↑ 85, 126

[DS06] Carsten Dachsbacher and Marc Stamminger. Splatting indirect illumination. In *Interactive 3D graphics and games*, I3D '06, page 93–100. Association for Computing Machinery, 2006. doi:10.1145/1111411.1111428. ↑ 86

[DSJN19] Adrien Doubouchet, Peter-Pike Sloan, Wojciech Jarosz, and Derek Nowrouzezahrai. Impulse Responses for Precomputing Light from Volumetric Media. *Eurographics Symposium on Rendering*, 2019. doi:10/gf6rx8. ↑ 34, 37, 99

[ENSD12] Thomas Engelhardt, Jan Novák, Thorsten-W Schmidt, and Carsten Dachsbacher. Approximate bias compensation for rendering scenes with heterogeneous participating media. *Computer Graphics Forum*, 31:2145–2154, 2012. doi:10.1111/j.1467-8659.2012.03207.x. ↑ 87

[Fro22] FromSoftware. *Elden Ring*. Bandai Namco Entertainment, 2022. ↑ 2

[FTH22] Shin Fujieda, Yusuke Tokuyoshi, and Takahiro Harada. Stochastic light culling for single scattering in participating media. *Computer Graphics Forum*, 2022. ↑ 86

[GKPB04] Pascal Gautron, Jaroslav Krivanek, Sumanta Pattanaik, and Kadi Bouatouch. A Novel Hemispherical Basis for Accurate and Efficient Rendering. In Alexander Keller and Henrik Wann Jensen, editors, *Eurographics Workshop on Rendering*. Eurographics Association, 2004. doi:10.2312/EGWR/EGSR04/321-330. ↑ 19, 20, 103, 128

[GN15] Aude Giraud and Derek Nowrouzezahrai. Practical shading of height fields and meshes using spherical harmonic exponentiation. Eurographics Association, 2015. doi:10.2312/sre.20151161. ↑ 33

[Gre03] Robin Green. Spherical harmonic lighting: The gritty details. In *Game Developers Conference*, 2003. ↑ 17, 29

[GS10] Iliyan Georgiev and Philipp Slusallek. Simple and Robust Iterative Importance Sampling of Virtual Point Lights. In H. P. A. Lensch and S. Seipel, editors, *Eurographics - Short Papers*. Eurographics Association, 2010. doi:10.2312/egsh.20101047. ↑ 85

[GSHG98] Gene Greger, Peter Shirley, Philip M. Hubbard, and Donald P. Greenberg. The irradiance volume. *IEEE Computer Graphics Applications*, 18(2):32–43, mar 1998. doi:10.1109/38.656788. ↑ 37

[GT05] Otavio Good and Zachary Taylor. Optimized photon tracing using spherical harmonic light maps. In *SIGGRAPH 2005 Sketches*, SIGGRAPH '05, page 53–es. Association for Computing Machinery, 2005. doi:10.1145/1187112.1187175. ↑ 88

[GvL13] Gene H Golub and Charles F van Loan. *Matrix Computations*. JHU Press, fourth edition, 2013. ↑ 22

[Han88] Jesper E Hansen. *Spherical near-field antenna measurements*, volume 26. Iet, 1988. ↑ 49

[HDHN16] Eric Heitz, Jonathan Dupuy, Stephen Hill, and David Neubelt. Real-time polygonal-light shading with linearly transformed cosines. *ACM Trans. Graph.*, 35(4), jul 2016. doi:10.1145/2897824.2925895. ↑ 35

[HHS05] Vlastimil Havran, Robert Herzog, and Hans-Peter Seidel. Fast final gathering via reverse photon mapping. *Computer Graphics Forum*, 2005. doi:10.1111/j.1467-8659.2005.00857.x. ↑ 84

[HKWB09] Miloš Hašan, Jaroslav Křivánek, Bruce Walter, and Kavita Bala. Virtual Spherical Lights for Many-Light Rendering of Glossy Scenes. *ACM Trans. Graph.*, 28(5):1–6, 2009. doi:10.1145/1618452.1618489. ↑ 87, 92, 109, 127

[Hob31] Ernest William Hobson. *The theory of spherical and ellipsoidal harmonics*. Cambridge University Press, 1931. ↑ 17

[HPB07] Miloš Hašan, Fabio Pellacini, and Kavita Bala. Matrix row-column sampling for the many-light problem. *ACM Trans. Graph.*, pages 26–es, 2007. doi:10.1145/1276377.1276410. ↑ 86

[HRKM03] Dennis M Healy, Daniel N Rockmore, Peter J Kostelec, and Sean Moore. Ffts for the 2-sphere-improvements and variations. *Journal of Fourier analysis and applications*, 9:341–385, 2003. doi:10.1007/s00041-003-0018-9. ↑ 21, 111

[HSSS12] Sebastian Herholz, Timo Schairer, Andreas Schilling, and Wolfgang Straßer. Screen Space Spherical Harmonic Occlusion. In Michael Goesele, Thorsten Grosch, Holger Theisel, Klaus Toennies, and Bernhard Preim, editors, *Vision, Modeling and Visualization Workshop*. Eurographics Association, 2012. doi:10.2312/PE/VMV/VMV12/071-078. ↑ 38

[HVAPB08] Miloš Hašan, Edgar Velázquez-Armendáriz, Fabio Pellacini, and Kavita Bala. Tensor clustering for rendering many-light animations. *Computer Graphics Forum*, 27:1105–1114, 2008. doi:10.1111/j.1467-8659.2008.01248.x. ↑ 86

[HW10] Ralf Habel and Michael Wimmer. Efficient irradiance normal mapping. In *Interactive 3D graphics and games*, pages 189–195. Association for Computing Machinery, 2010. doi:10.1145/1730804.1730835. ↑ 103

[IDYN07] Kei Iwasaki, Yoshinori Dobashi, Fujiichi Yoshimoto, and Tomoyuki Nishita. Precomputed radiance transfer for dynamic scenes taking into account light interreflection. In *Eurographics Symposium on Rendering*, EGSR'07, page 35–44. Eurographics Association, 2007. doi:10.5555/2383847.2383854. ↑ 33

[IKA+21] Julius Ikkala, Petrus Kivi, Joel Alanko, Markku Mäkitalo, and Pekka Jääskeläinen. Ddish-gi: Dynamic distributed spherical harmonics global illumination. In *Computer Graphics International Conference*, page 433–451. Springer, 2021. doi:10.1007/978-3-030-89029-2_34. ↑ 37, 107, 122

[Inf19] Infinity Ward. *Call of Duty: Modern Warfare*. Activision, 2019. ↑ 1

[IR96] Joseph Ivanic and Klaus Ruedenberg. Rotation matrices for real spherical harmonics. direct determination by recursion. *The Journal of Physical Chemistry*, 100(15):6342–6347, 1996. doi:10.1021/jp953350u. ↑ 28

[IR98] Joseph Ivanic and Klaus Ruedenberg. Additions and corrections-rotation matrices for real spherical harmonics. direct determination by recursion. *Journal of Physical Chemistry A*, 102(45):9099, 1998. ↑ 28

[JCJ09] Wojciech Jarosz, Nathan A Carr, and Henrik Wann Jensen. Importance sampling spherical harmonics. *Computer Graphics Forum*, 28:577–586, 2009. doi:10.1111/j.1467-8659.2009.01398.x. ↑ 40

[JDZJ08] Wojciech Jarosz, Craig Donner, Matthias Zwicker, and Henrik Wann Jensen. Radiance caching for participating media. *ACM Trans. Graph.*, 27, 2008. doi:10.1145/1330511.1330518. ↑ 36

[Jen01] Henrik Wann Jensen. *Realistic image synthesis using photon mapping*, volume 364. Ak Peters Natick, 2001. ↑ 84

[JWPJ16] Jorge Jiménez, Xianchun Wu, Angelo Pesce, and Adrian Jarabo. Practical real-time strategies for accurate indirect occlusion. *SIGGRAPH Courses*, 2016. ↑ 38, 62

[JZJ08] Wojciech Jarosz, Matthias Zwicker, and Henrik Wann Jensen. Irradiance gradients in the presence of participating media and occlusions. In *Eurographics Symposium on Rendering*, 2008. doi:10.1111/j.1467-8659.2008.01246.x. ↑ 36

[Kaj86] James T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, aug 1986. doi:10.1145/15886.15902. ↑ 13, 121

[Kel97] Alexander Keller. Instant radiosity. In *Computer Graphics and Interactive Technique*, SIGGRAPH '97, page 49–56. Association for Computing Machinery, 1997. doi:10.1145/258734.258769. ↑ 84, 85, 127

[KGBP05] Jaroslav Křivánek, Pascal Gautron, Kadi Bouatouch, and Sumanta Pattanaik. Improved radiance gradient computation. In *Spring Conference on Computer Graphics*, 2005. doi:10.1145/1090122.1090148. ↑ 36, 37, 122

[KGW+08] Jaroslav Křivánek, Pascal Gautron, Greg Ward, Henrik Wann Jensen, Per H. Christensen, and Eric Tabellion. Practical global illumination with irradiance caching. In *SIGGRAPH Classes*, 2008. doi:10.1145/1401132.1401213. ↑ 36, 37

[KH01] Alexander Keller and Wolfgang Heidrich. Interleaved sampling. In *Proceedings of the 12th Eurographics Conference on Rendering*, EGWR'01, page 269–276. Eurographics Association, 2001. doi:10.5555/2383696.2383731. ↑ 102

[KHDN22] Aakash KT, Eric Heitz, Jonathan Dupuy, and PJ Narayanan. Bringing linearly transformed cosines to anisotropic ggx. *Computer Graphics and Interactive Technique*, 5(1):1–18, 2022. ↑ 36

[KISS15] Benjamin Keinert, Matthias Innmann, Michael Sänger, and Marc Stamminger. Spherical fibonacci mapping. *ACM Trans. Graph.*, 34(6), oct 2015. doi:10.1145/2816795.2818131. ↑ 15, 104, 105, 109

[KK06] Thomas Kollig and Alexander Keller. Illumination in the presence of weak singularities. In *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 245–257. Springer, 2006. ↑ 87

[KKP+06] Jaroslav Křivánek, Jaakko Konttinen, Sumanta Pattanaik, Kadi Bouatouch, and Jirí Žára. Fast approximation to spherical harmonic rotation. In *Proceedings of the 22nd Spring Conference on Computer Graphics*, SCCG '06, page 49–58. Association for Computing Machinery, 2006. doi:10.1145/2602161.2602167. ↑ 29

[KSS02] Jan Kautz, Peter-Pike Pike Sloan, and John Snyder. Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics. In *Eurographics Workshop on Rendering*, volume 2, pages 291–296. Eurographics Association, 2002. doi:10.5555/581896.581934. ↑ 29, 30, 34, 130

[KYM22] Takahiro Kuge, Tatsuya Yatagawa, and Shigeo Morishima. Real-time shading with free-form planar area lights using linearly transformed cosines. *Journal of Computer Graphics Techniques*, 11(1):1–16, February 2022. ↑ 36

[LDdLRB16] Gilles Laurent, Cyril Delalandre, Grégoire de La Rivière, and Tamy Boubekeur. Forward light cuts: A scalable approach to real-time global illumination. *Computer Graphics Forum*, 35:79–88, 2016. doi:10.1111/cgf.12951. ↑ 63, 86

[Lea07] L Gary Leal. *Advanced transport phenomena: fluid mechanics and convective transport processes*, volume 7. Cambridge University Press, 2007. ↑ 75

[Leh07] Jaakko Lehtinen. A framework for precomputed and captured light transport. *ACM Trans. Graph.*, 26, 2007. doi:10.1145/1289603.1289604. ↑ 34

[LHL+21] Linjie Lyu, Marc Habermann, Lingjie Liu, Mallikarjun B R, Ayush Tewari, and Christian Theobalt. Efficient and differentiable shadow computation for inverse problems. In *IEEE International Conference on Computer Vision*, 2021. ↑ 38

[LLK07] Samuli Laine, Jaakko Lehtinen, and Janne Kontkanen. Incremental instant radiosity for real-time indirect illumination. *Eurographics Symposium on Rendering*, pages 4–8, 2007. doi:10.2312/EGWR/EGSR07/277-286. ↑ 85

[LLW06] Ping-Man Lam, Chi-Sing Leung, and Tien-Tsin Wong. Noise-resistant fitting for spherical harmonics. *IEEE Transactions on Visualization and Computer Graphics*, 12:254–265, 2006. doi:10.1109/TVCG.2006.34. ↑ 22, 111, 128

[LTL+19] Hsueh-Ti Derek Liu, Michael Tao, Chun-Liang Li, Derek Nowrouzezahrai, and Alec Jacobson. Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer. In *ICLR*, 2019. ↑ 38, 49

[LWLY22] Tianyu Li*, Wenyou Wang*, Daqi Lin, and Cem Yuksel. Virtual blue noise lighting. *Computer Graphics and Interactive Technique(Proceedings of HPG 2022)*, 5(3):3:1–3:25, 07 2022. (*Joint First Authors). doi:10.1145/3543872. ↑ 88

[LY19] Daqi Lin and Cem Yuksel. Real-Time Rendering with Lighting Grid Hierarchy. *Computer Graphics and Interactive Technique*, 2(1):1–17, 2019. doi:10.1145/3321361. ↑ 85

[LZBD21] Fujun Luan, Shuang Zhao, Kavita Bala, and Zhao Dong. Unified Shape and SVBRDF Recovery using Differentiable Monte Carlo Rendering. *Computer Graphics Forum*, 40(4):101–113, 2021. doi:10.1111/cgf.14344. ↑ 38

[Mas16] Massive Entertainment. *Tom Clancy's The Division*. Ubisoft, 2016. ↑ 5

[Mas22] Massive Monster. *Cult of the Lamb*. Devolver Digital, 2022. ↑ 2

[MDVP22] Pierre Mézières, François Desrichard, David Vanderhaeghe, and Mathias Paulin. Harmonics virtual lights: Fast projection of luminance field on spherical harmonics for efficient rendering. *Computer Graphics Forum*, 2022. doi:10.1111/cgf.14564. ↑ 7

[MGNM19] Zander Majercik, Jean-Philippe Guertin, Derek Nowrouzezahrai, and Morgan McGuire. Dynamic diffuse global illumination with ray-traced irradiance fields. *Journal of Computer Graphics Techniques*, 8:1–30, 2019. ↑ 37

[MJJG18] Julio Marco, Adrian Jarabo, Wojciech Jarosz, and Diego Gutierrez. Second-order occlusion-aware volumetric radiance caching. *ACM Trans. Graph.*, 37, 2018. doi:10.1145/3185225. ↑ 62

[MK20] Jonathan B Metzgar and Semwal Sudhanshu K. Scalable spherical harmonics hierarchies. *WSCG*, 2020. doi:10.24132/CSRN.2020.3001.14. ↑ 107

[MKC07] Ricardo Marroquim, Martin Kraus, and Paulo Roma Cavalcanti. Efficient Point-Based Rendering Using Image Reconstruction. In M. Botsch, R. Pajarola, B. Chen, and M. Zwicker, editors, *Eurographics Symposium on Point-Based Graphics*. Eurographics Association, 2007. doi:10.2312/SPBG/SPBG07/101-108. ↑ 86

[MMBP22] Pierre Mézières, Nicolas Mellado, Loïc Barthe, and Mathias Paulin. Recursive analytic spherical harmonics gradient for spherical lights. *Computer Graphics Forum*, 41(2):393–406, 2022. doi:10.1111/cgf.14482. ↑ 7, 70

[MMNL17] Morgan McGuire, Mike Mara, Derek Nowrouzezahrai, and David Luebke. Real-time global illumination using precomputed light field probes. *Interactive 3D graphics and games*, pages 1–11, 2017. ↑ 37

[MMSM21] Zander Majercik, Adam Marrs, Josef Spjut, and Morgan McGuire. Scaling probe-based real-time dynamic global illumination for production. *Journal of Computer Graphics Techniques*, 10:1–29, 5 2021. ↑ 37

[MP20] Pierre Mézières and Mathias Paulin. Projection efficace de lumières sphériques sur les harmoniques sphériques. In *Journées Françaises d'Informatique Graphique (JFIG 2020)*, Nancy (en distanciel), France, November 2020. Prix du meilleur papier. ↑ 7

[MP21] Pierre Mézières and Mathias Paulin. Efficient spherical harmonic shading for separable brdf. *ACM Digital Library*, 2021. doi:10.1145/3478512.3488597. ↑ 7

[MPBM03] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Trans. Graph.*, 22(3):759–769, July 2003. ↑ 12, 30, 129, 131

[MRC07] Dhruv Mahajan, Ravi Ramamoorthi, and Brian Curless. A theory of frequency domain invariants: Spherical harmonic identities for brdf/lighting transfer and image consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:197–213, 2007. doi:10.1007/11744085_4. ↑ 39

[MRNK21] Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. Real-time neural radiance caching for path tracing. *ACM Trans. Graph.*, 40:36:1–36:16, 8 2021. doi:10.1145/3450626.3459812. ↑ 37

[MS16] Josiah Manson and Peter-Pike Sloan. Fast filtering of reflection probes. *Computer Graphics Forum*, 35:119–127, 2016. doi:10.5555/3071773.3071786. ↑ 37

[MST+20] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020. ↑ 39

[MW11] Jason D McEwen and Yves Wiaux. A novel sampling theorem on the sphere. *IEEE Transactions on Signal Processing*, 59:5876–5887, 2011. doi:10.1109/TSP.2011.2166394. ↑ 21, 111

[Nau16] Naughty Dog. *Uncharted 4: A Thief's End*. Sony Interactive Entertainment, 2016. ↑ 4

[Nau20] Naughty Dog. *The Last of Us Part II*. Sony Interactive Entertainment, 2020. ↑ 1, 4

[NDDJK21] Merlin Nimier-David, Zhao Dong, Wenzel Jakob, and Anton Kaplanyan. Material and Lighting Reconstruction for Complex Indoor Scenes with Texture-space Differentiable Rendering. In *Eurographics Symposium on Rendering*, 2021. doi:10.2312/sr.20211292. ↑ 38

[NED11] Jan Novák, Thomas Engelhardt, and Carsten Dachsbacher. Screen-space bias compensation for interactive high-quality global illumination with virtual point lights. In *Interactive 3D graphics and games*, I3D '11, page 119–124. Association for Computing Machinery, 2011. doi:10.1145/1944745.1944765. ↑ 87

[Ner22] Nerial. *Card Shark*. Devolver Digital, 2022. ↑ 2

[NNDJ12a] Jan Novák, Derek Nowrouzezahrai, Carsten Dachsbacher, and Wojciech Jarosz. Progressive virtual beam lights. *Computer Graphics Forum*, 31(4):1407–1413, jun 2012. doi:10.1111/j.1467-8659.2012.03136.x. ↑ 88

[NNDJ12b] Jan Novák, Derek Nowrouzezahrai, Carsten Dachsbacher, and Wojciech Jarosz. Virtual ray lights for rendering scenes with participating media. *ACM Trans. Graph.*, 31, 07 2012. doi:10.1145/2185520.2185556. ↑ 88

[NRH03] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. All-frequency shadows using nonlinear wavelet lighting approximation. *ACM Trans. Graph.*, 22(3):376–381, jul 2003. doi:10.1145/882262.882280. ↑ 39, 40, 123

[NRH04] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. Triple product wavelet integrals for all-frequency relighting. *ACM Trans. Graph.*, 23:477–487, 8 2004. doi:10.1145/1015706.1015749. ↑ 23, 40

[NSF12] Derek Nowrouzezahrai, Patricio Simari, and Eugene Fiume. Sparse zonal harmonic factorization for efficient sh rotation. *ACM Trans. Graph.*, 31(3), jun 2012. doi:10.1145/2167076.2167081. ↑ 25, 26, 29, 34, 36, 53, 122

[NW09] Greg Nichols and Chris Wyman. Multiresolution splatting for indirect illumination. In *Interactive 3D graphics and games*, I3D '09, page 83–90. Association for Computing Machinery, 2009. doi:10.1145/1507149.1507162. ↑ 86

[NWNB15] Caitlin Nortje, Wil Ward, Bartosz Neuman, and Li Bai. Spherical harmonics for surface parametrisation and remeshing. *Mathematical Problems in Engineering*, 2015:1–11, 01 2015. doi:10.1155/2015/582870. ↑ 22

[OBM06] Brian Osman, Mike Bukowski, and Chris McEvoy. Practical implementation of dual paraboloid shadow maps. In *SIGGRAPH Symposium on Videogames*, page 103–106. Association for Computing Machinery, 2006. doi:10.1145/1183316.1183331. ↑ 86

[OP11] Jiawei Ou and Fabio Pellacini. Lightslice: matrix slice sampling for the many-lights problem. *ACM Trans. Graph.*, 30:179, 2011. ↑ 86

[OS07] Christopher Oat and Pedro V. Sander. Ambient aperture lighting. In *Interactive 3D graphics and games*, I3D '07, page 61–64. Association for Computing Machinery, 2007. doi:10.1145/1230100.1230111. ↑ 92

[OSK+14] Ola Olsson, Erik Sintorn, Viktor Kämpe, Markus Billeter, and Ulf Assarsson. Efficient virtual shadow maps for many lights. In *Interactive 3D graphics and games*, page 87–96. Association for Computing Machinery, 2014. doi:10.1145/2556700.2556701. ↑ 86

[Pet21] Christoph Peters. Brdf importance sampling for polygonal lights. *ACM Trans. Graph.*, 40(4), jul 2021. doi:10.1145/3450626.3459672. ↑ 35

[PJH16] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation (3rd ed.)*. Morgan Kaufmann Publishers Inc., 3rd edition, November 2016. ↑ 95, 109

[PKD12] Roman Prutkin, Anton Kaplanyan, and Carsten Dachsbacher. Reflective Shadow Map Clustering for Real-Time Global Illumination. *Eurographics*, 2012. doi:10.2312/conf/EG2012/short/009-012. ↑ 88

[Ram09] Ravi Ramamoorthi. Precomputation-based rendering. *Found. Trends. Comput. Graph. Vis.*, 3(4):281–369, apr 2009. doi:10.1561/0600000021. ↑ 34

[RBMS17] Mickaël Ribardière, Benjamin Bringier, Daniel Meneveaux, and Lionel Simonot. STD: Student's t-Distribution of Slopes for Microfacet Based BSDFs. *Computer Graphics Forum*, 36(2):421–429, 2017. doi:10.1111/cgf.13137. ↑ 95

[RBRD22] Gilles Rainer, Adrien Bousseau, Tobias Ritschel, and George Drettakis. Neural precomputed radiance transfer. *Computer Graphics Forum*, 41(2):365–378, 2022. doi:10.1111/cgf.14480. ↑ 41, 123

[RDGK12] Tobias Ritschel, Carsten Dachsbacher, Thorsten Grosch, and Jan Kautz. The state of the art in interactive global illumination. *Computer Graphics Forum*, 31(1):160–188, 2012. doi:10.1111/j.1467-8659.2012.02093.x. ↑ 83

[REG+09] Tobias Ritschel, Thomas Engelhardt, Thorsten Grosch, H-P Seidel, Jan Kautz, and Carsten Dachsbacher. Micro-rendering for scalable, parallel final gathering. *ACM Trans. Graph.*, 28:1–8, 2009. doi:10.1145/1618452.1618478. ↑ 85

[REH+11] Tobias Ritschel, Elmar Eisemann, Inwoo Ha, James D K Kim, and Hans-Peter Seidel. Making imperfect shadow maps view-adaptive: High-quality global illumination in large dynamic scenes. *Computer Graphics Forum*, 30:2258–2269, 2011. doi:10.1111/j.1467-8659.2011.01998.x. ↑ 86

[RGK+08] Tobias Ritschel, Thorsten Grosch, Min H Kim, H. P. Seidel, Carsten Dachsbacher, and Jan Kautz. Imperfect shadow maps for efficient computation of indirect illumination. In *ACM SIGGRAPH Asia 2008 Pap. SIGGRAPH Asia'08*, volume 27, pages 1–8. Association for Computing Machinery, 2008. doi:10.1145/1457515.1409082. ↑ 86

[RH01] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *SIGGRAPH*, pages 497–500, 2001. doi:10.1145/383259.383317. ↑ 12, 30, 67, 98, 107, 122

[RH02] Ravi Ramamoorthi and Pat Hanrahan. Frequency space environment map rendering. *ACM Trans. Graph.*, 21(3):517–526, jul 2002. doi:10.1145/566654.566611. ↑ 33

[RLP+20] Simon Rodriguez, Thomas Leimkühler, Siddhant Prakash, Chris Wyman, Peter Shirley, and George Drettakis. Glossy probe reprojection for interactive global illumination. *ACM Trans. Graph.*, 39:1–16, 2020. doi:10.1145/3414685.3417823. ↑ 37

[Roc18] Rockstar Studios. *Red Dead Redemption II*. Rockstar Games, 2018. ↑ 2, 119

[RWS+06] Zhong Ren, Rui Wang, John Snyder, Kun Zhou, Xinguo Liu, Bo Sun, Peter-Pike Sloan, Hujun Bao, Qunsheng Peng, and Baining Guo. Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. In *SIGGRAPH 2006 Papers*, SIGGRAPH '06, page 977–986. Association for Computing Machinery, 2006. doi:10.1145/1179352.1141982. ↑ 25, 32, 33, 67

[SAWG91] Françis X. Sillion, James R. Arvo, Stephen H. Westin, and Donald P. Greenberg. A global illumination solution for general reflectance distributions. *SIGGRAPH*, 25(4):187–196, jul 1991. doi:10.1145/127719.122739. ↑ 14

[SBN15] Cyril Soler, Mahdi M. Bagher, and Derek Nowrouzezahrai. Efficient and accurate spherical kernel integrals using isotropic decomposition. *ACM Trans. Graph.*, 34(5), November 2015. doi:10.1145/2797136. ↑ 25, 35, 122

[Sch13] Nathanaël Schaeffer. Efficient spherical harmonic transforms aimed at pseudospectral numerical simulations. *Geochemistry, Geophysics, Geosystems*, 14:751–758, 2013. doi:10.1002/ggge.20071. ↑ 21, 111

[SHD15] Florian Simon, Johannes Hanika, and Carsten Dachsbacher. Rich-VPLs for Improving the Versatility of Many-Light Methods. *Computer Graphics Forum*, 34:575–584, 2015. doi:10.1111/cgf.12585. ↑ 88, 111

[SHHS03] Peter-Pike Sloan, Jesse Hall, John Hart, and John Snyder. Clustered principal components for precomputed radiance transfer. *ACM Trans. Graph.*, 22:382–391, 2003. doi:10.1145/882262.882281. ↑ 22, 34, 111, 128

[SIE18] SIE Santa Monica Studio. *God of War*. Sony Interactive Entertainment, 2018. ↑ 1, 4

[SIMP06] Benjamin Segovia, Jean Claude Iehl, Richard Mitanchey, and Bernard Péroche. Bidirectional instant radiosity. In *Eurographics Symposium on Rendering*, pages 389–397. Eurographics Association, 2006. doi:10.5555/2383894.2383944. ↑ 85, 126

[SIP07] Benjamin Segovia, Jean Claude Iehl, and Bernard Péroche. Metropolis instant radiosity. *Computer Graphics Forum*, 26:425–434, 2007. doi:10.1111/j.1467-8659.2007.01065.x. ↑ 85, 126

[SJJ12] Jorge Schwarzhaupt, Henrik Wann Jensen, and Wojciech Jarosz. Practical Hessian-based error control for irradiance caching. *ACM Trans. Graph.*, 31(6), November 2012. doi:10/gbb6n4. ↑ 62

[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*, pages 527–536, 2002. doi:10.1145/566570.566612. ↑ 3, 31, 34, 40, 120, 122

[SL17] Ari Silvennoinen and Jaakko Lehtinen. Real-time global illumination by precomputed local reconstruction from sparse radiance probes. *ACM Trans. Graph.*, 36(6), 2017. doi:10.1145/3130800.3130852. ↑ 37, 110

[Sla99] Gregory G Slabaugh. Computing euler angles from a rotation matrix. 1999. ↑ 26

[Slo06] Peter-Pike Sloan. Normal mapping for precomputed radiance transfer. In *Interactive 3D graphics and games*, pages 23–26. Association for Computing Machinery, 2006. doi:10.1145/1111411.1111415. ↑ 34

[Slo08] Peter-Pike Sloan. Stupid Spherical Harmonics (SH) Tricks. *Game Developers Conference*, 9:320–321, 2008. ↑ 17, 22, 23, 46, 51, 54, 55

[Slo13] Peter-Pike Sloan. Efficient Spherical Harmonic Evaluation. *Journal of Computer Graphics Techniques*, 2(2):84–90, sep 2013. ↑ 19, 43, 46, 47, 48, 50, 51, 62, 103, 124

[Slo17] Peter Pike Sloan. Deringing spherical harmonics. In *SIGGRAPH Asia 2017 Technical Briefs*, pages 1–4, 2017. doi:10.1145/3145749.3149438. ↑ 22

[SLS05] Peter Pike Sloan, Ben Luna, and John Snyder. Local, deformable precomputed radiance transfer. In *ACM Trans. Graph.*, volume 24, pages 1216–1224. Association for Computing Machinery, 2005. doi:10.1145/1073204.1073335. ↑ 25, 28, 33, 34, 122

[SR09] Bo Sun and Ravi Ramamoorthi. Affine double-and triple-product wavelet integrals for rendering. *ACM Trans. Graph.*, 28:1–17, 2009. doi:10.1145/1516522.1516525. ↑ 40

[SS21] Ari Silvennoinen and Peter-Pike Sloan. Moving Basis Decomposition for Precomputed Light Transport. *Computer Graphics Forum*, 40(4):127–137, 2021. doi:10.1111/cgf.14346. ↑ 37

[SSS+20] Dario Seyb, Peter-Pike Sloan, Ari Silvennoinen, Michał Iwanicki, and Wojciech Jarosz. The design and evolution of the uberbake light baking system. *ACM Trans. Graph.*, 39, 7 2020. doi:10.1145/3386569.3392394. ↑ 1

[STC+22] Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Computer Vision and Pattern Recognition*, 2022. ↑ 39, 62, 112

[TH16] Yusuke Tokuyoshi and Takahiro Harada. Stochastic light culling. *Journal of Computer Graphics Techniques*, 5(1), 2016. ↑ 86

[TH17] Yusuke Tokuyoshi and Takahiro Harada. Stochastic light culling for VPLs on GGX microsurfaces. *Computer Graphics Forum*, 36(4):55–63, jul 2017. doi:10.1111/cgf.13224. ↑ 86

[Tok15a] Yusuke Tokuyoshi. Fast Indirect Illumination Using Two Virtual Spherical Gaussian Lights. In *SIGGRAPH Asia 2015 Posters*, SA '15. Association for Computing Machinery, 2015. doi:10.1145/2820926.2820929. ↑ 99, 100

[Tok15b] Yusuke Tokuyoshi. Virtual Spherical Gaussian Lights for Real-time Glossy Indirect Illumination. In *Computer Graphics Forum*, volume 34, pages 89–98. Eurographics Association, 2015. doi:10.1111/cgf.12748. ↑ 88, 99, 100, 101, 102, 127

[Ver06] Verbinski Gore. *Pirates of the Caribbean: Dead Man's Chest*. Walt Disney Pictures, Jerry Bruckheimer Films, 2006. ↑ 6

[VGS+19] N. Vibert, A. Gruson, H. Stokholm, T Mortensen, W. Jarosz, T. Hachisuka, and D. Nowrouzezahrai. Scalable virtual ray lights rendering for participating media. *Computer Graphics Forum*, 38(4):57–65, 2019. doi:10.1111/cgf.13770. ↑ 88

[VS83] Adriaan Van Oosterom and Jan Strackee. The Solid Angle of a Plane Triangle. *IEEE Transactions on Biomedical Engineering*, BME-30(2):125–126, 1983. doi:10.1109/TBME.1983.325207. ↑ 72

[VSG+13] K. Vanhoey, B. Sauvage, O. Génevaux, F. Larue, and J.-M. Dischler. Robust fitting on poorly sampled data for surface light field rendering and image relighting. *Computer Graphics Forum*, 32(6):101–112, 2013. doi:10.1111/cgf.12073. ↑ 22, 111, 128

[WABG06] Bruce Walter, Adam Arbree, Kavita Bala, and Donald P Greenberg. Multidimensional lightcuts. *ACM Trans. Graph.*, pages 1081–1088, 2006. doi:10.1145/1141911.1141997. ↑ 86

[WCZR20] Lifan Wu, Guangyan Cai, Shuang Zhao, and Ravi Ramamoorthi. Analytic spherical harmonic gradients for real-time rendering with many polygonal area lights. *ACM Trans. Graph.*, 39, 2020. doi:10.1145/3386569.3392373. ↑ 6, 7, 35, 36, 43, 47, 53, 54, 55, 58, 59, 60, 61, 62, 67, 72, 73, 75, 77, 112, 115, 120, 123, 124, 125

[WFA+05] Bruce Walter, Sebastian Fernandez, Adam Arbree, Kavita Bala, Michael Donikian, and Donald P Greenberg. Lightcuts: A scalable approach to illumination. *ACM Trans. Graph.*, 24:1098–1107, 2005. doi:10.1145/1073204.1073318. ↑ 86, 127

[WH92] Gregory J. Ward and Paul S. Heckbert. Irradiance gradients. In *SIGGRAPH Classes*, 1992. doi:10.1145/1401132.1401225. ↑ 36

[Wil78] Lance Williams. Casting curved shadows on curved surfaces. In *Computer Graphics and Interactive Technique*, pages 270–274. Association for Computing Machinery, 1978. doi:10.1145/800248.807402. ↑ 84, 86

[WKB+02] Ingo Wald, Thomas Kollig, Carsten Benthin, Alexander Keller, and Philipp Slusallek. Interactive global illumination using fast ray tracing. In *Eurographics Workshop on Rendering*, pages 15–24, 2002. doi:10.5555/581896.581899. ↑ 85

[WKB12] Bruce Walter, Pramook Khungurn, and Kavita Bala. Bidirectional lightcuts. *ACM Trans. Graph.*, 31:1–11, 2012. doi:10.1145/2185520.2185555. ↑ 86

[WKKN19] Yue Wang, Soufiane Khiat, Paul G. Kry, and Derek Nowrouzezahrai. Fast non-uniform radiance probe placement and tracing. In *Interactive 3D graphics and games*, I3D '19. Association for Computing Machinery, 2019. doi:10.1145/3306131.3317024. ↑ 37

[WMB15] Beibei Wang, Xiangxu Meng, and Tamy Boubekeur. Wavelet point-based global illumination. *Computer Graphics Forum*, 34(4):143–154, 2015. doi:10.5555/2858834.2858849. ↑ 88

[WMLT07] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. In *Eurographics Symposium on Rendering*, volume 2007, page 18th. Eurographics Association, 2007. doi:10.5555/2383847.2383874. ↑ 12

[WR18] Jingwen Wang and Ravi Ramamoorthi. Analytic spherical harmonic coefficients for polygonal area lights. *ACM Trans. Graph.*, 37, 2018. doi:10.1145/3197517.3201291. ↑ 6, 7, 34, 35, 36, 43, 47, 59, 60, 62, 73, 75, 76, 77, 115, 120, 122, 123, 125

[WRC88] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. A ray tracing solution for diffuse interreflection. *SIGGRAPH Comput. Graph.*, 22(4):85–92, jun 1988. doi:10.1145/378456.378490. ↑ 14, 36

[WRG+09] Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. All-frequency rendering of dynamic, spatially-varying reflectance. *ACM Trans. Graph.*, 28(5):1–10, dec 2009. doi:10.1145/1618452.1618479. ↑ 40, 123

[WXZ+06] Jiaping Wang, Kun Xu, Kun Zhou, Stephen Lin, Shimin Hu, and Baining Guo. Spherical harmonics scaling. *The Visual Computer*, 22:713–720, 2006. doi:10.1007/s00371-006-0057-8. ↑ 27, 33

[XCM+14] Kun Xu, Yan-Pei Cao, Li-Qian Ma, Zhao Dong, Rui Wang, and Shi-Min Hu. A practical algorithm for rendering interreflections with all-frequency brdfs. *ACM Trans. Graph.*, 33(1), feb 2014. doi:10.1145/2533687. ↑ 40, 63

[XSD+13] Kun Xu, Wei-Lun Sun, Zhao Dong, Dan-Yong Zhao, Run-Dong Wu, and Shi-Min Hu. Anisotropic spherical gaussians. *ACM Trans. Graph.*, 32(6), nov 2013. doi:10.1145/2508363.2508386. ↑ 40, 41

[XZA+21] Hanggao Xin, Zhiqian Zhou, Di An, Ling-Qi Yan, Kun Xu, Shi-Min Hu, and Shing-Tung Yau. Fast and accurate spherical harmonics products. *ACM Trans. Graph.*, 40(6), dec 2021. doi:10.1145/3478513.3480563. ↑ 25, 33

[YLT+21] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *International Conference on Computer Vision*, 2021. ↑ 39, 62, 112

[Yuk20] Cem Yuksel. Point light attenuation without singularity. In *ACM SIGGRAPH 2020 Talks*, SIGGRAPH '20. Association for Computing Machinery, 2020. doi:10.1145/3388767.3407364. ↑ 88, 109, 113

[YY17] Can Yuksel and Cem Yuksel. Lighting grid hierarchy for self-illuminating explosions. *ACM Trans. Graph.*, 36(4), jul 2017. doi:10.1145/3072959.3073604. ↑ 85

[ZBN19] Yangyang Zhao, Laurent Belcour, and Derek Nowrouzezahrai. View-dependent radiance caching. In *Graphics Interface*, 2019. doi:10.20380/GI2019.22. ↑ 37

[ZHL+05] Kun Zhou, Yaohua Hu, Stephen Lin, Baining Guo, and Harry Shum. Precomputed shadow fields for dynamic scenes. *ACM Trans. Graph.*, 24:1196–1201, 2005. doi:10.1145/1186822.1073332. ↑ 24, 32, 33

[ZMY+20] Cheng Zhang, Bailey Miller, Kai Yan, Ioannis Gkioulekas, and Shuang Zhao. Path-space differentiable rendering. *ACM Trans. Graph.*, 39, 2020. doi:10.1145/3386569.3392383. ↑ 38, 39

[ZW20] Zihong Zhou and Li-Yi Wei. Spherical light integration over spherical caps via spherical harmonics. In *SIGGRAPH Asia 2020 Technical Communications*, SA '20. Association for Computing Machinery, 2020. doi:10.1145/3410700.3425427. ↑ 36

[ZWZ+19] Cheng Zhang, Lifan Wu, Changxi Zheng, Ioannis Gkioulekas, Ravi Ramamoorthi, and Shuang Zhao. A differential theory of radiative transfer. *ACM Trans. Graph.*, 38:1–16, 2019. doi:10.1145/3355089.3356522. ↑ 38