Analyse fréquencielle du transport de la lumière : de la théorie aux applications

Laurent Belcour

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques Appliquées, Informatique**

Présentée par

Laurent Belcour

Thèse dirigée par **Nicolas Holzschuch**
et codirigée par **Cyril Soler**

préparée au sein du **Laboratoire Jean Kuntzmann**
et de **École Doctorale Mathèmatiques, Sciences et Technologies de l'Information, Informatique de Grenoble**

A Frequency Analysis of Light Transport

from theory to implementation

Thèse soutenue publiquement le **30 Octobre 2012**,
devant le jury composé de :

Mathias Paulin
Professeur à l'Université Paul Sabatier, Toulouse, Rapporteur
Matthias Zwicker
Professeur à l'Université de Bern, Rapporteur
Valérie Perrier
Professeur à l'École Nationale Supérieure d'Informatique et de Mathématiques Appliquées de Grenoble, Examinatrice
Wojciech Jarosz
Research Scientist à Walt Disney Company, Examinateur
Nicolas Holzschuch
Directeur de Recherche Inria, Directeur de thèse
Cyril Soler
Chargé de Recherche Inria, Co-Directeur de thèse

Contents

List of Figures

Acknowledgements

The following data where used to produce results for this document:

- *Lena* image (USC Signal and Image Processing Institue)
  `http://sipi.usc.edu/database/database.php?volume=misc`
- *Cornell box* scene (Goral et al. [61])
  `http://www.graphics.cornell.edu/online/box/`
- *Suzanne* model (Willem-Paul van Overbruggen)
  `http://www.blender.org/`
- *Snooker* scene (Soler et al. [164])
- *Helicopter* scene (modeled by vklidu from BlenderArtists.org)
  `http://blenderartists.org/forum/showthread.php?228226-Damaged-Helicopter-(Merkur)`
- *Spot fog* scene (Pharr and Humphreys [136])
  `http://www.pbrt.org/`
- *Sibenik cathedral* scene (modeled by Marko Dabrovic)
  `http://graphics.cs.williams.edu/data/meshes.xml`
- *Cautics from the glass sphere* scene (Jensen and Christensen [92])
- *Soccer boy figurine* scene (Sun et al. [168])

Copyrighted images used in this dissertation:

- *Genesis* by Axel Ritter (Figure 1.2)
- *Nobody is lucky in the same way* by Vasily Bodnar (Figure 1.2)
- *Pianist* by Timour Abdulov (Figure 1.2)
- *Fog* by Ric Coles (Figure 1.3b)
- *Detector eye* by Madeleine Price Ball

Associated Material

The following publications are part of this thesis:

- Laurent Belcour and Cyril Soler. Frequency based kernel estimation for progressive photon mapping. In *SIGGRAPH Asia 2011 Posters*, SA '11, pages 47:1–47:1. ACM, 2011. ISBN 978-1-4503-1137-3. doi: 10.1145/2073304.2073357

- Mahdi M. Bagher, Cyril Soler, Kartic Subr, Laurent Belcour, and Nicolas Holzschuch. Interactive rendering of acquired materials on dynamic geometry using bandwidth prediction. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '12, pages 127–134. ACM, 2012. doi: 10.1145/2159616.2159637

The following reference is currently in preparation for publication:

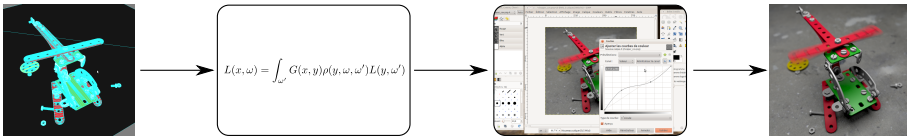
- Laurent Belcour, Cyril Soler, Kartic Subr, Frédo Durand, and Nicolas Holzschuch. 5D Covariance Tracing for Efficient Depth of Field and Motion Blur. *ACM Transactions of Graphics*, 201X. Accepted with minor modifications

The following reference is a work in preparation for submission:

- Laurent Belcour, Cyril Soler, and Kavita Bala. Frequency analysis of participating media. Technical report, INRIA and Cornell University, 2012

1 | Introduction

Rendering consists in the synthetic generation of digital pictures from a set of virtual geometric primitives, lights, materials and camera. Such picture is considered photo-realistic if it is generated following the principles of physics (Figure 1.1). Thus, a physical simulation of the light transport in this virtual scene has to be done [48]. This simulation involves an intricate combination of integrals [95], over all paths followed by light (called *light-paths*). This has to be solved numerically using either Monte-Carlo integration or kernel density estimation methods. Those methods are used in modern rendering softwareis [136]. Moreover, to include a large body of light phenomena, the simulation must provide various models for the interaction between light and the scene elements (referred as *scattering*), and between light and the camera.



3D scene *rendering* *post-processing* *digital picture*

Figure 1.1 *– In a photo-realistic rendering engine, an 3D scene is used as input of a lighting simulation following principles of physics. The resulting picture is post-processed and adjusted to be displayed on a media.*

Although light transport is well understood, a complete simulation can typically take days [84] for complex scattering models. We must keep in mind that realistic rendering is just another tool for artists to express their creativity: simulation times shouldn't be a restriction (Figure 1.2). Indeed, artists usually work iteratively by modifying a first draft until the desired emotion is reached. Our goal is to provide tools for artists that permit to produce accurate pictures, with a large range of light effects, while keeping rendering time short to permit a large number of iterations.

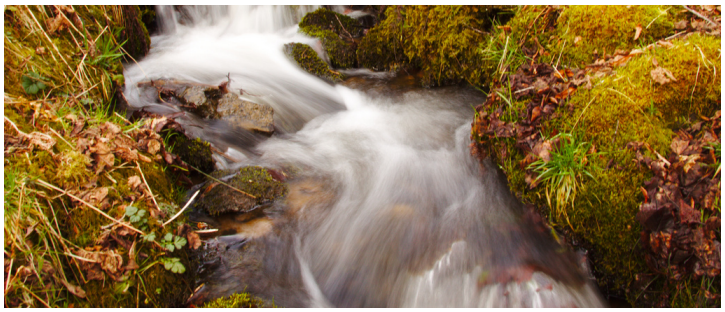
1.1 Motivation

We start our analysis from artistic photographs. We focus on three different effects: *depth-of-field*, *motion blur* and *scattering in fog* (See Figure 1.3):

Figure 1.2 *– Example of photo-realistic synthetic images. While the generation of the picture follows the laws of physics, it does not imply that the result will be realistic. Furthermore, the complexity of the simulation should not be a bottleneck for artists as what matters to them is creativity.*

(a) Depth-of-field

(b) Scattering in fog



(c) Motion blur

Figure 1.3 *– Light phenomenons available to artists in photography are numerous. Using a lens an artist can create a depth-of-field effect (a) to focus the attention on a particular detail. Light interaction with non opaque medium, such as fog, creates atmospheric scenes (b). Opening the camera shutter during a long time generates motion blur (c) and enforces the feeling of motion. Rendering synthetic images that mimic these phenomena remains a challenge.*

Depth-of-field results from the convergence of photons (light particles) from various places in the scene to a single point on the camera's sensor. This effect can be achieved with a lens and results in the blurring of elements that are not

on the *focal plane*. The focal plane is the region where there is no convergence, and one point on the sensor correspond to one point on the scene. This effect isolates the subject of the photograph from the background, or foreground (Figure 1.3a). While this effect results in blurry images (with less apparent geometrical details), it is more complicated to render an image with depth-of-field than an image without it. This is due to the fact that the area of simulation is bigger for a region out of the focal plane.

Motion blur results from the accumulation of light on a sensor with time. If an object moves during the exposition of the sensor to the light, a point on the object will scatter light to different positions of the sensor, blurring the appearance of the object along its motion (Figure 1.3c). Motion blur requires to simulate the evolution of the objects in time, and to accumulate the scattering of light with the moving objects on the sensor.

Scattering in fog diffuses the light in the volume covered by the fog, generating halos around lights (Figure 1.3b). While those halos blur the shape of the lights, scattering inside a foggy medium is more difficult than scattering on surfaces as it adds extra dimensions to the simulation. Indeed for scattering on opaque surfaces we only need to simulate the light transport between two dimensional surface elements while scattering inside a medium requires to simulate the light transport between three dimensional regions of space.

The three light phenomena reviewed share the tendency to blur the content of the image while being difficult to simulate. From a signal processing point of view, the image produced with a blurring phenomenon contains less information than the image produced without it. Thus, the quantity of information required to reconstruct the signal should be lower. Our goal is to theoretically identify blurry regions and to reconstruct the image with less computation than required for the same image without blur (from the signal processing perspective). To do so, we propose to study light phenomenons in a space where variations of the signal are naturally expressed: the *Fourier transform*.

1.2 Fourier Transforms

The *Fourier transform* is a tool to express a signal in terms of amplitude with respect to frequency (number of variation per unit cycle) rather than in terms of amplitude with respect to position. It defines an alternative space in which a signal can be studied (Figure 1.4). For example, if the Fourier transform of a spectrum is tight around the origin of the Fourier domain, the signal will not exhibit many variations (Figure 1.4, red inset). On the contrary, a Fourier transform that spreads in the Fourier domain will exhibit a lot of variations (Figure 1.4, green inset). Thus, the Fourier transform provides us a tool to depict blurry regions.

Another domain where using Fourier transforms are interesting is *numerical integration* (such as Gaussian quadrature or Monte-Carlo integrals). Numerical integration propose to approximate the solution of an integral using a discrete sum. The elements of the sum are called the *samples*. The quality of the

Figure 1.4 *– The Fourier transform of a signal depicts its variations. We illustrate this notion using the Lena image. We select portions of the image and display the Fourier transform in insets. Low frequency regions of the image are compacted around the origin of the Fourier domain while high frequency regions distribute in the Fourier domain.*

approximation is a function of the number of samples used. But, for the same number of samples, this quality varies for different integrand.

In fact, integration has an alternative formulation in the Fourier domain. There, the source of error in numerical integration is well understood [33, 46]. From the knowledge of the integrand spectrum, we can predict the required number of samples to obtain a perfect estimate of the integral. But, the integrand's spectrum is not known in practice.

1.3 Goals

The present work is motivated by the need to evaluate Fourier spectra. Indeed, the knowledge of the integrand's spectrum or of the image's spectrum allows to specify where the blur occurs or to define how many samples will be required to calculate an integral. We want to bring such knowledge to the area of rendering. But this has to be done for a complex set of lighting effects in order to be used by artists. We separate our goals into three categories:

1.3.1 Frequency Analysis of Light Transport

Frequency analysis of light transport is the area of computer graphics seeking to provide the knowledge of the integrand spectrum. This thesis is in the continuity of pioneering works on this topic [47, 164, 49, 51, 50]. Our goal is to enrich the set of effects analyzed. This is mandatory if we want our work to be used by artists in the future.

1.3.2 Denoising Applications

When the required number of samples can not be achieved, denoising algorithms can remove part of the remaining noise. Those algorithms are often driven by an estimate of the local variations. Frequency analysis can provide such knowledge. Our goal here is to provide algorithms to reconstruct smooth

regions from an uncomplete simulation in order to reduce the time needed to generate images.

1.3.3 Understanding Light Transport

Another goal of this thesis is to provide more understanding of the light transport process. Studying the Fourier spectrum allows us to understand how angular variations of the signal are blurred by diffuse reflection, how a lens affects the convergence of light on the sensor, or how the fog blurs the lights, in a different perspective than previously stated.

1.4 Contributions

This dissertation presents the following contributions:

- We enrich the analysis of Durand et al. [47] on frequency analysis of light transport. We define new operators such as volume scattering and absorption. We generalize previous operators, such as lens, reflection and occlusion (Chapter 3).
- We present the covariance analysis of light transport, a new analysis of the covariance of the local radiance's spectrum which is compatible with Monte-Carlo integration (Chapter 4.2).
- We present two data structures in the form of voxel grids to evaluate an approximation of the local occlusion of light by objects (Chapter 4.3).
- We present applications of the covariance matrix to validate our claim that frequency information can allow optimizations for ray-tracing algorithms (Chapter 5).

This dissertation is organized as follows: First, we will present the current state-of-the-art for generating photo-realistic images using light-path integrals (Chapter 2). Then, our contributions will be presented in three distinct chapters. In the first one (Chapter 3), we will coherently present the frequency analysis of light transport. This theoretical analysis will contain works we build upon as well as our contributions. The second chapter (Chapter 4) will study the tools provided to perform this analysis in a numerical integration context. We will present there the *covariance matrix*, a versatile tool developped to overcome the limitations of previously proposed tools. The last contribution chapter (Chapter 5) will present various algorithms to speed-up the rendering of photo-realistic images from the knowledge of frequency information.

2 | Theory of Light Transport

LIGHT transport simulation requires the definition of what light is, how interacts with matter (called *scattering*), and how it interacts with a sensor. A light transport model defines those elements. In this chapter, we will quickly review different models available (Section 2.1) and focus on the model used in physically based rendering: *Geometrical optics*. Then, from the integral definition of light transport, we will review the light-path integration algorithms (Chapter 2.2). Finally, we will review noise reduction methods for those integration algorithms (Chapter 2.3).

2.1 A Model of Light Transport for Computer Graphics

The goal of photo-realistic image synthesis is to estimate the amount of light on a virtual sensor. The corresponding physical quantity is the *radiance* (usually noted L). It is defined as the energy passing through a unit surface from directions inside a unit solid angle during a unit time period for a particular wavelength.

Estimating the radiance emitted by a light source on a sensor, after interacting with the world, requires a model for light-objects and light-sensor interactions. There exist several models to describe how light will interact with its environment:

- *Geometrical optics* assumes that light is composed of corpuscles: *photons*. Photons travel in the world along lines: *photons paths*. In a uniform medium (or in vacuum), the photons travel in straight lines. Photons can be absorbed and emitted by objects. The emission of a photon by a medium after an absorption is called scattering. Scattering is described statistically using a *phase function* (usually noted ρ) which describe how much of the absorbed photon is emitted in a particular direction.

- *Wave optics* models light as a wave. This model incorporates diffraction effects that geometrical optics cannot model for example.

- *Quantum electrodynamics* describes the interaction of light and matter using interactions between electron and photons in space and time. This model is derived from the *quantum physics* which describe physical phenomena at microscopic scale. This model incorporates Compton scattering (change of the photon's wavelength after a interaction) that wave optics cannot describe for example.

Geometrical optics is the model used in computer graphics. It is commonly accepted because of its simplicity (intersection of straight lines with geometrical objects) and because it captures most of the light effects humanly perceptible.

2.2 Algorithms for Light Transport Simulation

In the geometrical optics model, the estimation of how much power a surface or sensor receive is proportional to the density of photon paths arriving at this particular location.

2.2.1 Radiance estimation as an integration

The interaction of light with opaque media, is described by the *rendering equation* (Equation 2.1). This formulation of the rendering problem was proposed by Kajiya [95]:

$$\boxed{L(x,\omega) = \int_{\omega'} G(x,y)\rho(y,\omega,\omega')L(y,\omega')} \tag{2.1}$$

Where $L(x,y)$ is the radiance from position x to position y, $G(x,y)$ is called the geometrical term and accounts for occlusion, and for the relative geometry at position x and y. $\rho(y,\omega,\omega')$ is the scattering function at position y for an incoming direction ω and an outgoing direction ω'. For reflection scattering, the phase function is called BRDF (Bidirectional Reflectance Distribution Function) [125]. For refraction scattering, the phase function is called BTDF (Bidirectional Transmission Distribution Function). Because of its recursive definition, the solution to the rendering equation lies in the computation of a high dimensional function. In his thesis, Veach [175, Chapter 8] proposed an alternative formulation of this integral: *the light-paths formulation*. A light-path is a set of points on the surface of object, or inside participating medium, that form the virtual path that could be followed by photons. In this formulation, the integration of radiance arriving at a particular position in space is estimated by the integration of the density of light-paths (or photon paths) connecting this position with the light sources of the virtual 3D scene:

$$\boxed{L_j = \int_{l\in\Omega} f_j(l)\mathrm{d}\mu(l)} \tag{2.2}$$

Where L_j is the radiance value for pixel j, f_i is the function giving the radiance density for a particular light-path l in the set of all coherent light-paths with associated measure $\mathrm{d}\mu(l)$.

Light interaction inside non opaque volumes (e.g., smoke, clouds, water, skin) is described by the *Radiative Transfer Equation*, or RTE (See Ishimaru's monograph [82]):

$$(\vec{\omega}.\nabla + c(\mathbf{x}))L(\mathbf{x},\vec{\omega}) = b(\mathbf{x})\int_{\vec{\omega}'\in S^2} P(\vec{\omega},\vec{\omega}')L(\mathbf{x},\vec{\omega}')\mathrm{d}\vec{\omega} + Q(\mathbf{x},\vec{\omega}) \tag{2.3}$$

This differo-integral equation can also be expressed as an integral of light-paths [134]. It allows to combine the integration of participating media and

surface reflections in the same framework. Theoretical analysis showed that the RTE can be solved by using integration of discrete light-paths [32]. Continuous paths have to be used in the context of highly scattering medium. Tessendorf [172] proposed the use of *path integrals* to solve the RTE for strong forward scattering medium. It was later adapted to computer graphics by Premože et al. [138].

2.2.2 Integration methods for high-dimensional integrand

Classical numerical integration methods, like Gaussian quadrature, become intractable as the number of dimension grows (they converge in $N^{-\frac{1}{d}}$, where d is the number of dimensions and N the number of samples). The number of dimensions covered by the RTE is theoretically unbounded. Consequently, the computer graphics community prefers to use statistical integration tools that are independent to the number of dimensions.

In this section, we describe the two kind of statistical integration methods used in computer graphics: *Monte-Carlo* integration and *density estimation* methods.

2.2.2.1 Monte-Carlo integration

Monte-Carlo integration methods use principles of statistics to estimate the integral of a density function. The idea is to look at the integrand as a *probability density function* (noted PDF). Our aim is to evaluate its mean value, or *expected value*, which is proportional to the integral of the function. We can do it numerically using random evaluations of the PDF:

$$\boxed{L_j \simeq \frac{U}{N} \sum_{l_i} f_j(l_i)} \tag{2.4}$$

Where L_j is the radiance at pixel j, U is the area of integration (size of the domain of definition), f_j is the function giving the radiance contribution of light-path l_i to pixel j. N is the number of samples drawn (the l_i) uniformly over the domain of definition of f_j.

Metropolis [116] gives an historical perspective as well as an intuitive explanation of Monte-Carlo methods.

Monte-Carlo integration is independent from the number of dimensions for convergence, as all the dimensions are explored independently. The resulting variance reduction with respect to the number of samples is in $\frac{1}{\sqrt{N}}$ (where N is the number of samples).

2.2.2.2 Density estimation using kernel method

Another Monte-Carlo method to estimate the radiance at a given pixel is density estimation. This method is also known as kernel based density estimation.

Density estimation methods try to reconstruct a density function from a set of samples. At a position p, the value $f(p)$ is estimated using a kernel function K over the samples. In the original formulation, the density of samples has to be proportional to the function to be reconstructed. Recent works showed that is not mandatory if the sampling density is known and the function f can be

estimated at sample position. We use this method to reconstruct the radiance on the sensor. In this reconstruction, the integration of light-paths is implicit. (See Silverman's monograph for an introduction on density estimation [163]):

$$\boxed{f_j \simeq \frac{1}{Nh^d} \sum_{i=1}^{N} K_{h,j}(p_i)} \tag{2.5}$$

Where p_i are the samples used to reconstruct the density at position j, h is the window width, d the dimension of the space and $K_{h,j}(p_i)$ is the kernel function. The window should be estimated carefully as it will influence the resulting appearance. A big radius will blur the results while a small radius might not catch any samples leading to holes in the reconstruction.

2.2.3 Applications in Computer Graphics

To estimate light-path density, several algorithms have been proposed. We can categorize them into two categories: *Monte-Carlo* methods, and *kernel* methods. Beside the theoretical differences, those methods usually differ from where they perform the integration. *Monte-Carlo* methods estimate the radiance at the sensor location while kernel based methods estimate the radiance in the scene.

Methods presented here are often classified using *unbiased* and *convergent* classes. An *unbiased* algorithm provide the correct answer statistically. Averaging M results of independent run of the algorithm with N samples is equivalent to running the algorithm with $M \times N$ samples. We call the error to the solution the *variance*. A *convergent* algorithm converge towards the correct solution as the number of samples augment. The error to the solution is decomposed into a variance term and a *bias* term. This classification is interesting for a theoretical point of view. Practically speeching, this information is of little help and the question of how much samples to draw stays, no matter the class of the algorithm.

2.2.3.1 Monte-Carlo Integration

We review here the different kind of Monte-Carlo algorithms proposed until now in computer graphics. Those algorithms are often coupled with a light-path generation algorithm.

Unidirectional Path Tracing: Kajiya was the first to introduce the integral formulation of Equation 2.1. He proposed to solve this integration of the radiance using recursive ray-tracing [95] from the sensor to the light (referred by *eye-path*). The dual exploration scheme, *light-path* methods, follow the propagation of photons. Light-path tracing has been proposed in the context of radiosity textures where the connection to the eye is supposed to be diffuse [5].

Bidirectional Path Tracing: Eye-path and light-path tracing methods have their own strength. On one hand, light-path tracing is very good for creating specular paths from the light source, but fails to connect specular paths to a camera. On the other hand, eye-path tracing will perform well to generate specular paths from the eye while failing to connect specular path to the light. Bidirectional methods propose to alleviate these restrictions by combining those two methods.

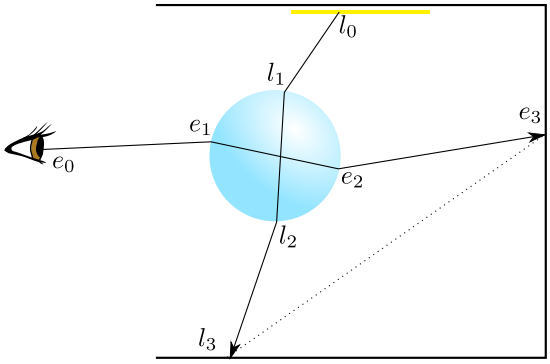


Figure 2.1 *– Using a bidirectional path-tracing method allows to generate complex light-paths like the path composed of a double refraction in the glass sphere from the light $l_0 \dots l_3$ and the double refraction in the glass from the camera $e_0 \dots e_3$. For that, we sample the light-path and the eye-path and connect the two based on visibility.*

The idea is to create concurrently both forward and backward paths and to connect them to create full light to eye paths (see Figure 2.1 for an example with a complex refraction). This method was first proposed by Heckbert [73] who stored the radiance from light-path into radiosity textures and used eye-path to evaluate the radiance at the sensor. Lafortune and Willems [103] and Veach and Guibas [176] published concurrently methods to produce light-paths from bi-directions

Metropolis Light Transport: Veach and Guibas [178] brought the Metropolis-Hasting [67] sampling method to Computer Graphics. This genetic algorithm generates light-paths as samples from mutations of a light-path seed (as illustrated with Figure 2.2). Mutations are accepted based on a defined probability density function, proportional to the radiance. The distribution of light-paths (after an infinite number of drawn samples) gives the energy distribution.

Defining mutations strategies for Metropolis is a challenge. As explained by Veach and Guibas, the set of possible mutations should allow ergodic changes. From a given light-path there should be a non zero probability of generating any other light-path (carrying energy toward the camera) from the set of possible mutations. Veach and Guibas [178] proposed a set of mutation based on typical cases (e.g., moving a diffuse point, moving a caustic path, etc). Another work on mutations was done by Kelemen et al. [96]. They looked at two kind of mutations (local and global ones) on a virtual hyper-cube.

Pauly et al. [135] extended the set of possible light-paths to be used by adding the theoretical foundation and mutation strategy for participating me-

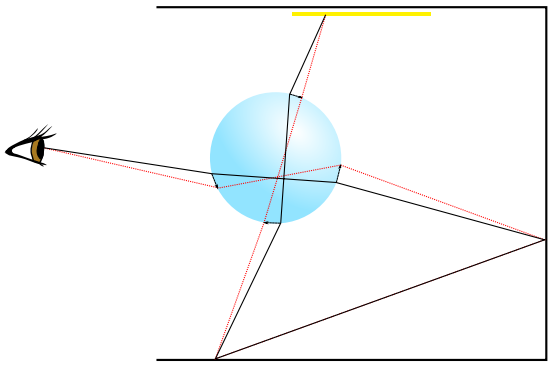


Figure 2.2 *– Metropolis light transport generate a Markov chain of mutation (in dotted red) from a seed light-path (in black). The radiance will be given per pixel by the density of mutated light-paths per pixel.*

dia. Segovia et al. [155] applied results from the applied statistic community on generating multiple candidates per mutation pass to further reduce the variance of the estimate.

Metropolis can also be used to generate light-paths for other integration methods and give the user some intuitive control (e.g., a maximum density per m^2) using the acceptance function. Segovia et al. [156], Fan et al. [54] or Chen et al. [23] used this technique to populate scene with either virtual point lights or photons.

Virtual Point Lights Virtual point lights (or VPL) are used to fake indirect illumination by adding more direct sources to the scene (Figure 2.3). This technique produces forward light-paths and store the resulting hit points on surfaces. Those hit points are then used as new light sources.

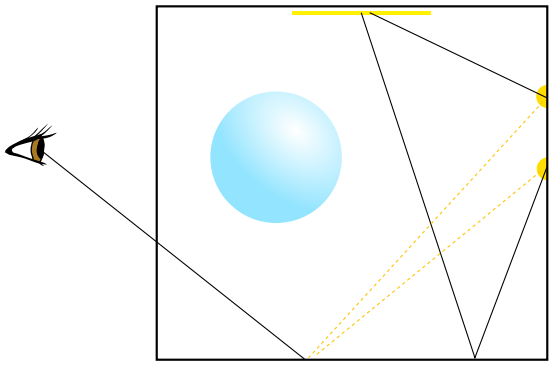


Figure 2.3 *– Virtual point lights (VPL) are created using the same first pass as photon mapping. The second pass differs as the algorithm integrate one bounce illumination using stored light-paths' end as primary sources.*

This idea was introduced by Keller [97] to bring global illumination effects into real-time rendering engines. This work was extended to be fully operational in a global illumination renderer [182, 183, 126, 127]. The number of

VPL per pixel is evaluated based on a perceptive metric. The same metric was used in an interactive setting using matrix clustering [70]. This solution is approximate and issues arise with near-specular glossy BRDFs. Techniques such as *Virtual Spherical Lights* [71] and combining global illumination from VPL with traced local illumination [38] overcome those limitations.

2.2.3.2 Density Estimation

Kernel methods differ mostly from the definition of the kernel to be used, or the domain in which the reconstruction is performed (either on surfaces, on the screen or in the volume). In this section, we review the different kernels type used in computer graphics. Then we will present derived methods such as iterative density estimation and splatting methods.

Photon Mapping Density estimation methods are commonly referred as *photon mapping* methods in the computer graphics community. Photon mapping is a two step algorithm: first, light-paths are sampled and the intersections with diffuse elements of the scene are stored in a data-structure: the *photon map*. Then eye-path are drawn and the intersections with diffuse surfaces are used to estimate radiance based on a local kernel method centered on the hit point.

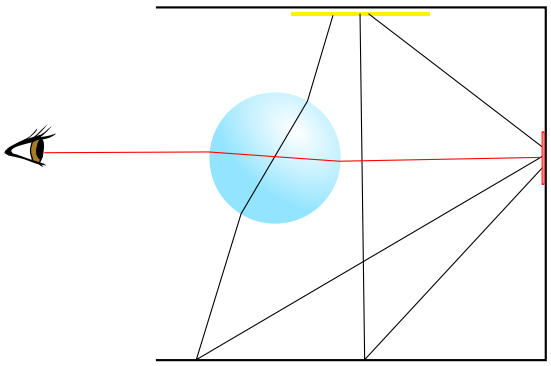


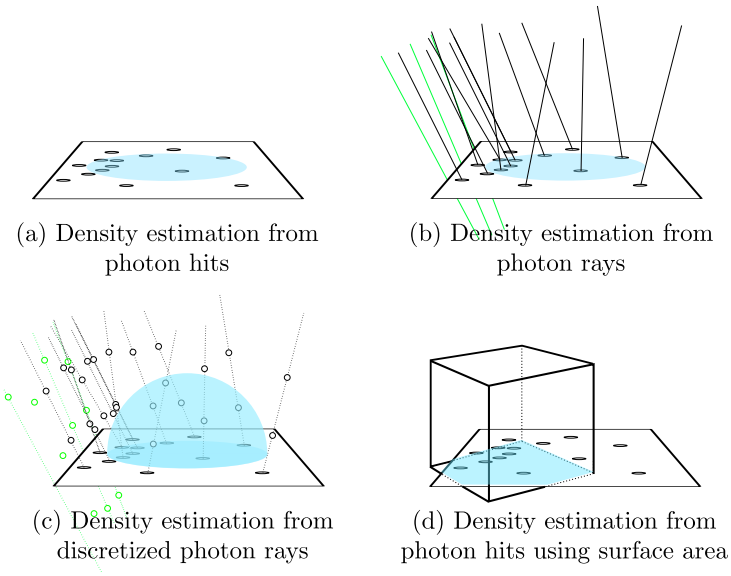
Figure 2.4 *– Photon mapping uses a two pass algorithm. First, light-paths (in black) are traced from the light sources and stored into a data structure: the photon map. Then eye-paths are traced from the camera until they hit a non-specular surface. Light-paths are created by accumulating the number of stored light-paths close to the end of the eye-path.*

Jensen and Christensen [91] described the photon mapping in the presented form. They added another pass to reduce some low frequency noise: the *final gathering*. Kernel estimation methods where pioneered (in Computer Graphics) by Chen et al. [25] who used density estimation to estimate caustics on diffuse surfaces, and Shirley et al. [161] who used density estimation for irradiance on adaptive meshes.

Lastra et al. [108] proposed a better estimate of the incoming radiance using photon rays instead of photon hits. This method is better at reconstructing sharp borders for example as it estimates the photon flux on a disc area. Evaluation of disc intersection was later improved by Havran et al. [68] who used

a lazy evaluated kd-tree to store rays. The evaluation of the intersection in Plücker coordinates due to the use of rays makes those techniques rather slow compared to traditional point density estimation. Zinke and Weber [190] discretized the photon ray into photons points in space and perform integration using half sphere rather than discs.

Another estimator was proposed by Hey and Purgathofer [79] who estimated the density using the geometry surface in a cubic kernel area. This method avoid the darkening of corner that arise with planar surface estimators. This method needs to account for occlusion during the selection of surfaces since the we are looking at a volume and no longer at a surface. Other methods used polygons to perform the density estimation on a fine tesselation [181], or to estimate a kernel shape adapted to the geometry [173]. Figure 2.5 sums up the different density estimators for surfaces.



(a) Density estimation from photon hits

(b) Density estimation from photon rays

(c) Density estimation from discretized photon rays

(d) Density estimation from photon hits using surface area

Figure 2.5 *– Here we review the different density estimators proposed for radiance estimation on surfaces. Jensen and Christensen [91] used photon hits in a k-nearest fashion (a). Lastra et al. [108] evaluated the light-field radiance on a planar disc using photon rays (b). Zinke and Weber [190] discretized photon rays to accelerate the density evaluation (c). Hey and Purgathofer [79] used the surfaces inside a cube perform the integration. Rays in green are not intersecting the surface but also contribute to the estimator.*

Photon mapping also works for participating media [92]. Instead of storing photons on surfaces, photons are also stored in the volume. Note that the kernel are 3D spheres in such a configuration. However, beam shaped kernels [86] can be preferred, as they increase the mean number of collected samples. Jarosz et al. [88] replaced photon points by *photon beams* for participating media in order to further accelerate convergence. This result in a gather in line space. Data structure in Plücker space can be used [168].

Photon Splatting The dual of the kernel density estimation method is to distribute the photon energy on the surface, or in the medium using individual kernels. This property is used to derive *photon splatting* techniques where the photons' kernels are rasterized on the screen [109, 17, 77, 89] or *reverse photon-mapping* [69, 77] where the photon energy is distributed onto eye-paths. With the power of graphics cards one obtains faster convergence, but the splat's size needs to adapt the geometry (e.g., occlusion, curvature). Part of this limitation can be addressed using a *progressive* scheme [89].

Progressive Photon Mapping *Progressive photon mapping* (originally proposed by Hachisuka et al. [65], then theoretically reformulated by Knaus and Zwicker [98]) leverage the storage cost of photons by breaking the algorithm into iterations. At each iteration, a small number of photons is sent into the scene and the density estimation is performed. Then, the photons are discarded and we begin another pass. In this new pass, we reduce all the kernels. This technique was pioneered by Boudet et al. [16] who iterated photon passes. But the kernels were not reduced after each pass leading to a biased result. Recent methods provided conditions on the radius reduction to satisfy the convergence [65, 98].

Progressive photon mapping has been extended to other kind of integration than surface density estimation. The effect of participating media [89, 98] can be integrated. Depth-of-field and motion blur effects are done using stochastic integration [63, 98].

Since the kernel is reduced at each pass, we don't need to adapt the kernel to the geometry. This error is converted into bias which decrease during the rendering time thanks to the iterative scheme and radius reduction.

2.2.3.3 Coupling Monte-Carlo and Density Estimation

Recent works proposed to couple the benefits of both methods [66, 59]. Bidirectional path-tracing is modified to accept connection using *vertex merging* which is defined with a close to photon mapping kernel gathering. The pure bidirectional and the vertex merging statistics are combined to produce a more robust estimator.

2.3 Noise reduction

While getting image faster requires a good implementation of these rendering methods, it is not the only place where we can achieve better performances (rendering quality per number of samples, or time). In this section, we will review classes of methods that decrease the noise present in the resulting integral with the same number of samples. This exposé is not complete as we review methods in relation with this thesis:

- *Importance sampling* (Section 2.3.1) draws more samples in regions of higher values.
- *Stratification* and *Adaptive sampling* (Section 2.3.2) adapt the number of samples to draw in regions with more variations. This is done in subspaces such as image space, lens space or time space.

- *Filtering methods* (Section 2.3.3) use already drawn samples and a filter algorithm to estimate a smoother results. Those methods are biased (compared to the precedent) as they use of a small number of samples (or an unconverged image) as input.
- *Caching methods* (Section 2.3.4) reuse previously computed samples for smoothly varying indirect illumination effects.
- *Kernel methods* (Section 2.3.5) have their own noise reductions methods. Either the data-structure or the kernel can be adapted to reduce noise.

2.3.1 Importance Sampling

Definition: Several methods have been proposed in the applied statistic community to accelerate the convergence of Monte-Carlo integrals. With *importance sampling* [94] the abscissa samples are not chosen uniformly in the integration domain, but they are drawn from an *importance function*.

Given that our samples are not drawn uniformly other the domain, the integration of radiance (Equation 2.4) becomes:

$$\boxed{L_j \simeq \frac{U}{N} \sum_{l_i} \frac{f_j(l_i)}{g_j(l_i)}} \quad \text{where } l_i \sim g_j \tag{2.6}$$

To keep the estimate unbiased, we need to put conditions on the importance function. For example, the importance function should always be strictly positive when the integrand is different from zero. This assumption allows to draw samples anywhere on the support of the integrand.

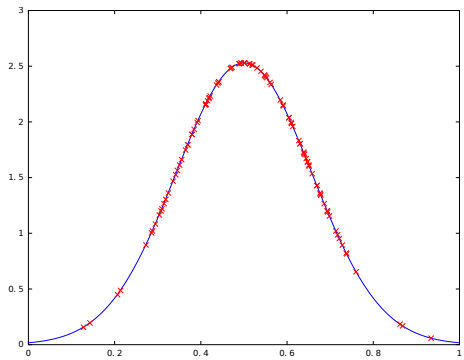


Figure 2.6 *– We illustrate the process of importance sampling a 1D distribution. From the target pdf (in blue), we draw samples (in red) those density is equal to the pdf.*

Generating light-path with importance sampling: Light-paths are created using importance sampling of the BRDF to be of a higher mean energy (Figure 2.7). Eye-paths can also use importance sampling of BRDF, but Veach and Guibas [177] showed that this could lead to a poor estimate. They combine multiple importance functions (such as BRDF importance and light importance) into the construction of the estimator and derive the corresponding

weights. Yet, using *multiple importance functions* can lead to poor performances when only one importance function decrease significantly the variance (as half the samples will significantly decrease the variance). Pajot et al. [133] adapted the ratio of samples assigned to a given importance function per pixel to overcome this limitation.

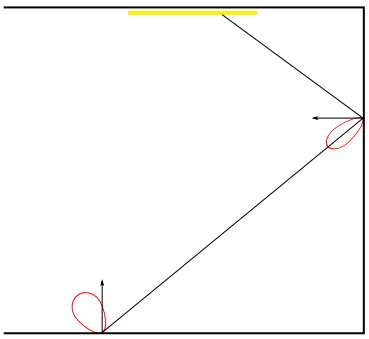


Figure 2.7 *– When creating a light-path, one can use importance function based on the BRDFs to create a light-path with a higher energy on average. The red lobes represent the angular importance function for the reflected direction. of the light-path in black.*

Importance sampling BRDF: Most of the analytical BRDF models provide a way to perform importance sampling of outgoing directions ([7, 184, 105] among others). Please refer to Montes Soldado and Ureña Almagro's survey [122] for a broader view on importance friendly BRDF models. When it is not possible, sampling patterns from another BRDF [8] or a quad-tree data structure [121] can be used. Acquired materials require an alternate representation such as Wavelets [106, 114, 28, 29], decompositions into lower dimensional factored terms [110], or rational functions [131].

Importance sampling light sources: We can importance samples distant illumination models (e.g., environment maps). We generate a set a point with distribution proportional to the intensity can be done using a quadrature rule [99], median cut algorithms [40, 180], or a hierarchical Penrose tiling [128]. Donikian et al. [45] importance sampled clustered light sources and tracked coherence within blocks of pixels.

Importance sampling product of functions: The target distribution can be achieved by a warping the space in which uniform samples are drawn. This can be done using product of Wavelets [27], or from hierarchical partition [30]. *Bidirectional Importance Sampling* [20, 185] and *Importance Resampling* [171] allows to importance sample the product of illumination and BRDF using rejection an resampling. Rousselle et al. [144] importance sample the triple product of distant illumination, BRDF and visibility by using tabulated maxima. The set of samples is refined using a local mean of the product. Lafortune and Willems [104] used a 5D cache of radiance to importance sample the product of visibility, light and BRDFs.

Importance sampling scattering: Several target density function can be used to reduce the noise for scattering of light in participating media depending on the lighting configuration. The angular domain can be importance sampled for lineic [126] or beam [127] light sources using an overestimate of the source contribution. Samples can be distributed along a ray based on the distance to the source or to get a constant angular density from the source point of view [100]. Scattering functions such as hair [113] can benefit from importance sampling [76]. Phase functions like Raleigh [57] have also been studied.

Nevertheless, building an importance function is a complicated task. As shown by Owen and Zhou [130], even though one might find a good approximation of the function to integrate and use it as the importance function, a close importance function can have an infinite asymptotic variance, leading to bad convergence rates.

2.3.2 Stratification and Adaptive Sampling

Definition: *Stratification* and *Adaptive sampling* reduce variance by separating the space of integration into strata based on the variation of the integrand. *Stratification* separates the domain into a set of N strata of equal variance and performs one computation (either evaluating the color of a pixel, or evaluating a sample) in each stratum. *Adaptive sampling* separates the domain into a grid (that can be adaptive) and adapts the number of samples inside each grid element based on the variance of the integrand.

Image space stratification: Mitchell [119] analyzed the convergence of stratification in image space. He reported that smooth regions converge in N^{-2}, while regions with a small number of edges converge in $N^{-3/2}$ and highly varying regions do not benefit from stratification.

Between traditional importance sampling and stratification: Agarwal et al.'s method [1] allows to remove some of the visibility variance using first stratification due to visibility, and then importance sampling based on intensity and stratum area.

Image space adaptive sampling: An estimate of the variance (or of the error) inside a pixel drives where to affect samples. Most of the algorithms use an iterative scheme where previous passes refine the estimate. Dippé and Wold [44] proposed an adaptive sampling algorithm, using the relative signal to noise ratio between two sampling rates. Simplified human visual systems, from the vision community, can be used to evaluate the perceptual differences between samples. It requires a basis to store samples, either a Discrete Cosine Transform [14], a Wavelet decomposition [15] or a Delaunay triangulation [55] of the image. Mitchell [118] used the perceptual metric of contrast. Rousselle et al. [145] used the Mean Square Error (MSE) estimate per pixel. Estimated variance from the samples can be used, extracted from a kD-tree [132], a Wavelet decomposition [129], or a block structure [45]. This variance can be enriched with depth information [22]. Sbert et al. [149] showed how to use information

theory to drive adaptive sampling. They used the notion of entropy of samples informations (such as radiance value, hit point's normal, ...) to estimate in which part of the scene information was missing for reconstruction.

Close to our work, bandwidth of the local Fourier spectrum or gradient informations have been used to derive a sampling rate per pixel [47, 141]. These methods rely on filtering collected regions.

Multidimensional adaptive sampling: Adaptive sampling can be done in higher space than the image space. Hachisuka et al. [64] performed adaptive sampling in the domain of image, lens and time to adaptively sample motion-blur and depth-of-field effects. Engelhardt and Dachsbacher [52] proposed to adapt samples along the eye ray in the context of single scattering integration. The samples are refined around the discontinuity of the visibility function to reduce variance in god rays.

Local Fourier spectrum analysis can be done in part of image, time, lens and visibility. This frequency estimate can be used to drive adaptive sampling [47, 164, 49, 51, 50].

Adaptive sampling requires the definition of remaining error to sample a region more than another. This is tightly linked to the variations of the integrand [119]. Most of the proposed methods rely on distance between samples (either absolute, perceptual, ...). This estimate can be of low reliability if the sampling is insufficient or the spatial distance between samples large [44]. Knowing how much the integrand varies locally around a given sample is of particular interest.

2.3.3 Filtering

Filtering methods are related to the area of *image's noise reduction*. Given an input image, we can reduce noise using a per pixel filter (function performing a weighted sum of neighboring pixels) that will suppress the noise from the image (high frequency, low intensity part of the signal) while preserving edges (high frequency, high intensity). Filtering methods can also be extended to be used in higher dimension space such as light-path space, or on parts of the integrand.

There exist a large domain for those techniques in the signal processing community (See Motwani et al. [123] or Buades et al. [19] for a survey on such techniques). But as stated by Mitchell and Netravali [120], the ray tracing community cannot use them directly as we filter a different source of noise.

We differentiate three kind of filtering methods based on a algorithmic criteria:

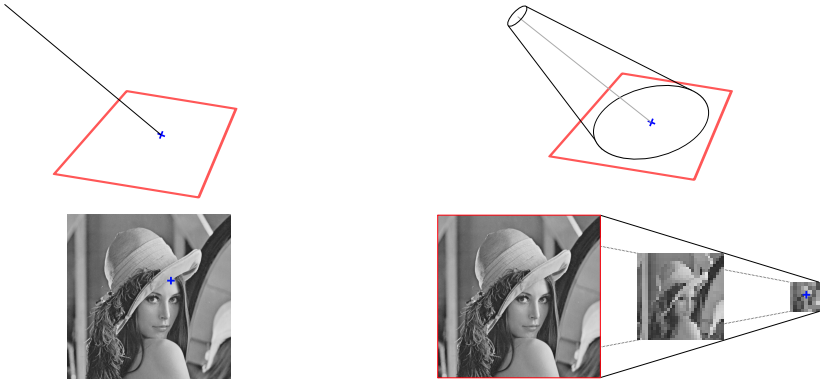
- *Prior methods* precompute filtered elements of the scene (such as textures, BRDFs, ...) and adapt the filter size at runtime based on the integration footprint.

- *Gathering methods* loop over the domain and accumulate samples information based on a filter function.

- *Splatting methods* loop over the samples and assign to each point in the domain a portion of its value based on a filter function.

2.3.3.1 Prior filtering methods

We present the idea of prior filtering methods (or *pre-filtering* methods). We do not present a complete survey of the field as our motivation is transverse to this domain (See Bruneton and Neyret's survey for more informations [18]). Our goal is to perform integration with no prior knowledge of the integrand, whereas prefiltering in its latest development pre-filters the complete scene [74]. Pre-filtering precomputes a filtered hierarchy, where higher levels correspond to larger filters, and evaluate the correct level during the evaluation of the value.



(a) Using a texture

(b) Using a texture with prefiltered hierarchy

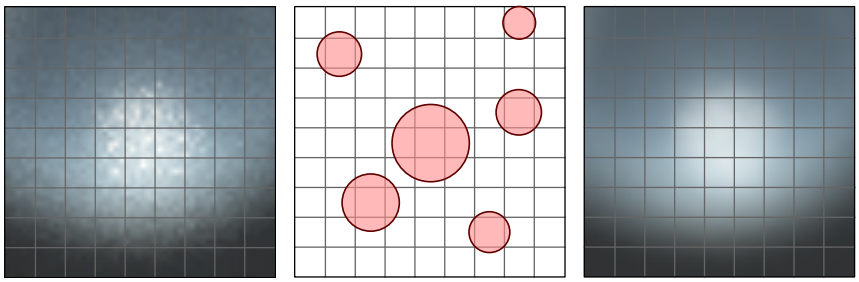
Figure 2.8 *– Pre-filtered texture can be used to avoid aliasing caused by high frequency textures for example. In this case, sampling the texture would require a fair amount of samples as it varies a lot (a). Using a pre-filtered hierarchy of the texture (b) allow to avoid the high variations of the texture using an already filtered texture in replacement. We need to have the ray's footprint information.*

Such methods require the knowledge of the footprint of a ray to evaluate the corresponding level of filtering. This information can be computed using cone tracing [3, 124] or ray differentials [81]. We will see later that frequency analysis can bring such information and we will the differences with other methods (Section 3.4).

2.3.3.2 Gathering filtering methods

Gathering methods work as follows (Figure 2.9): for all the pixels in the image, the algorithm estimates a reconstruction filter (or gathering kernel) and performs the weighted average of the samples inside the filter based on a distance from the pixel. For a given pixel $p \in P$, a kernel function $h : \mathbb{R}^+ \to \mathbb{R}^+$, a distance function from a pixel to a sample $d : P \times S \to \mathbb{R}^+$, and a set of samples S:

$$I_p = \frac{1}{H} \sum_s h(d(s,p))s \quad \text{where } H = \sum_s h(d(s,p)) \tag{2.7}$$

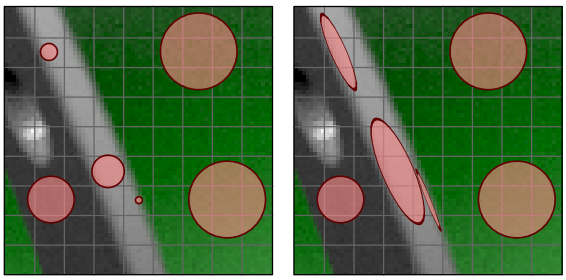


(a) Input samples **(b)** Reconstruction filters **(c)** Output image

Figure 2.9 *– Given a set of samples (that can be distributed in all pixels (a)), a gathering algorithm estimate a set of samples (b) and reconstruct a filtered version of the image using a weighted sum of the samples belonging to a filter (c).*

Isotropic filters rely on rotationally symmetric filter functions. Rousselle et al. [145], for example, proposed to use a fixed set of isotropic filters to estimate the variance per filter and then select the optimal filter and drive an adaptive sampling algorithm.

Isotropic filters are limited because of the anisotropy of the integrand. For example, they perform badly in presence of edges (Figure 2.10a). Anisotropic filters on the other hand are better to filter edges (Figure 2.10b).



(a) Isotropic filters **(b)** Anisotropic filters

Figure 2.10 *– Isotropic filters fail when the function to reconstruct is anisotropic. Given a discontinuity of the integrand, the isotropic filters (a) cover small areas near the discontinuities The anisotropic filters will perform better as their areas are bigger (b).*

Non-isotropic filters use a non isotropic filter kernel. The idea is to adapt the filter to the local variations of the function (Figure 2.10b). They use more parameters than the color only, for example using the depth buffer and the normal buffer in the distance metric of the filter. Dammertz et al. [36] used an à-trous wavelet transform with bilateral weights . Shirley et al. [162] filtered samples in a depth order with adaptive filter size per sample depending previously used filters.

Basis projection: The projection of a noisy input onto a basis which first components are smooth allow to filter out the noise by reducing the influence of "noisy" basis elements. For example, Meyer and Anderson [117], Chen et al. [26] used a PCA on the image and time domains allow to filter the noise from unconverged animations. Overbeck et al. [129] used a Wavelet decomposition to remove noise in the high frequency coefficients of the decomposition. Compressive sensing allows to reconstruct a signal from a sparse input by imposing sparsity on the basis used. It has been applied to denoising in the image plane using a Wavelet basis [157, 158].

Bilateral filters: Durand et al. [47] used the bandwidth (Section 4.1.1) of the local light-field to derive a bilateral filtering method in image space using a sparse set of samples. In a first step, they predict bandwidth for all pixels of the image. Then, they estimate a set of image samples with a density proportional to the bandwidth. Finally, they reconstruct the image using a bilateral filter which space width is proportional to the bandwidth prediction and takes depth and normal informations into account for reconstruction. Other uses of bilateral filters include *Bilateral upsampling* of a low resolution buffer of indirect illumination[143]. Correlation between the input random seeds and the generated samples can be used as another dimension of the bilateral filter [159].

Correlation between samples Ernst et al. [53] showed that, using Mitchell's filter on samples generated on the entire image, correlation effects could result in filters being worst than averaging. Due to the coherence of samples inside of a filter's footprint, the result is unlikely to contain such artifacts.

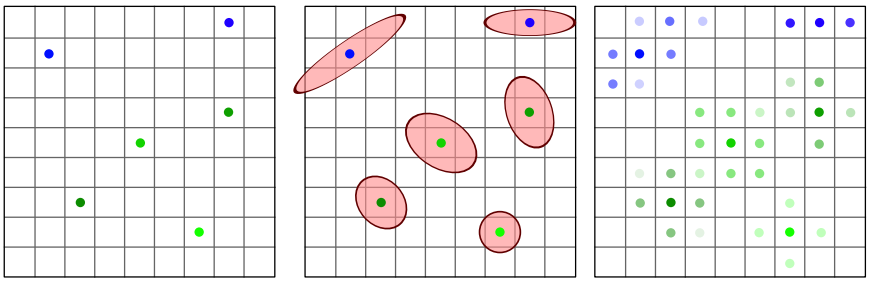
Gathering methods need to estimate the noise present in the reconstruction from the samples. This information has to be evaluated while preserving the function's variation (such as edges). As a result, those methods can blur too much the image.

2.3.3.3 Splatting filtering methods

Splatting methods are the dual of gathering methods. Instead of looping over all the pixels in the image, they iterate over all the samples and evaluate all the pixels covered by a filter centered on the sample (Figure 2.11).

Screen-space splatting: Rushmeier and Ward [146] splatted samples whose values are far from the local mean value. The more distant the sample value is from the mean, the wider the filter will be. While this method allows to diffuse outliers from unconverged simulation, it can also blur converged regions where the function has high variation.

Using edges information is useful in order to reconstruct an image from a set of sparse points [62, 10]. Those methods permit a reconstruction that is aware of the principal source of high frequency in an image. McCool [115] used an anisotropic diffusion method in screen space that preserved edges using depth and normal informations.



(a) Input samples **(b)** Samples with associated filters **(c)** Ouput sampling pattern

Figure 2.11 *– Given a set of input samples on an image (a), the algorithm estimate the splatting filter per sample (b) and reconstruction is done by calculating the integral of the filter over the pixel footprint (c).*

High dimensional splatting: Cline et al. [31] diffused energy from a given light-path using Metropolis mutations. Here, splatting is not done in screen-space, but in the light-path space. distributed effects such as depth-of-field, motion blur or soft shadows can be splatted (for direct illumination) in a higher-dimensional space [111]. The idea is to look at the space where the effect is defined and to splat samples along linear directions assuming a diffuse behaviour of the last bounce. In the case of depth-of-field, the space of image and lens is reconstructed from a sparse set of samples that are splatted along the first order direction. This method can be extended to indirect illumination [112].

Splatting methods need the knowledge of the splatting kernel size. It has to adapt to the signal bandwidth and to the geometry. This is often inferred from the samples [111, 112] or from other sources of information such as the depth-map [10]. Local variations analysis can provide the knowledge of the validity domain of a sample, and thus the size.

2.3.4 Cache methods

Cache methods (e.g., irradiance caching or radiance caching) use a data structure that sparsely stores indirect illumination information. Those methods are used to get smoother estimate of low frequency global illumination effects movie production renderers [170]. Cache methods are related to high dimensional filtering methods. We separate them from the mentioned section due to large body of work present on this matter.

Irradiance caching: Caching irradiance relies on the fact that irradiance is often smooth for mostly diffuse scenes. Cache entry consist of the color and intensity of the diffuse component of indirect illumination at the cache position. The cache require to define a bound of the irradiance gradient to adapt the density of cache elements. This can be done using the sphere lit approximation [187], or an estimate over a fixed number of samples over the

hemisphere [186]. Recently, Jarosz et al. [90] used the irradiance Hessian to adapt the density of caches and to improve reconstruction. A volumetric irradiance gradient for participating media can be derived [87, 142].

Radiance caching: Instead of storing a scalar value, radiance caching requires to store a distribution of radiance over the hemisphere (Usually using Spherical Harmonics). It also require a new interpolation method for the cache entries. Křivánek et al. [101] proposed to estimate the gradient assuming that the incoming radiance to the cache point was diffuse. This method neglect occlusion to evaluate the gradient and works for low frequency BRDFs [101, 102]. It is possible to account for the directionality of the signal using anisotropic gradients [78]. radiance's gradient with participating media was also derived by Jarosz et al. [87].

Gradient and Hessian estimations are at the core of the density evaluation of caches elements. Those methods rely on a good approximation of those quantities to optimize the size of the cache and its efficiency.

2.3.5 Noise reduction for kernel methods

Kernel based methods can benefit from more noise reduction methods than those previously mentioned. One can modify the density estimator by modifying the kernel or by modifying the distribution of photons.

Filtering kernels: is done by adapting the kernel size to the integrand variation. The bias introduced by the kernel can be estimated using a k-nearest search in the photon map [153] and then used to change the kernel size to minimize it. Schjøth et al. [150] used an anisotropic diffusion filter for the kernel. They derived an estimate of the irradiance gradient to adapt the kernel's shape. Spencer and Jones [166] used a hierarchical photon map to smooth low frequency content.

Filtering photons: Spencer and Jones [165] removed noise from the density estimation by performing a Lloyd relaxation step on the photon positions. The photons are then distributed with local equi-distance between them. This reduces the number of photon used in the k-nearest kernels, and filters the radiance by diffusion. Suykens and Willems [169] reduced the storage density by changing the weight of already stored photons when the photon map is locally too dense. Jakob et al. [85] fit the photons to a hierarchichal Gaussian mixture using an Expectation-Maximization algorithm.

Weber et al. [188] filtered the photon density in object and time space using bilateral filters. This removes flickering in low sampling density regions but does not account for angular variation of the reconstructed density (this could be problematic for near specular surfaces visible from the camera).

Noise reduction techniques aim to compensate the defaults of the k-nearest neighboors agregation method, or for the lack of information in low density regions. But modified kernels or filtered photon density have to adapt to the variation of the function to be reconstructed. Local variations of the fonction have to be evaluated.
Some algorithm try to reduce the number of samples in region of high energy but of low variation. We postulate that importance sampling the variations of the integrand rather than its intensity would be beneficial to photon mapping and avoid the need to use those algorihtms.

2.4 Conclusion

We saw that Radiative transport of photons can be modeled as an integration problem. To evaluate this integral, we can either use Monte-Carlo integration or kernel density estimation algorithms. Those algorithms rely on random samples: light-paths.

Those algorithms suffer from noise when the sample density is not sufficient to capture the variation of the integrand. We saw various noise reduction algorithms. *Importance sampling* alters the samples' distribution to favor high energy regions. *Adaptive* and *Stratification* algorithms distribute the number of samples according to the complexity of the integrand. Those methods require to estimate the variation based on a sparse set of samples as there is no notion of variation of the integrand nearby a sample. *Filtering* algorithms smooth the set of samples and are also derived from samples' statistic.

We aim to bring the integrand variation information to those algorithm. For that, we need to define a notion of the instantaneous variation of the integrand. We need to evaluate this information with from sample while doing raytracing. This information could then be used to adapt the sample density, derive filters adapted to the integrand, or importance sample the integrand to distribute samples in high variation regions. We also need this information to be anisotropic to permit better reconstructions when filtering.

Not much has been done in the field of filtering noise in participating media. Some images based techniques are general enough to handle it but they don't take the physical process into account. We know that scattering acts as diffusion process [167] and thus blurs the radiance.

In the following chapters, we will first introduce an analysis of local variations of the radiance function of a light-path in order to characterize variations of the final integrand using the *Fourier transform* (Chapter 3). Then, we will propose a new anisotropic descriptor of the local variance: the *covariance matrix* (Chapter 4). Finally we will propose applications of this tool to the two kind of integration methods we presented, *Monte-Carlo* and *kernel* methods (Chapter 5).

3 | Frequency Analysis of Light Transport

In previous chapters we showed the need for local variation information in the context of integration (Chapter 1). We reviewed radiance integration methods using light-paths and remaked that they can benefit from local variations information (Chapter 2). This chapter presents a theory to express local variations of radiance around a light-path sample.

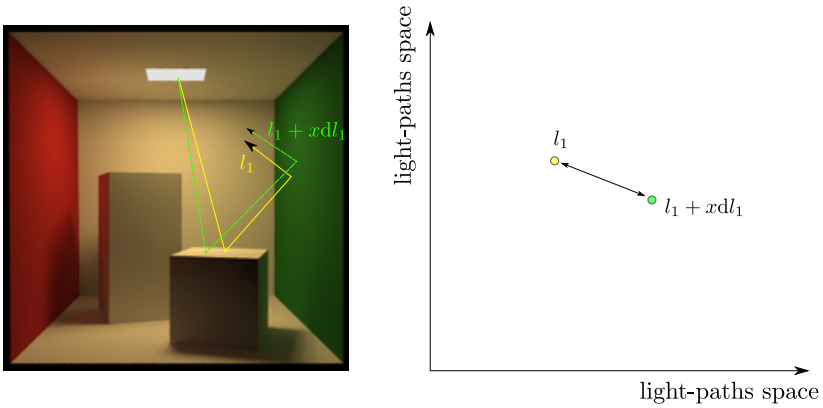


Figure 3.1 *– We want to express the variation of the radiance function L for small variations of its input argument l. For that, we need to define the local variation of l, dl and to look at the variation of L on this subdomain.*

Our goal is the following: given an input light-path l, we want to obtain the local variations of the radiance function $\mathrm{d}L(l + x\mathrm{d}l)$ (Figure 3.1). This theory builds on two elements:

- *Paraxial optics* defines a local neighborhood around a ray. We use it to express the local neigborhood of a light-path (Section 3.1).
- *Fourier transform* expresses a function using a dual one with arguments in a frequency domain (Section 3.2). We use it to express the variations of the radiance function in the paraxial domain.

In the first two sections, we will present *Paraxial optics* (Section 3.1) and the *Fourier transform* (Chapter 3.2), the required tools for our analysis. The

third section (Section 3.3) will present the frequency analysis of local radiance (introduced by Durand et al. [47]). The Fourier transform will be used to express the radiance function in the paraxial domain of rays along a light-path. In the last section (Section 3.4), we will compare this frequency analysis to other local variations analysis methods that use derivatives.

In this chapter we present the following contributions:

- We present the frequency analysis of light transport as a whole, in a 3D setting. Previous publications often presented the theory in a simpler 2D setting. But it hides some complex parts of the analysis such as equators alignment, or that convolution is along one angular dimension.
- We redefine some elements of the theory to make it more practical and more general. We redefine the analysis of *reflection*, *lens*, *occlusion* and *motion*.
- We add the analysis of *refraction* of light as well as *scattering* and *attenuation* in the context of participating media.

3.1 Paraxial Optic

A light-path is defined as a list of chained rays. We analyse a light-path in a space composed of neighbor rays close to the ray defining the light-path (we call this ray the *central ray*). Those rays are little perturbations of the central ray. This representation is close to the definition of ray differentials [81] and to Chen and Arvo's differential of a specular light-paths [24], but we do not allow the extension of variations domain. For example ray differentials propagate the angular extend of the derivative. If two highly curved surface are chained, the spanned differential angle can be large. Instead, we control the variation domain. We call *local light-field* (or *light-field function*) the radiance function defined in this local domain.

3.1.1 Parametrization of a Ray Neighborhood

For a given position on a light-path, our light-field function is defined over a 5D space, two dimensions define spatial variations, two dimensions define direction variations and one dimension corresponds to time variations. The direction components will be defined as angles measured from the central ray, along the principal directions of the tangent space (Figure 3.2). They define a spherical parametrization of angles, with the ray direction along the equatorial plane. In the following, we will denote $\vec{\delta r}$[1] a 5D vector which spatial component will be $\vec{\delta x} = (\delta x, \delta y)$, angular component will be $\vec{\delta\theta} = (\delta\theta, \delta\phi)$ and spatial component will be δt. This parametrization comes from paraxial optics theory [60].

$$\boxed{\vec{\delta r} = (\delta x, \delta y, \delta\theta, \delta\phi, \delta t)} \tag{3.1}$$

Figure 3.3, present our parametrization around a ray and an example of the radiance function with this parametrization.

[1] δr represent a small value, not a differential. While it might be confusing at first, it allows to separate small values from potentially large one in equations.

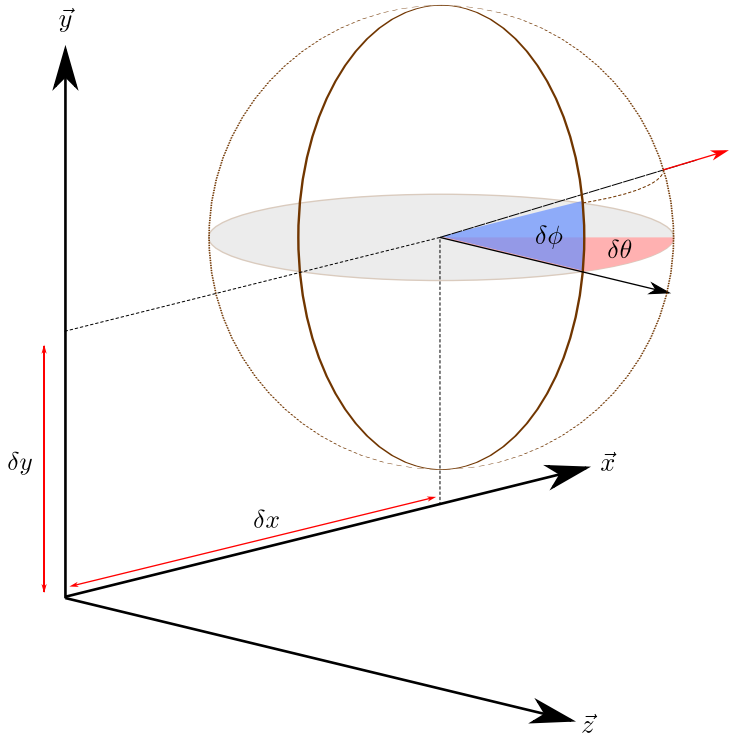


Figure 3.2 *– We parametrize the space of rays around a central ray (Z axis) with a 2D position in the tangent plane $(\delta x, \delta y)$ and a 2D angle in spherical coordinates $(\delta\theta, \delta\phi)$ (The grey plane on the spherical parametrization is the equatorial plane). We are interested in infinitesimal values of those parameters, the dot product with a ray and the central ray will always be close to one.*

3.1.2 Infinitesimal Analysis

We study infinitesimal variations of position δx, or angle $\delta\theta$, or time d around the central ray. It allows to perform first order simplifications in the analysis. One common simplification we use is the linearisation of trigonometric functions:

$$\boxed{\tan(\delta\theta) \simeq \delta\theta} \tag{3.2}$$

In such approximation, second order terms can be neglected. A correct way of mathematically writing this would be $f(u) = f(0) + u\frac{\delta f}{\delta u} + O(u^2)$, (the $O(u^2)$ regroup all the second order, and above, terms). Such approximation remains accurates for small values: $u \ll 1$. In this context, we do not distinguish local rays parametrized with angles $\delta\theta$ from those parametrized with tangent deviations from a unit distance $\delta u = \tan(\delta\theta)$.

3.2 Frequency Analysis and Fourier Space

Frequency analysis of functions was introduced by Fourier to solve the heat diffusion problem using what we call now Fourier series [56, p.159]. It was later extended, by Plancherel and Leffler [137] to general analysis of functions in L^2

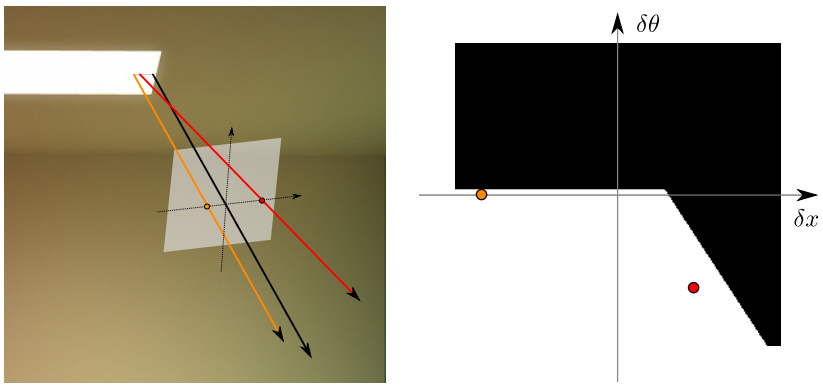


Figure 3.3 *– The local light-field parametrization is illustrated here. A given position $\vec{\delta r} = (\delta x, \delta y, \delta\theta, \delta\phi, \delta t)$ corresponds to a ray. We study the radiance going through local rays in this parametrization. We can only present two axes of the local space here.*

(what we call the *Fourier transform*), later to distributions by Schwartz [154], and even to probability density functions, where it is called the *characteristic functional* [174, Chapter IV.2]. In this thesis, we use functions with well defined Fourier transform (L^1 space or tempered distributions).

3.2.1 Fourier Transform

Given a function f (defined over argument $x \in \mathbb{R}^N$ called the primal space), the Fourier transform of f , noted $\mathcal{F}[f]$ (or $\hat{f}$), is a function with definition:

$$\boxed{\mathcal{F}[f](\vec{\mu}) = \int_{\vec{x}\in\mathbb{R}^N} f(\vec{x}) e^{-2i\pi\vec{\mu}\vec{x}} \mathrm{d}\vec{x}} \qquad \vec{\mu} \in \mathbb{R}^N \tag{3.3}$$

The space of in which the Fourier transforms of functions are defined is called the *Fourier space*, the *Fourier domain*, or the *dual space*. The input function of a Fourier transform is called the *primal function*. The Fourier transform of a function is sometimes called *frequency spectrum* of the primal function.

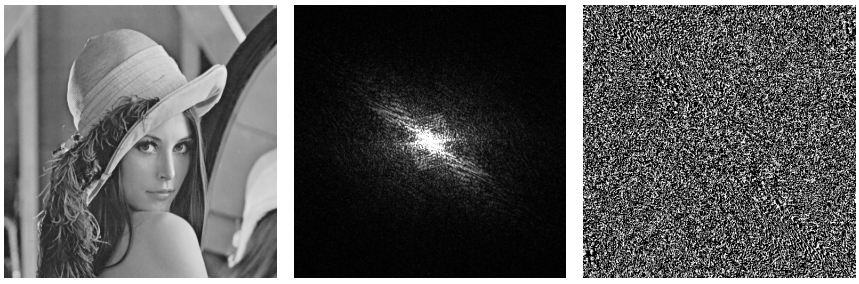
3.2.2 An example

Understanding Fourier transform is straightforward when we look at the resulting spectrum. The resulting function has values in the complex domain, which makes it difficult to analyze. We separate this complex signal into two components: the *amplitude* (Figure 3.4b), and the *phase* (Figure 3.4c).

$$\hat{f}(\vec{\mu}) = \rho(\vec{\mu}) e^{i\phi(\vec{\mu})} \tag{3.4}$$

Where $\rho : \mathbb{R}^N \to \mathbb{R}_+$ is the *amplitude* and $\phi : \mathbb{R}^N \to \mathbb{R}$ is the *phase*.

The amplitude, $\rho(\mu)$, represents the portion of energy from the primal function that correspond to the frequency μ. The phase, $\phi(\mu)$, represents the shift



(a) Input Lena image **(b)** Amplitude of the Fourier transform **(c)** Phase of the Fourier transform

Figure 3.4 *– We decompose an input 2D signal 3.4a into its amplitude (b) and phase (b) components as described in Equation 3.4. The amplitude corresponds to the energy assigned for a particular frequency. The phase corresponds to the shift of a particular frequency.*

associated with a given frequency μ. This can be intuitively explained using the transform of a cosine function:

$$f(x) = \cos(2\pi a x + p) \tag{3.5}$$

$$\mathcal{F}\big[f\big](\mu) = \frac{1}{2}\Big(\delta(\mu - a) + \delta(\mu + a)\Big)e^{2i\pi p} \tag{3.6}$$

The amplitude of the cosine's spectrum is: $\frac{1}{2}\Big(\delta(\mu - a) + \delta(\mu + a)\Big)$, and its phase is: $e^{2i\pi a}$. a regulates the frequency of the cosine (number of oscillations per cycle) and is correlated with the amplitude. p regulates the shift of the cosine and is correlated with the phase.

3.2.3 Properties

We present a short summary of different properties of the Fourier transform of functions. This section is not exhaustive, we will only cover properties that are of interest for our analysis. For more detailed examples, properties and theorems, please refer to [58] for example.

3.2.3.1 Linear Operations

The Fourier transform of a linear transformation is a linear transformation. Given a function f defined over a N-dimensional space $f : \mathbb{R}^N \to \mathbb{R}$. The Fourier transform of this function is:

$$\mathcal{F}\big[f\big](\vec{\mu}) = \int_{\vec{x}\in\mathbb{R}^N} f(\vec{x})e^{-2i\pi\vec{\mu}^T\vec{x}}\mathrm{d}\vec{x} \tag{3.7}$$

The Fourier transform of f under the linear transformation A of the input space (described by its matrix A), noted $f_A = f(A\vec{x})$, is:

$$\mathcal{F}\big[f_A\big](\vec{\mu}) = \int_{\vec{x}\in\mathbb{R}^N} f_A(\vec{x})e^{-2i\pi\vec{\mu}^T\vec{x}}\mathrm{d}\vec{x} \tag{3.8}$$

$$= \int_{\vec{x}\in\mathbb{R}^N} f(A\vec{x})e^{-2i\pi\vec{\mu}^T\vec{x}}\mathrm{d}\vec{x} \tag{3.9}$$

If the transformation matrix is invertible, we can proceed to a change of variable $\vec{y} = A\vec{x}$ in the integral. All the linear transformation matrices we will find in our analysis (rotation, symmetry, non-zero scale and shear) are indeed invertible.

$$\mathcal{F}\big[f_A\big](\vec{\mu}) = \int_{\vec{y}\in\mathbb{R}^N} f(\vec{y}) e^{-2i\pi\vec{\mu}^T A^{-1}\vec{y}} |A^{-1}| \mathrm{d}\vec{y} \tag{3.10}$$

Now we want to express this Fourier transform in terms of the Fourier transform of f (Equation 3.7). For that, we change the $\vec{\mu}$ variable to $\vec{\phi} = \vec{\mu}^T A^{-1}$. This gives us:

$$\boxed{\mathcal{F}\big[f_A\big](\vec{\mu}) = \frac{1}{|A|}\mathcal{F}\big[f\big](A^{-1^T}\vec{\mu})} \tag{3.11}$$

Where $|A|$ is the determinant of the matrix A and A^{-1^T}, the inverse transposed of A is the *comatrix*

An example: Rotation of the input signal We illustrate this property in the case of a rotation transformation. Given a 2D signal parametrized by (x, y) if we apply the rotation matrix defined by:

$$R_\alpha = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{3.12}$$

The dual operator on the frequency signal will be the same rotation (The inverse and the transpose of a rotation are the same). The Figure 3.5 shows the amplitude of the Fourier transform of the Lena picture with and without a rotation of the input space.

3.2.3.2 Product and Convolution

The product and convolution are symmetrical operations with respect to the Fourier transform. The Fourier transform of a product becomes a convolution in the Fourier space. Similarly, the Fourier transform of a convolution is a product in the Fourier space:

$$\boxed{\mathcal{F}\big[fg\big] = \mathcal{F}\big[f\big] \star \mathcal{F}\big[g\big]} \tag{3.13}$$

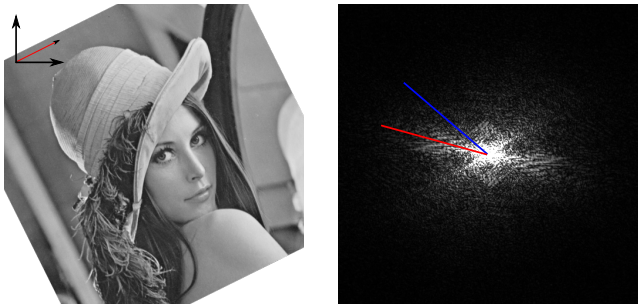
$$\boxed{\mathcal{F}\big[f \star g\big] = \mathcal{F}\big[f\big]\mathcal{F}\big[g\big]} \tag{3.14}$$

We give the proof for the convolution theorem. The multiplication theorem being symmetrical, its proof uses the same principles.

Proof: Given two functions $f(\vec{x})$ and $g(\vec{x})$ defined over $\mathbb{R}^N$, the convolution of those two functions $[f \star g](\vec{x})$ is:

$$[f \star g](\vec{x}) = \int_{\vec{y}\in\mathbb{R}^N} f(\vec{y}) g(\vec{x} - \vec{y}) \mathrm{d}\vec{y} \tag{3.15}$$

(a) Input Lena image and amplitude of its Fourier transform



(b) Rotated Lena image and amplitude of its Fourier transform

Figure 3.5 *– We apply the rotation matrix defined in Equation 3.12 to the Lena image and compute its Fourier transform. We display only the amplitude and highlight the first principal direction of the spectrum in red. Note that we removed the effect of the border discontinuity by multiplying the image by an isotropic cosine function.*

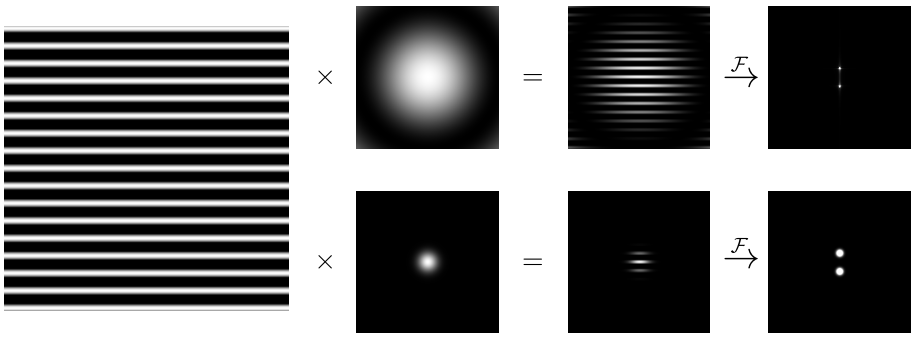
If we express the Fourier transform of the convolution using Equation 3.15, we obtain:

$$\begin{aligned} \mathcal{F}\big[f \star g\big](\vec{\mu}) &= \int_{\vec{x}} \int_{\vec{y}} f(\vec{y}) g(\vec{x}-\vec{y}) \mathrm{d}\vec{y} e^{-2i\pi\vec{\mu}\vec{x}} \mathrm{d}\vec{x} \\ &= \int_{\vec{y}} f(\vec{y}) \int_{\vec{x}} g(\vec{x}-\vec{y}) e^{-2i\pi\vec{\mu}\vec{x}} \mathrm{d}\vec{x} \mathrm{d}\vec{y} \end{aligned} \tag{3.16}$$

Using a change of variable $\vec{x'} = \vec{x} - \vec{y}$ and evaluating the inside integral, we obtain the following equations:

$$\begin{aligned} \mathcal{F}\big[f \star g\big](\vec{\mu}) &= \int_{\vec{y}} f(\vec{y}) \mathcal{F}\big[g\big](\vec{\mu}) e^{-2i\pi\vec{\mu}\vec{y}} \mathrm{d}\vec{y} \\ &= \mathcal{F}\big[f\big](\vec{\mu}) \mathcal{F}\big[g\big](\vec{\mu}) \end{aligned} \tag{3.17}$$

Example: We provide an example of the multiplication of an input signal with different window functions with increasing frequencies in Figure 3.6.



(a) input (b) window function (c) resulting function (d) Fourier amplitude

Figure 3.6 *– Multiplying a function with another is equivalent to the convolution of the Fourier transforms. In this example, an input function (a clamped cosine function (a)) is multiplied by a power of cosine with respect to the distance to the center (power of 2 on the top row, power of 50 on the bottom row (b)). The resulting functions (c) exhibit both the input function and the window functions characteristics. The resulting amplitudes (d) show that for a window function with a higher frequency, the input spectrum is convolved with a larger kernel (as higher frequencies spread more in the Fourier domain).*

3.2.3.3 Integration

Given a function $f(\vec{x})$ defined in a N-dimensional space, we express the Fourier transform of the partial integration of f along one of its dimension as:

$$\boxed{\mathcal{F}\big[\int_{x_i} f(\vec{x})\mathrm{d}x_i\big](\vec{\mu}) = \Big[\mathcal{F}\big[f\big](\vec{\mu})\Big]_{\mu_i=0}} \tag{3.18}$$

Where $\big[f(\vec{x})\big]_{x_i=0}$ is the $N-1$-dimensional function composed of the function f where the i^{th} component is zero. This property is often referred as the *slice theorem*.

The Fourier transform of a one dimensional function h for $\mu = 0$ is the integral along all dimensions of f. This term is called the *constant component* (or DC).

$$\begin{aligned}\mathcal{F}\big[h\big](0) &= \int_{x\in\mathbb{R}} h(x)e^{-2i\pi 0x}\mathrm{d}x \\ &= \int_{x\in\mathbb{R}} h(x)\mathrm{d}x \end{aligned} \tag{3.19}$$

3.2.4 Well defined space for Fourier transformation

In this section, we show that the Fourier transform can be performed on the paraxial domain defined in Section 3.1. The Fourier transform is defined for an infinite domain. But our analysis is correct on an infinitesimal portion of the space. To ensure the well defined property of the Fourier transform, we assume that the functions we are studying are defined on $\mathbb{R}^5$ but have values

on the region of analysis only. We separate the spatial and temporal case from the angular case:

Space and time: are kept local by multiplying the input primal function with a window being null outside a region of interest. This mathematical trick allows to keep the definition of the Fourier transform for our local function.

Angles: to keep spaces consistent with one another we consider the tangent space defined as $\delta u = \tan(\delta\theta) \simeq \delta\theta$ instead of angles. It allows to keep the same Fourier transform definition for both space, angle and time. Again, using a window function we keep our analysis local.

This use of window functions to keep the analysis to the first order introduces a bias in our analysis. The windowed results in a convolution in the Fourier domain of the input spectrum with the Fourier transform of the window (as seen in Figure 3.6) which increases the frequency content of the analyzed signal. But as we aim to estimate frequency conservatively this is not an issue.

3.2.5 Why not Another Transform ?

The Fourier transform is not the only possible way to analyze local variations of a function. In this section, we discuss several options that could be used instead of the Fourier transform.

3.2.5.1 Wavelets transform

The *Wavelets transform* is a frequency transform local in both space and frequency. Given a basis function $\phi(x)$, called the *mother wavelet*, we define the *child wavelet* function for a frequency band of $[1/a, 2/a]$ and a shift of b as:

$$\phi_{a,b}(x) = \frac{1}{\sqrt{a}}\phi\big(\frac{x-b}{a}\big)$$

The wavelets transform of the signal $f(x)$ is:

$$\mathcal{W}\big[f\big](a,b) = \int_{x\in\mathbb{R}} f(x)\phi_{a,b}(x)\mathrm{d}x$$

We do not use wavelets transform for two reasons:

- The wavelets transform has more dimensions than the input signal. The wavelets transform of a function defined over a five dimensions space has a ten dimensions domain of definition.
- There is no simple equivalent of the convolution theorem for the wavelets transform.

3.2.5.2 Short Time Fourier Transforms

The *short time Fourier transform* (STFT) is defined as the Fourier transform of a function with a sliding window:

$$\mathcal{TF}[f](t,\mu) = \int_{x\in\mathbb{R}} w(t-x)f(x)\mathrm{d}x$$

As for the wavelets transform, those transforms add dimension to the resulting spectrum. This is mandatory to locate the frequency content in the input domain. Our analysis is built upon light-path samples which already give a localization of the spectrum.

3.2.5.3 Hilbert-Huang Transform

The *Hilbert-Huang Transform* (HHT) is a decomposition of a signal into mode functions with the same number of extrema and zero crossings than the input signal. This transformation is done using and empirical search of the global and local extrema of a function to characterize an envelope for the signal. This empirical method cannot be fitted into our analysis.

3.2.5.4 Spherical-Harmonics basis

Spherical harmonics (SH) is a discrete frequency basis of functions defined over the sphere. They are widely used in computer graphics to store distance illumination such as environment maps. While SH could be used to express the frequency content in the angular domain, they are non-local. It would require a lot of coefficients to express the variation of radiance in a small angular domain.

3.2.5.5 Derivatives

The derivative of the local radiance function with respect to our local parametrization could be exploited as a descriptor of the radiance variation. Moreover, the derivative of a function is a purely instantaneous notion of variation. It is to note that using derivatives (or gradient) restrict the number of information about the local radiance function as the derivative is defined from the Fourier transform by:

$$\frac{\delta f}{\delta \mu_i}(\vec{0}) = \int_{\vec{\mu}\in\mathbb{R}^N} i\mu_i \mathcal{F}[f]\mathrm{d}\vec{\mu} \tag{3.20}$$

The resulting analysis would be to the first order in both space and variations.

Ramamoorthi et al. [141] derived operators defined by Durand et al. [47] for the gradient and Hessian of the local radiance function. We will see later (Section 4.2) a more detailed comparison of the two methods.

3.2.5.6 Summary

We summarize the different pros and cons for the presented transforms. We separate into four categories:

- *locality* is fulfilled if the transform is compatible with our small domain definition.

- *well-defined* is fulfilled if the transform can always be defined in our case of study (positive 5D functions).
- *small dimensionality* is fulfilled if the transform does not add extra dimensions into the analysis.
- *mathematical equivalent* is fulfilled if there exists equivalences in the primal for convolution, multiplication and integration.
- *richness* is fulfilled if the transform allows to recover all the information of the primal function.

The Fourier transform fulfill all the requirement for our analysis (Table 3.1). Derivatives is fulfill almost all of our requirement but cannot estimate all the variations of the local radiance function at once.

Transform	well-defined	local	dimensions	equivalents	richness
FT	√	√	√	√	√
WT	√	√			√
STFT	√	√		√	√
HHT	√		√		√
SH			√	√	√
Derivatives	√	√	√	√	

Table 3.1 *– We compare different transformations to find the best suited for our analysis. We need a transform that is well-defined for any 5D function with finite support, that supports local analysis, that does not add more dimensions to the analysis and that has mathematical equivalents for the operations we will study.*

3.3 Operators on the Light-Field Function

We define operators on the light-field function that represent the different operations it will undergo along the light-path. We note those operators using bold fonts. Those operators can be chained using a composition formula to represent the evolution of the light-field for given light-path (Figure 3.7):

$$LSDE \rightarrow \mathbf{L}_{(d_1,d_2)} \circ \mathbf{T}_{d1} \circ \mathbf{R}_{\rho_d} \circ \mathbf{T}_{d2} \circ \mathbf{O} \circ \mathbf{T}_{d3} \circ \mathbf{R}_{\rho_s} \circ \mathbf{T}_{d4}(l) \qquad (3.21)$$

Here $LSDE$ represents a light-path using Heckbert's notations [73]. $\mathbf{L}_{(d_1,d_2)}$, $\mathbf{T}_d$, $\mathbf{R}_\rho$ and $\mathbf{O}$ are functions we call operators that map a local light-field function into another after a physical process such as transport, visibility, reflection, etc. The composed operator takes as input the local light-field of the source l and it outputs the local light-field after all the operations.

This formulation is richer as it characterizes the local behaviour of light. The operations are arranged in a reversed order compared to the light-path notation to be consistent with a composition notation.

The list of operators we define for a light-path is given below. We illustrate some operators in Figure 3.8.

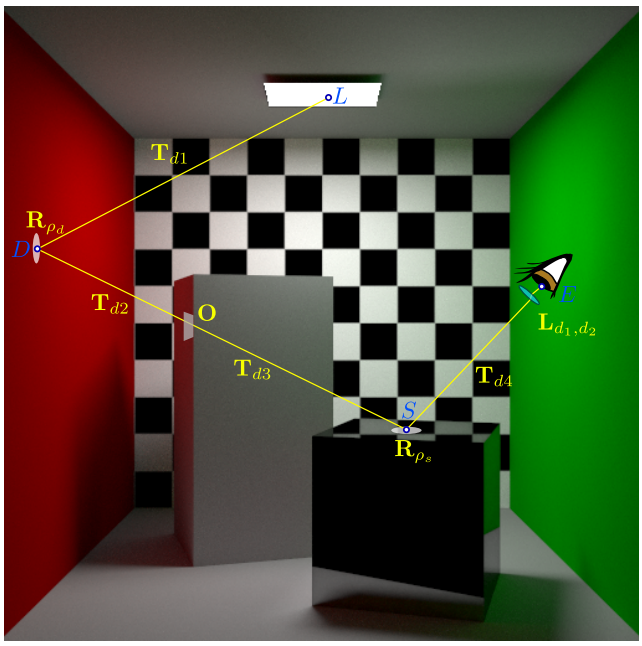


Figure 3.7 – *We illustrate our operator notation with the light-path connecting the light, the diffuse green wall, the specular box and the lens (in yellow). Heckbert's notation is represented in blue and specifies edges of the light-path. Our operators (in yellow) characterize richer effects such as partial occlusions of the local light-field, refraction in a lens, etc..*

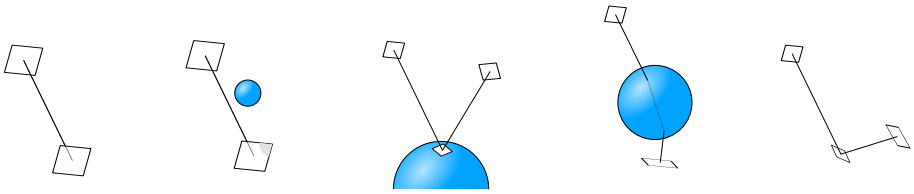


Figure 3.8 – *We expose different types of operators. Given an input and output parametrizations, the light in the local domain defined by the radiance transferd byall the variations of the input light-path that link the input parametrization to the output parametrization.*

- **Travel** defines the behaviour of radiance from different positions along the same ray spaced by d meters. We write this operator $\mathbf{T}_d$ (Section 3.3.1).

- **Occlusion** defines the behaviour of radiance when it travels close to an object while the central ray does not intersect it. We write this operator $\mathbf{O}$ (Section 3.3.2).

- **Reparametrizations** define how to express our radiance function in another local frame. Those operators are useful to express the radiance

on an object when an intersection occurs or when we want to take into account moving occluders. We write those operators $\mathbf{Rot}_\alpha$ and $\mathbf{P}_\alpha$ (Section 3.3.3).

- **Curvature reparametrization** defines the local radiance function on a curved surface based on the incoming local radiance function and on the local curvature matrix of the surface at the point of intersection. We write this operator $\mathbf{C}_K$ (Section 3.3.4.1).

- **Reflection, Transmission** define how the local radiance function will be affected by a reflection, or a refraction, on a virtually planar surface based on the BRDF or BTDF ρ. We write those operators $\mathbf{R}_\rho$ and $\mathbf{Tr}_\rho$ (Section 3.3.4.5 and Section 3.3.5).

- **Lens** defines how the local radiance function is modified when passing through a thin lens. We give two definitions for this operator: one that output irradiance on the sensor, the other that output radiance on the sensor (allowing to chain multiple lenses). We write those operators $\mathbf{L}_{(d_1,d_2)}$, (Section 3.3.6).

- **Participating media**'s effect on the local radiance function is handled by two operators: the attenuation operator $\mathbf{A}$ and the scattering operator (associated with phase function ρ) $\mathbf{S}_\rho$. (Section 3.3.7).

- **Motion** allows to track the time variations of the local radiance function by projecting it onto the static frame of a moving object. On this static frame, we can apply all the operators described before and express moving effects as a projection into the static frame. We project the input ligth-field, apply a set of static operators and then project the result back onto the frame of the light-path. We write this operator $\mathbf{M}_{\vec{v},\vec{r}}$ (Section 3.3.8).

3.3.1 Travel in free space

We assume that light travels along straight lines. Thus we avoid varying indices or relativistic effects. When we want to express a light-field after a travel distance of d meters, we need to take into account that rays that are not parallel to the central ray will not intersect the tangent plane at the same position after travel. Figure 3.9 expresses in 2D this deviation for a particular position $\delta x, \delta\theta$ and a traveling distance of d meters.

The local light-field after a travel of d meters can be expressed using the input local light-field with a transformation of the domain of definition:

$$\boxed{\mathbf{T}_d\big(l\big)(\vec{\delta x},\vec{\delta\theta}) = l(\vec{\delta x} - d\tan(\vec{\delta\theta}),\vec{\delta\theta})} \tag{3.22}$$

Given our infinitesimal assumption, a first order approximation of Equation 3.22 is:

$$\boxed{\mathbf{T}_d\big(l\big)(\vec{\delta x},\vec{\delta\theta}) \simeq l(\vec{\delta x} - d\vec{\delta\theta},\vec{\delta\theta})} \tag{3.23}$$

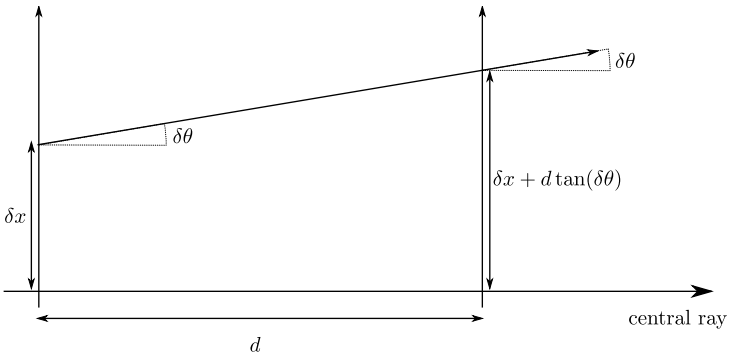


Figure 3.9 *– Given a ray with coordinates $(\delta x, \delta\theta)$ (we take 2D coordinates for the clarity of the figure) with respect to the central ray. The coordinate of this ray after a travel in a "free space" of d meters is $(\delta x + d\tan(\delta\theta), \delta\theta)$. Thus the radiance function after a travel of d meters l_d value at coordinate $(\delta x + d\tan(\delta\theta), \delta\theta)$ is the value of the radiance function before the travel at coordinate $(\delta x, \delta\theta)$.*

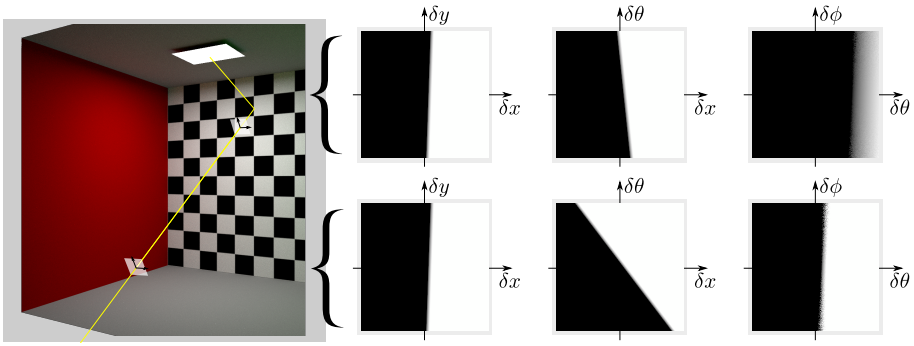


Figure 3.10 *– We show here the local light-field function at two different positions along a ray. Since the local light-field is 5 dimensional, we only display slices of this function (the other components are set to zero).*

We express this linear transformation using a matrix A. The matrix of the shear is given implicitly by Equation 3.23:

$$A = \begin{pmatrix} 1 & 0 & -d & 0 & 0 \\ 0 & 1 & 0 & -d & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.24}$$

The matrix used for the Fourier transform is defined as the comatrix of A (Equation 3.11):

$$A^{-1^T} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ d & 0 & 1 & 0 & 0 \\ 0 & d & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.25}$$

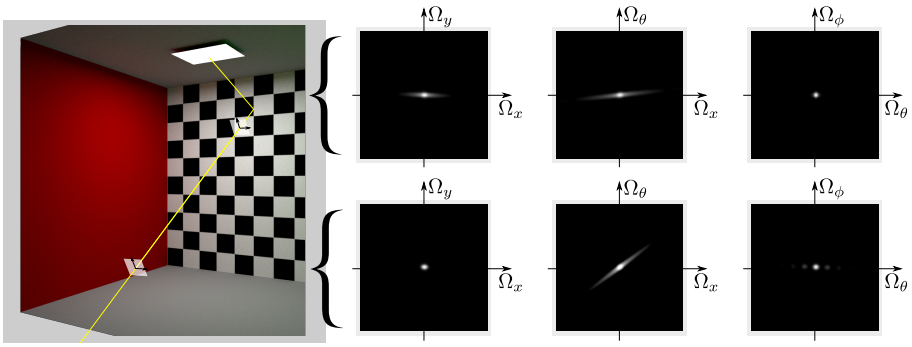


Figure 3.11 *– We display slices of the Fourier amplitude. The shear effect is visible on the spatio-angular slice. Indirect effect of the shear is visible in the spatial slice as a part of the energy go to the angular domain.*

This shear in the primal space is expressed in the frequency domain by a shear but with a symmetry on the dimensions it is applied to (Equation 3.23). The travel transfers energy from the spatial domain to the angular domain:

$$\boxed{\mathbf{T}_d\big(\hat{l}\big)(\vec{\Omega_x},\vec{\Omega_\theta}) \simeq \hat{l}(\vec{\Omega_x},\vec{\Omega_\theta}+d\vec{\Omega_x})} \tag{3.26}$$

Figure 3.11 present the effect of travel on the amplitude of the local light-field spectrum. The shear effect is noticeable in the angular slice. The effect of energy transport from the spatial domain to the angular domain is visible in the spatial slice (The tail of the distribution shrinks).

3.3.2 Partial Occlusion

As we are looking at a small neighborhood around a ray, we need to keep track of partial occlusion in this space by the geometry. A close solid object will occlude part of the light-field (Figure 3.12).

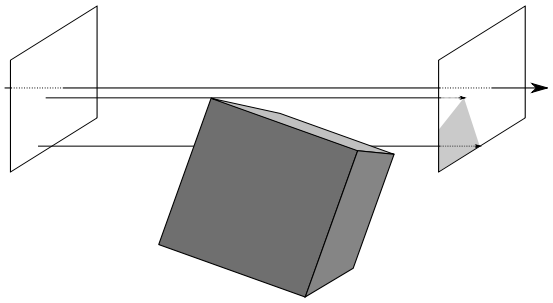


Figure 3.12 *– Given that we are not looking at a punctual position on the light-path domain, we have to look at the visibility function in this domain. Rays in this neighborhood can intersect the geometry. In that case, the radiance function has to be attenuated by the visibility function.*

The general equation for occlusion is:

$$\boxed{\mathbf{O}\big(l\big)(\vec{\delta x},\vec{\delta\theta}) = v(\vec{\delta x},\vec{\delta\theta})\; l(\vec{\delta x},\vec{\delta\theta})} \tag{3.27}$$

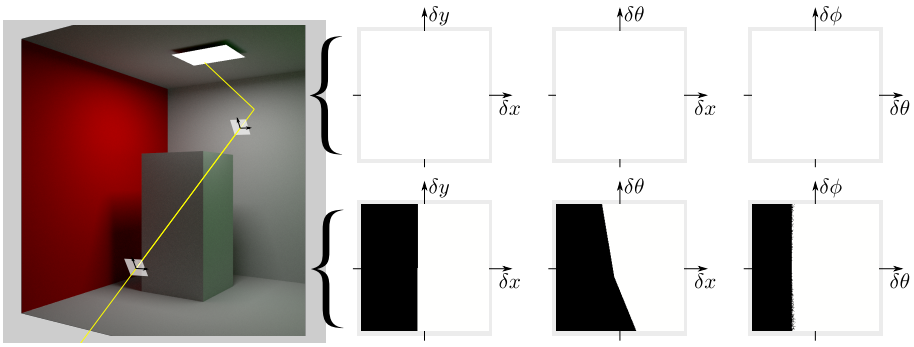


Figure 3.13 *– Planar approximation of the occluder does not capture the reality correctly. This figure shows the angular effect of occlusion, and the correlation it creates between space and angles. A constant input local light-field is occluded by the box.*

Where $v(\vec{\delta x}, \vec{\delta\theta})$ is the visibility function. This function is one if the position $\vec{\delta r}$ is not occluded and zero if it is. For non-opaque occluders, the visiblity function takes values between zero and one.

Planar occluder: In the initial formulation of Durand et al., the occlusion is modeled by the multiplication of the light-field with a visibility function in the spatial domain. This approach assumes that occluder is planar. Instead, we define this process as a windowing of the signal where the window restricts the signal in a region where it is unoccluded.

While these two visions have the same mathematical expression, they denote different views. The first one looks at the light-field evolution along the ray while the second one estimates the region of the light-field where the analysis is still meaningful.

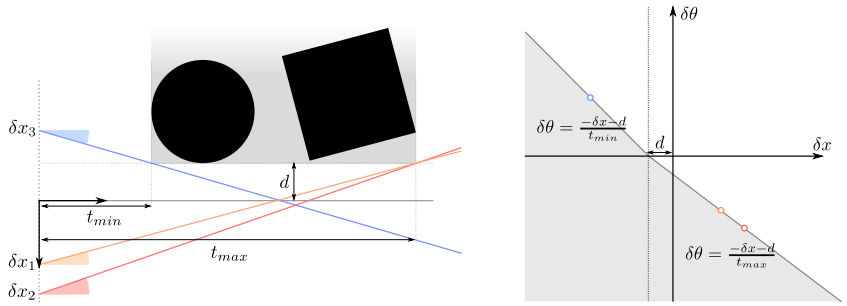
$$\boxed{\mathbf{O}\big(l\big)(\vec{\delta x}, \vec{\delta\theta}) = v(\vec{\delta x})\; l(\vec{\delta x}, \vec{\delta\theta})} \tag{3.28}$$

Occlusion is defined in the primal as the product of the light-field with a visibility function (or a window function). The equivalent operator in Fourier is a convolution. The convolution of the light-field spectrum with the Fourier transform of the visibility function extends the light-field spectrum along the discontinuity direction ($\otimes_{(x,y)}$ is the convolution in the spatial domain only):

$$\boxed{\mathbf{O}\big(\hat{l}\big)(\vec{\Omega_x}, \vec{\Omega_\theta}) = \big(\hat{v} \otimes_{(x,y)} \hat{l}\big)(\vec{\Omega_x}, \vec{\Omega_\theta})} \tag{3.29}$$

The planar approximation is incorrect even to the first order as it misses the effect of occlusion in the angular domain and the resulting correlation between the spatial and angular domain (Figure 3.13).

Non planar occluder: Ramamoorthi et al. [141] derived a non-planar approximation for curved occluders. Lanman et al. [107] and Egan et al. [51, 50] looked at the wedge signal defined by the min and max distant occluders (assuming that each occluder is planar). We propose here an occluder approximation that accounts for non-planar occluders. Our approximation neglect



(a) Configuration

(b) Approximate visibility function

Figure 3.14 *– We model the correlation between space and angles using the following configuration (a). Given a set of occluders, we analyse the resulting occlusion bounding box. Close to Egan et al. [51] analysis, we define the minimum and maximum distance of occlusion and look at the resulting visibility function. We obtain a piece-wise rotated half-plane occluder (b).*

occluders' curvature and leads to a more conservative estimate than the curved approximation.

Given a set of occluders, and a ray passing next to them, we define the visibility window using the minimum distance d from the occluders to the ray and the minimum and maximum distances, t_{min} and t_{max} from the beginning of the ray to the occluders (Figure 3.14(a)).

The visibility window is a piece-wise rotated half-plane function (Figure 3.14(b)). We found the same type of visibility function in our experimentation (Figure 3.13).

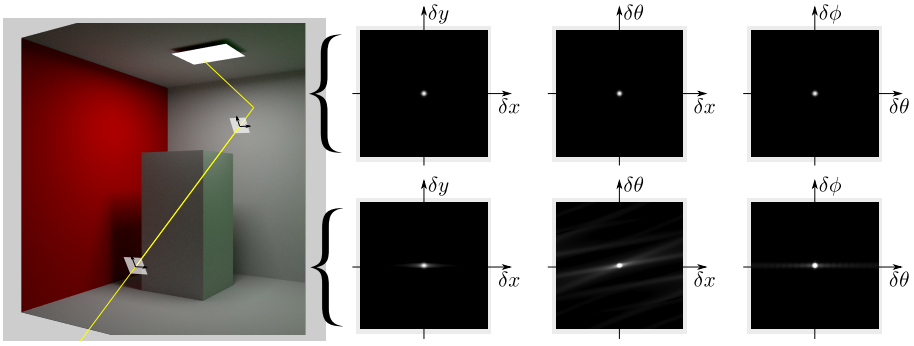


Figure 3.15 *– The effect of the partial occlusion of the local light-field travelling near the grey box is analysed in the Fourier domain. The occlusion creates high frequencies in the spatial domain and in the angular domain. The size and depth of the occluder affects the correlation between the spatial and angular domains.*

We model the visibility function in the primal with the multiplication of two rotated 1D sign functions. A 1D rotated sign function has the following

definition:

$$r_{t,d}(\delta x, \delta\theta) = \begin{array}{l} 1 \text{ if } \delta x + t\delta y > -d \\ 0 \text{ else} \end{array}$$

The resulting visibility function of a non-planar occluder is the product of the two rotated sign functions shifted by the distance to the occluder:

$$v(\delta x, \delta\theta) = r_{t_{min},d}(\delta x, \delta\theta) r_{t_{max},d}(\delta x, \delta\theta)$$

The resulting Fourier spectrum is the convolution of the individual Fourier spectrum of the rotated sign functions:

$$\hat{v}(\Omega_x, \Omega_\theta) = \hat{r}_{t_{min},d}(\Omega_x, \Omega_\theta) \otimes \hat{r}_{t_{max},d}(\Omega_x, \Omega_\theta)$$

If $t_{min} \neq t_{max}$ we get the following formula for the spectrum of the visibility function (See Appendix A.1 for complete derivation):

$$\hat{r}_{t_{max},d} \otimes \hat{r}_{t_{min},d}(\Omega_x, \Omega_\theta) = \frac{e^{2i\pi d\Omega_x}}{4\pi^2 \frac{(t_{max}\Omega_x + \Omega_\theta)(t_{min}\Omega_x + \Omega_\theta)}{(t_{max} - t_{min})^2}} \tag{3.30}$$

The resulting spectrum is a wedge function 3.16. Contrary to Lanman et al. [107] we do not need to estimate the occluder spectrum by multiple slices. The spread of the minimum and maximum distance is similar to Egan et al. analysis. It validates the use of the minimum and maximum slices to approximate occluders.

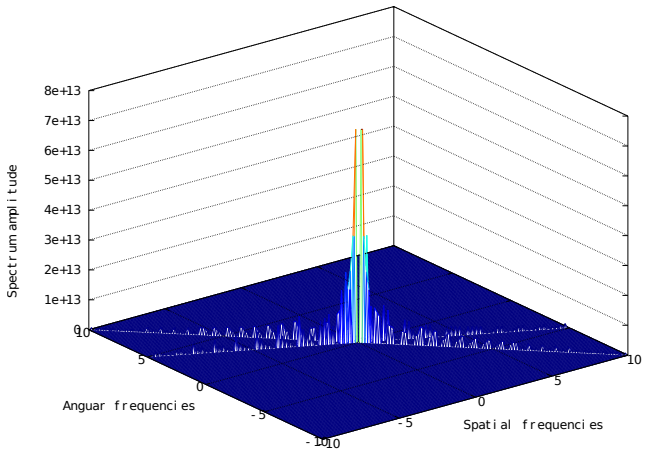


Figure 3.16 *– The non-planar occluder approximation is a wedge function. It explains why the planar slice approximation of occluders works [107, 51, 50]. Furthermore, we only need to provide the first and last slice of the occluder to obtain a decent estimate of the occlusion. For this figure, we used* $t_{min} = 50cm$ *and* $t_{max} = 1m$*. We used* $d = 0$ *to have a real valued spectrum.*

3.3.3 Reparametrization on another plane

Any operation involving the intersection with an object (reflection, refraction) will be described in its local frame. Thus we need to reparametrize our light-field function in the local frame of the object.

Reparametrizing to a new local frame is a three steps process. First, we rotate the frame of the light-field so that its X axis is along the intersection between the tangent plane of the light-field and the tangent plane of the object at the point of intersection (Figure 3.17). Second, we express the local light-field on the tangent plane using a projection of the Y axis onto the local tangent plane (Figure 3.18). Finally, we perform another rotation to align the light-field X and Y axis to the local X_p and Y_p axis.

3.3.3.1 Rotation

The rotation of a local X, Y plane of α radians around the central ray Z (Figure 3.17)) is written:

$$\boxed{\mathbf{Rot}_\alpha\big(l\big)(\vec{\delta x}, \vec{\delta\theta}) = l(R_\alpha^T \vec{\delta x}, R_\alpha^T \vec{\delta\theta})} \tag{3.31}$$

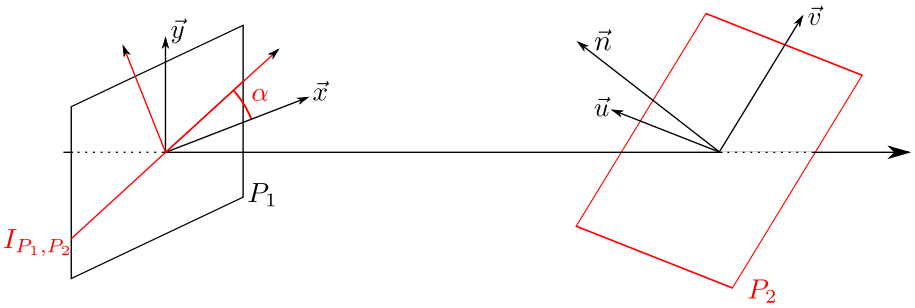


Figure 3.17 *– The first step of a projection of the frame of our local light-field (noted P_1) onto the local frame of an object (noted P_2) is to align the $\vec{x}$ vector on the plane P_2. This is done by a rotation of angle α, where α is the angle between $\vec{x}$ and the intersection of P_1 and P_2 (noted I_{P_1,P_2} in this figure).*

This linear transformation is formulated into one 5×5 matrix on the input parameters of the local light-field function:

$$R_\alpha = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 & 0 \\ 0 & 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.32}$$

This matrix is a compound of rotation matrices, its comatrix will be equal to the original matrix, ${R_\alpha^{-1}}^T = R_\alpha$ (Equation 3.11). This property allows to write the Fourier equivalent operator is the same fashion:

$$\boxed{\mathbf{Rot}_\alpha\big(\hat{l}\big)(\vec{\Omega_x}, \vec{\Omega_\theta}) = \hat{l}(R_\alpha^T \vec{\Omega_x}, R_\alpha^T \vec{\Omega_\theta})} \tag{3.33}$$

3.3.3.2 Projection

Projection aligns the last dimensions of the two local light-fields. This operator is a travel which distance varies with respect to the position on the input local light-field. (Figure 3.18). To compensate for the non alignment, half of the shear is in the light propagation direction and the other in the opposite direction.

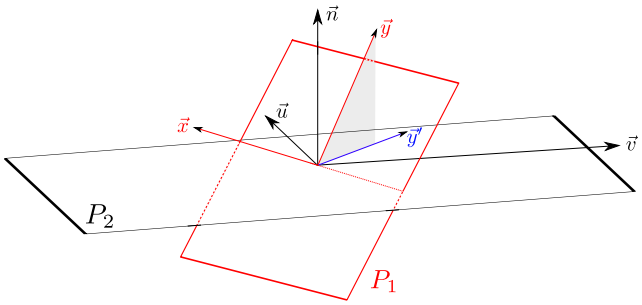


Figure 3.18 *– The second step of a projection of the frame of our local light-field (noted P_1) onto the local frame of an object (noted P_2) is to project the 'vertical' part of the signal (along the $\vec{y}$ axis and ϕ angle) onto P_2. The linear approximation of this operation is a scale of the spatial component of $\frac{1}{\vec{y}.\vec{n}}$.*

We approximate this transport using a scaling of the spatial dimension (Formal proof can be found in Appendix A.2). The resulting light-field on the surface is:

$$\boxed{\mathbf{P}_\alpha\big(l\big)(\delta x, \delta y, \delta\theta, \delta\phi) = l(\delta x, \frac{\delta y}{\cos(\alpha)}, \delta\theta, \delta\phi)} \tag{3.34}$$

This scaling results in an inverse scaling of the Fourier transform:

$$\boxed{\mathbf{P}_\alpha\big(\hat{l}\big)(\Omega_x, \Omega_y, \Omega_\theta, \Omega_\phi) = \cos(\alpha)\hat{l}(\Omega_x, \cos(\alpha)\Omega_y, \Omega_\theta, \Omega_\phi)} \tag{3.35}$$

After the projection, the resulting light-field parametrization is not usual. The central axis is no longer perpendicular to the plane where the spatial domain is defined. The angles with respect to the normal of the $(\vec{x}, \vec{y})$ plane are $\alpha + \delta\theta$ and $\alpha + \delta\phi$ (Figure 3.19).

This is not an issue as the constant angle α results in a shift of the Fourier spectrum (See curvature operator 3.3.4.1).

3.3.4 Reflection

In this section, we characterize the local light-field after a reflection on a surface. We denote incoming (or input) local light-field the local light-field before reflection and outgoing (or output) local light-field the local light-field after reflection. Those local light-fields are defined on the same virtual plane tangent to the surface. We analyse the influence of the local curvature of the surface at the point of intersection on the input local light-field (Section 3.3.4.1). This operation expresses the local light-field on the surface of the object. Operations such as BRDF integration are defined using directions pointing outwards

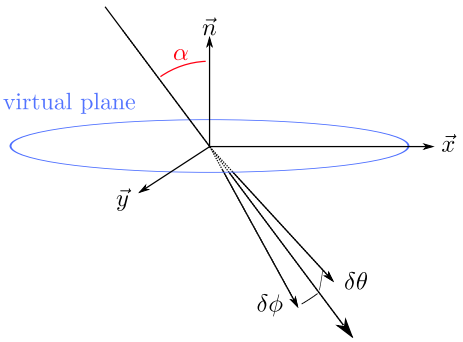


Figure 3.19 *– When we project the light-field on a plane, the spatial and angular components are no longer in an orthogonal setting. The angles $(\delta\theta, \delta\phi)$ are no longer measured with respect to the normal of the plane.*

of the surface. To respect this definition, we apply a symmetry of the signal (Section 3.3.4.2). Our angular parametrization (spherical parametrization) can produce distortions if non infinitesimals angles are used, which is the case of reflection. We describe how we can keep the analysis free of distortions by aligning the incoming and outgoing parametrization for any operator defined over angles (Section 3.3.4.3). The integration of the signal with the BRDF in done in two steps. We first apply the cosine term to the input local light-field (Section 3.3.4.4) and then perform the multiplication and integration with the BRDF (Section 3.3.4.5). As an example, we present two types of BRDF that reduce the multiplication and integration to a convolution: the *Phong* and *half-angle* BRDFs. Finally, we perform an invert curvature projection to obtain the local outgoing light-field on the tangent plane of the object. This last operation will not be described as it is the first curvature operation with the opposite curvature argument.

3.3.4.1 Curvature reparametrization

The first order effect of non-planar surfaces to an incoming light-field is the curvature. Curvature approximates the spatial deformation of the surface. This deformation affects the local normal [37]. Figure 3.20 shows the influence of a change of the curvature matrix on the surface.

The transport from the virtual tangent plane to the local surface of the radiance is expressed by (Figure 3.21):

$$\boxed{\mathbf{C}_k\big(l\big)(\delta x, \delta y, \delta\theta, \delta\phi) = l(\delta x, \delta y, \delta\theta + K(\delta x + \alpha), \delta\phi + K\delta y)} \tag{3.36}$$

Here, K denotes the curvature matrix of the surface with respect to the orientation of the light-field. α is the angle between the incoming direction and the normal. The Fourier equivalent of this operator is the symmetrical shear:

$$\boxed{\mathbf{C}_k\big(\hat{l}\big)(\vec{\Omega_x}, \vec{\Omega_\theta}) = e^{-2i\pi K\alpha\Omega_\theta}\hat{l}(\vec{\Omega_x} - K\vec{\Omega_\theta}, \vec{\Omega_\theta})} \tag{3.37}$$

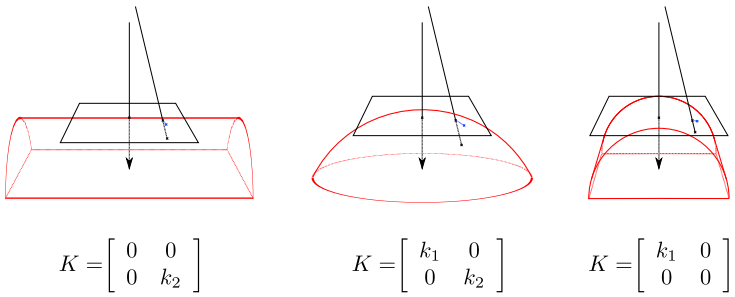


Figure 3.20 *– The matrix of curvature affects at the first order the shear from the planar light-field to the object surface light-field. We show here some examples of such a matrix.*

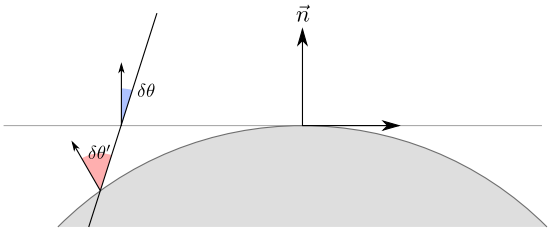


Figure 3.21 *– The curvature operator flatten the input local light-field against the local surface. While the position are approximately equals (at the first order), the angles are modified with respect to their position.*

3.3.4.2 Symmetry of the signal

BRDF operations are defined with respect to outward direction (even for the incoming direction). We express the incoming light-field as an outward light-field. To keep the parametrization right handed, we symmetries the x direction (Figure 3.22).

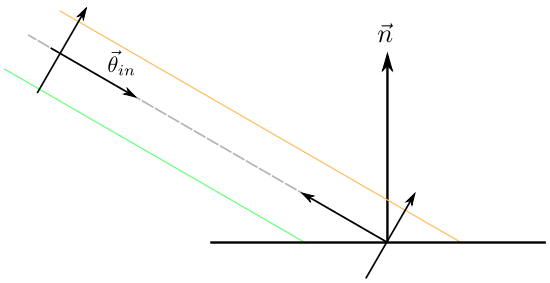


Figure 3.22 *– Before the integration with the BRDF, we express the input signal in the output frame. It corresponds to inverting $\vec{x}_{in}$ and $\vec{z}_{in}$ directions.*

We reverse the spatial parametrization of the input light-field:

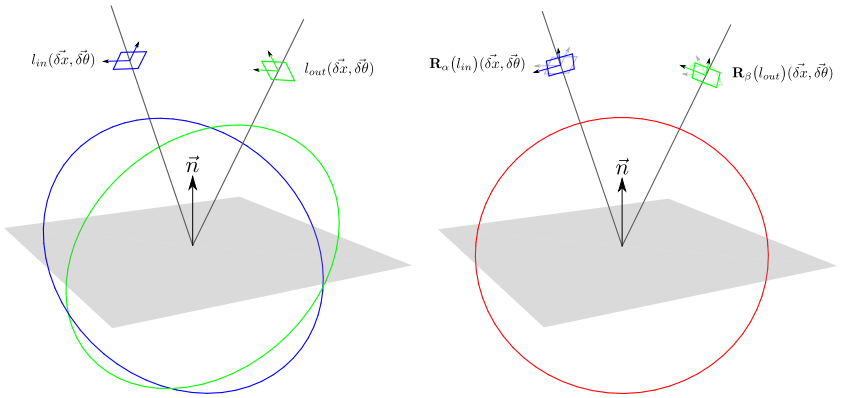
$$\boxed{\mathbf{Sym}\big(l\big)(\delta x, \delta y, \delta\theta, \delta\phi) = l(-\delta x, \delta y, -\delta\theta, \delta\phi)} \tag{3.38}$$

The Fourier equivalent of this partial symmetry is:

$$\boxed{\mathbf{Sym}\big(\hat{l}\big)(\vec{\Omega_x},\vec{\Omega_\theta}) = \hat{l}(-\Omega_x,\Omega_y,-\Omega_\theta,\Omega_\phi)} \tag{3.39}$$

3.3.4.3 Aligning local neighborhoods

Our analysis is valid on local neighborhoods. Thus we need to avoid global coordinates and define operations as relations between two neighborhoods. This is especially true for BRDF operations as phase functions are defined over global angles. To remove the need for a global analysis on angles, we need to keep $\delta\theta$ and $\delta\phi$ as infinitesimals. The solution proposed by Durand et al. [47] is to look at aligned equators of the spherical parametrization for angles (Figure 3.23).



(a) Unaligned equators from input and output frames

(b) Aligned equators from input and output frames

Figure 3.23 *– Given an input local light-field ((a) in blue), and an output local light-field ((a) in green), we rotate both frames to align the $\delta\theta$ angles to make infinitesimal values additives without distortion (b).*

For a given input neighborhood and a given output neighborhood, we rotate the input and output neighborhoods to align the equatorial planes. Then, we can describe angles as being on the equator of the same spherical parametrization, with $\delta\theta$ along the equator and $\delta\phi$ orthogonal to it (Figure 3.24).

The rotation the input local frame that aligns it with the output direction is given as follows: If $(\vec{x_i},\vec{y_i},\vec{z_i})$ is the input frame. and $\vec{o}$ is the output central ray, We project $\vec{o}$ on the input tangent plane:

$$\vec{o_t} = \frac{\vec{o} - \langle\vec{o},\vec{z_i}\rangle\vec{z_i}}{||\vec{o_t}||}$$

We obtain the rotation matrix of the tangent plane as:

$$R = \begin{bmatrix} \langle\vec{o_t},\vec{x_i}\rangle & -\langle\vec{o_t},\vec{y_i}\rangle \\ \langle\vec{o_t},\vec{y_i}\rangle & \langle\vec{o_t},\vec{x_i}\rangle \end{bmatrix}$$

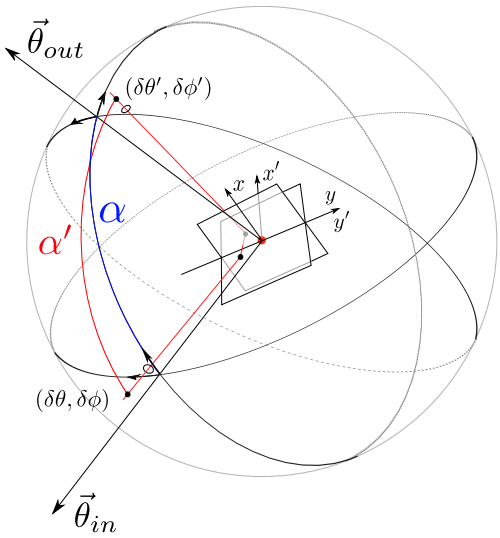


Figure 3.24 *– If the parametrization of the BRDF, or BTDF, follows the equatorial plane, we can use additive notations of angles. At the first order, the angular difference is* $\alpha' \simeq \alpha - (\delta\theta_1 - \delta\theta_2)$

Compound of non-infinitesimal and infinitesimal angles: Some operators reason on angular distances (e.g., Phong BRDF, Snell-Descartes law, ...). Given two local neighborhoods separated by the 2D angle $\vec{\theta} = (\theta, \phi)$ defined from the equator passing through the two central positions (Figure 3.24), the resulting angular distance between the two central positions is θ radians[2]. Given $\vec{\delta\theta}_1 = (\delta\theta_1, \delta\phi_1)$ and $\vec{\delta\theta}_2 = (\delta\theta_2, \delta\phi_2)$, the local angular components of the local neighborhoods, the angular distance between those two positions is given by the geodesic equality:

$$\cos(\theta') = \cos(\theta - (\delta\theta_1 - \delta\theta_2)) \cos(\delta\phi_1 - \delta\phi_2)$$

Given our small angles assumption, the last cosine can be neglected, the resulting angle becomes:

$$\theta' \simeq \theta - (\delta\theta_1 - \delta\theta_2) \tag{3.40}$$

3.3.4.4 Solid angles projection

The rendering equation (Equation 2.1) is defined over solid angles. We perform the integration over the hemisphere by weighting the integral with clamped cosine with respect to the normal of the surface. In a local frame aligned with the normal direction, we write:

$$\boxed{\mathbf{C}(l)(\vec{\delta x}, \vec{\delta\theta}) = l(\vec{\delta x}, \vec{\delta\theta}) cos_+(\delta\theta + \theta_{in})} \tag{3.41}$$

[2]This is a convention, we could derive similar property with ϕ. For that, it is only required to align ϕ with the equator. We made this convention to use θ notation in the Phong BRDF.

θ_{in} is the incoming angle with respect to the normal. This operation becomes a convolution in the Fourier domain:

$$\boxed{\mathbf{C}\big(\hat{l}\big)(\vec{\Omega_x}, \vec{\Omega_\theta}) = l(\vec{\Omega_x}, \vec{\Omega_\theta}) \otimes \big[B_1(\Omega_\theta)\delta(\Omega_x, \Omega_y, \Omega_\phi)\big]} \tag{3.42}$$

Where $B_1(\Omega_\theta)$ is the Fourier transform of the clamped cosine, and $\delta(\Omega_x, \Omega_y, \Omega_\phi) = \delta(\Omega_x)\delta(\Omega_y)\delta(\Omega_\phi)$ is a multidimensional dirac distribution.

3.3.4.5 Integration with the BRDF

Isotropic reflection is defined as the convolution of the input signal with the reflectance [47, 141]. The generalisation to anisotropic materials cannot be modelled the same way. We present the general formulation of the output local light-field after the reflection with the surface. We show that the isotropic case reduces to a convolution as previously mentioned.

General case: Given an incoming light-field $L_{in}(\vec{x}_{in}, \vec{\theta}_{in})$, incident to a planar surface with reflectance $\rho(\vec{\theta}_{in}, \vec{\theta}_{out})$, the reflected light-field $L_{out}(\vec{x}_{out}, \vec{\theta}_{out})$ is:

$$L_{out}(\vec{x}_{out}, \vec{\phi}_{out}) = \int_{\vec{\theta}} L_{in}(\vec{x}_{out}, \vec{\theta}_{in})\rho(\vec{\theta}_{in}, \vec{\theta}_{out}) \cos(\theta_{in}) \mathrm{d}\vec{\theta}_{in} \tag{3.43}$$

Given the main incoming angle $\vec{\theta}_{in}$ and the main outgoing angle $\vec{\theta}_{out}$, we are interested in local information. As we integrated the cosine factor into the local light-field (Section 3.3.4.4), we only have to describe the relation between the local incoming light-field $l_{in}(\vec{\delta x}, \vec{\delta\theta}_{in})$, the local outgoing light-field $l_{out}(\vec{\delta x}, \vec{\delta\theta}_{out})$ and the local BRDF with respect to the central directions $\rho_{\vec{\theta}_{in}, \vec{\theta}_{out}}$[3]:

$$\boxed{\mathbf{R}_\rho\big(l\big)(\vec{\delta x}_{out}, \vec{\delta\theta}_{out}) = \int_{\vec{\delta\theta}_{out}} l(\vec{\delta x}_{out}, \vec{\delta\theta}_{in})\rho_{\vec{\theta}_{in}, \vec{\theta}_{out}}(\vec{\delta\theta}_{in}, \vec{\delta\theta}_{out}) \mathrm{d}\vec{\delta\theta}_{in}} \tag{3.44}$$

This expression is not a convolution. We use the integration formula (Equation 3.18) to obtain the Fourier equivalent of Equation 3.44:

$$\boxed{\mathbf{R}_\rho\big(\hat{l}\big)(\vec{\Omega_{x}}_{out}, \vec{\Omega_\theta}_{out}) = \big[\hat{l}(\vec{\Omega_{x}}_{out}, \vec{\Omega_\theta}_{in}) \otimes_{\vec{\Omega_\theta}_{in}} \hat{\rho}(\vec{\Omega_\theta}_{in}, \vec{\Omega_\theta}_{out})\big]_{\vec{\Omega_\theta}_{in}=\vec{0}}} \tag{3.45}$$

The BRDF operator is defined in a higher dimensional space which is then sliced to obtain the outgoing light-field function. The convolution is defined over the input angles only. Figure 3.25 shows an example of the BRDF operation in the primal using a 2D space of incoming and outgoing angles $(\theta_{in}, \theta_{out})$.

Phong parametrization: Some BRDFs are described using the reflected direction of the incoming direction. The reflected direction is the symmetric of the incoming direction with respect to the normal of the surface. We start by aligning the incoming local light-field with the normal and express the reflected local light-field. No reparametrization is needed as the incoming frame is pointing outward of the surface (Figure 3.26).

[3] In the remainder of the document, we assume BRDF and BTDF are defined with respect to the main incoming and outgoing direction $(\vec{\theta}_{in}, \vec{\theta}_{out})$. We avoid to write the subscripts.

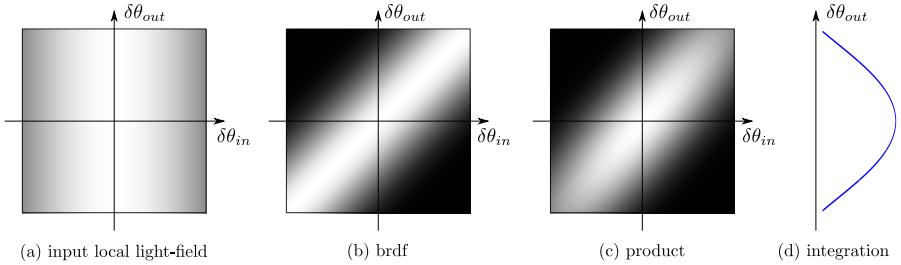


Figure 3.25 *– In the primal space, the integration with the BRDF can be seen in the space of incoming and outgoing angles. First, the input angular part of the local light-field (a) is multiplied with the BRDF (b). This function (c) is then integrated along the incoming angle space to give the outgoing local light-field (d).*

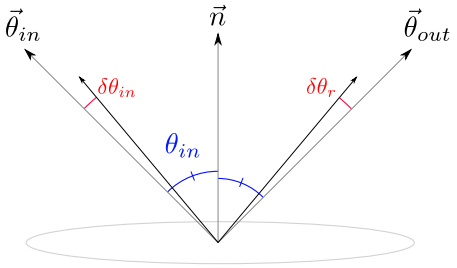


Figure 3.26 *– The reflected local light-field is defined as the reflection of the incoming local light-field with respect to the normal. The parametrization is unchanged by this operation.*

Then, we align the equator of the reflected light-field with the outgoing direction. Now the reflected local light-field and the outgoing local light-field share the same parametrization, we can express the BRDF. It has the following form:

$$\rho(\delta\theta_r, \delta\theta_{out}) = \rho_P(\delta\theta_{out} - \delta\theta_r)$$

The Fourier transform of this class of BRDF is:

$$\hat{\rho}(\Omega_{\theta r}, \Omega_{\theta out}) = \hat{\rho_P}(\Omega_{\theta out})\delta(\Omega_{\theta r} + \Omega_{\theta out})$$

Where $\delta(\Omega_{\theta r} + \Omega_{\theta out})$ is the Dirac distribution. If we inject this form of BRDF into the Fourier equation of the reflection of a local light-field (Equation 3.45), we obtain the convolution formulation [140, 141, 47], which Fourier equivalent is a multiplication:

$$\boxed{\mathbf{R}_\rho\big(\hat{l}\big)(\vec{\Omega_{x out}}, \vec{\Omega_{\theta out}}) = \hat{l}(\vec{\Omega_{x out}}, \vec{\Omega_{\theta out}})\hat{\rho_P}(\vec{\Omega_{\theta out}})} \tag{3.46}$$

Half-angle parametrization: Other BRDFs can be expressed using the half-angle parametrization [147]. Then, the BRDF is a function of the direction halfway between the incoming and the outgoing direction. We handle such BRDF with the following steps:

First we align the incoming local light-field equator with the outgoing direction. The half-angle local light-field share the same parametrization has it lies on the equatorial plane. In this parametrization, the BRDF as the following expression:

$$\rho(\vec{\delta\theta}_{in}, \vec{\delta\theta}_{out}) = \rho\Big(\frac{\vec{\delta\theta}_{in} + \vec{\delta\theta}_{out}}{2}\Big)$$

Its Fourier transform is the following. Note the scaling factor of 2 due to the averaging of angles:

$$\hat{\rho}(\vec{\Omega}_{\theta in}, \vec{\Omega}_{\theta out}) = 2\ \hat{\rho}(2\ \vec{\Omega}_{\theta out})\delta(\vec{\Omega}_{\theta in} - \vec{\Omega}_{\theta out})$$

For isotropic half-angle BRDFs, the phase function is defined with respect to the angular distance between the normal and the half-angle. We align the half-plane local frame with the normal to express the BRDF and rotate the result back in the alignment of incoming and outgoing directions.

$$\rho(\vec{\delta\theta}_{in}, \vec{\delta\theta}_{out}) = \rho_h\bigg(R\ \Big(\frac{\vec{\delta\theta}_{in} + \vec{\delta\theta}_{out}}{2}\Big)\bigg)$$

In the same way as Equation 3.46, we can express the BRDF operator as a simple product between the incident local light-field spectrum and the BRDF spectrum:

$$\boxed{\mathbf{R}_\rho\big(\hat{l}\big)(\vec{\Omega}_x, \vec{\Omega}_{\theta out}) = 2\ \hat{l}(\vec{\Omega}_x, -\vec{\Omega}_{\theta out})\hat{\rho}(2\ \vec{\Omega}_{\theta out})} \tag{3.47}$$

3.3.4.6 Spatially varying BRDFs

Textures We have defined the behaviour light reflecting on homogeneous surfaces on which the reflectance is the same. Most objects do have a spatially varying appearance (like a wood block, a paper sheet,...). We add this effect is by multiplying the BRDF by a spatially varying but uncorrelated signal: a texture (Heckbert proposed a survey of texture mapping techniques. [72]).

We can apply a texture function to our signal before reprojecting to the outgoing frame. Since the BRDF and the texture are uncorrelated, there is no need to apply one before the other: The BRDF modifies the angular part of the light-field and the texture its spatial part.

$$\boxed{\mathbf{Tx}\big(l\big)(\vec{\delta x}, \vec{\delta\theta}) = t(\vec{\delta x}) \times l(\vec{\delta x}, \vec{\delta\theta})} \tag{3.48}$$

The translation of this multiplication is a convolution in the Fourier domain. This operator spreads the spatial frequency of the texture to the outgoing light-field:

$$\boxed{\mathbf{Tx}\big(\hat{l}\big)(\vec{\Omega}_x, \vec{\Omega}_\theta) = \big(\hat{t} \otimes_{\vec{\Omega}_x} \hat{l}\big)(\vec{\Omega}_x, \vec{\Omega}_\theta)} \tag{3.49}$$

Varying roughness Another way to modify the appearance of a model using texture functions is to alter parameters of the BRDF based on the texture. Roughness textures contain the value of the exponent (or the standard deviation) of the Phong lobe (or normal distribution). These tools can be very efficient to depict the effect of corrosion on a metallic surface for example.

These textures bring correlation between space and angle. High variations of the spatial roughness will impact the angular appearance. In the general case, we assume that the BRDF is also a function of the spatial component:

$$\boxed{\mathbf{R}_\rho\big(l\big)(\vec{\delta x}, \vec{\delta\theta}_{out}) = \int_{\vec{\delta\theta}_{in}} l(\vec{\delta x}, \vec{\delta\theta}_{in})\rho(\vec{\delta x}, \vec{\theta}_{in} + \vec{\delta\theta}_{in}, \vec{\theta}_{out} + \vec{\delta\theta}_{out})\mathrm{d}\vec{\delta\theta}_{in}} \tag{3.50}$$

Like for the incoming angles, the formulation in Fourier space becomes a convolution along the spatial and angular dimensions. The BRDF can be seen as a 6D kernel:

$$\boxed{\mathbf{R}_\rho\big(\hat{l}\big)(\vec{\Omega_x}, \vec{\Omega_{\theta out}}) = \big[\hat{l}(\vec{\Omega_x}, \vec{\Omega_{\theta in}}) \otimes_{\vec{\Omega_{\theta in}}} \hat{\rho}(\vec{\Omega_x}, \vec{\Omega_{\theta in}}, \vec{\Omega_{\theta out}})\big]_{\vec{\Omega_{\theta in}}=\vec{0}}} \tag{3.51}$$

3.3.5 Refraction

Refraction is not part of the initial paper on Fourier analysis of light transport. We define the behaviour of the local light-field refracted by a rough glass surface. This operator replaces the BRDF operator when the material is refractive. The curvature, symmetry and alignment have to be performed before this operator.

As stated by Walter et al. [184], the refraction of rough surfaces is defined with respect to the refracted specular ray. We first give a definition of the specular refracted local light-field (Section 3.3.5.1). We need to add a windowing of the input light-field to model the extinction of light at the critical angle (Section 3.3.5.2). Then we derive the integration of the BTDF (Section 3.3.5.3).

3.3.5.1 Specular transmission

The Snell-Descartes law for refraction describes how the angular part of a light-field is affected by the interface. The relation between incoming and outgoing angles is well known:

$$n_1 \sin(i_1) = n_2 \sin(i_2) \tag{3.52}$$

Remind that we are looking at small neighborhoods of rays, our incoming angles are thus $i_1+\delta i_1$ and the outgoing angles are $i_2+\delta i_2$. The Fresnel relation between the incoming neighborhood and the outgoing neighborhood becomes:

$$n_1 \sin(i_1 + \delta i_1) = n_2 \sin(i_2 + \delta i_2) \tag{3.53}$$

Which gives the following first order approximation:

$$\delta i_2 \simeq \frac{n_1 \sin(i_1) - n_2 \sin(i_2)}{n_2 \cos(i_2)} + \delta i_1 \frac{n_1 \cos(i_1)}{n_2 \cos(i_2)} \tag{3.54}$$

The Snell-Descartes law (Equation 3.52) still holds for the main incoming angle i_1 and the main outgoing angle i_2. This supplementary condition cancels the shift:

$$\delta i_2 \simeq \delta i_1 \frac{n_1 \cos(i_1)}{n_2 \cos(i_2)} \tag{3.55}$$

The formulation of the transmitted light-field from the incoming light-field is thus:

$$\boxed{\mathbf{Tr}\big(l\big)(\vec{\delta x}, \delta\theta, \delta\phi) = l\left(\vec{\delta x}, \frac{n_2\cos(i_2)}{n_1\cos(i_1)}\delta\theta, \delta\phi\right)} \tag{3.56}$$

The translation of this operator in Fourier space is a phase shift and a scale of the space.

$$\boxed{\mathbf{Tr}\big(\hat{l}\big)(\vec{\Omega_x}, \Omega_\theta, \Omega_\phi) = \frac{n_1\cos(i_1)}{n_2\cos(i_2)}\hat{l}\left(\vec{\Omega_x}, \frac{n_1\cos(i_1)}{n_2\cos(i_2)}\Omega_\theta, \Omega_\phi\right)} \tag{3.57}$$

3.3.5.2 Critical angle

There exists an incoming angle above which the Snell-Descartes law is no longer applicable. This angle is called the critical angle θ_c. Since we are looking at a neighborhood, we need to add the critical angle window of the transmission. We add before the specular transmission scale a window function:

$$\boxed{\mathbf{Tr}\big(l\big)(\vec{\delta x}, \delta\theta, \delta\phi) = w_{\theta_c-\theta_i}(\delta\theta)l\left(\vec{\delta x}, \frac{n_2\cos(i_2)}{n_1\cos(i_1)}\delta\theta, \delta\phi\right)} \tag{3.58}$$

The window function $w_{\theta_{ext}-\theta_i}(\delta\theta)$ zeros when $\delta\theta > \theta_{ext} - \theta_i$. This window restricts our analysis to a smaller domain where analysis is still meaningful. The Fourier equivalent is:

$$\boxed{\mathbf{Tr}\big(\hat{l}\big)(\vec{\Omega_x}, \Omega_\theta, \Omega_\phi) = \frac{n_1\cos(i_1)}{n_2\cos(i_2)}w_{\theta_c-\theta_i}(\Omega_\phi) \otimes_{\Omega_\theta} \hat{l}\left(\vec{\Omega_x}, \frac{n_1\cos(i_1)}{n_2\cos(i_2)}\Omega_\theta, \Omega_\phi\right)} \tag{3.59}$$

3.3.5.3 Rough materials

When the surface is not microscopically planar, such as for microfacets materials, the distribution of normals is described statistically and must be integrated to get the final radiance. After the refraction (Equation 3.56) we have to perform the convolution with a BTDF characterized by the distribution of normals:

$$\boxed{\mathbf{Tr}_\rho\big(l\big)(\vec{\delta x}, \vec{\delta\theta_{out}}) = \int_{\vec{\delta\theta_{in}}} \rho(\vec{\delta\theta_{in}}, \vec{\delta\theta_{out}})\; l(\vec{\delta x}, \vec{\delta\theta_{in}})\mathrm{d}\vec{\delta\theta_{in}}} \tag{3.60}$$

Note that we didn't add the macrosurface normal into Equation 3.60 as we suppose the normal to be the up direction of our virtual surface. We modulate the BTDF in function of the sign instead of using it as a parameter.

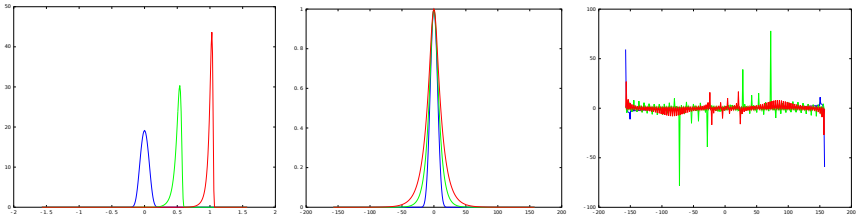
The above equation is exactly like the BRDF integration (Equation 3.44) with the notable difference that our angles are defined below the virtual surface.

Thus, the Fourier transform will have the same formulation of a convolution in a 7D space, with a 4D kernel, followed by a slice:

$$\boxed{\mathbf{Tr}_\rho\big(\hat{l}\big)(\vec{\Omega_x},\vec{\Omega_{\theta out}}) = \big[\hat{l}(\vec{\Omega_x},\vec{\Omega_{\theta in}}) \otimes_{\vec{\Omega_{\theta in}}} \hat{\rho}(\vec{\Omega_{\theta in}},\vec{\Omega_{\theta out}})\big]_{\vec{\Omega_{\theta in}}=\vec{0}}} \tag{3.61}$$

Spatially varying BTDFs are handled the same way. We did not recopy Equation 3.50 and Equation 3.51 as they are identical.

Fourier transforms of BTDFs: Walter et al. [184] described the theoretical model of rough refraction from the classical microfacets model. They proposed to use the GGX distribution for the PDF of normals. de Rousiers et al. [39] fitted isotropic Gaussians on the resulting BTDF. While this fitting is wrong because of the anisotropy of the BTDF (Figure 3.27a), the Fourier spectrum is isotropic in amplitude (Figure 3.27b).



(a) GGX BTDF **(b)** Amplitude of the Fourier transform of the GGX BTDF **(c)** Phase of the Fourier transform of the GGX BTDF

Figure 3.27 *– GGX BTDFs are not isotropic for grazing angles (a), but the amplitude of their Fourier transform is (b). The non-symmetrical information is stored in the phase (c). We approximate the Fourier spectrum of GGX BTDFs by Gaussians which standard deviation correspond to the inverse roughness, much like de Rousiers et al. [39]. We used a glass index of* $n = 1.1$*, the incoming angles are* 0*,* $\frac{\pi}{6}$*, and* $\frac{\pi}{3}$

We can approximate the amplitude with a Gaussian distribution. However, phase does not have such an approximate formulation.

While de Rousiers et al. introduced an error in their isotropic fitting, remark that fitting the BTDF lobe (with an isotropic Gaussian) is equivalent to fitting the amplitude of the Fourier transform. Thus, what they obtain is a BTDF that has almost the same frequency response than the original one. Doing so, the perception of blur in the rendered images with this fitted BTDF will be close enough to the original BTDF for a real-time applications purpose.

3.3.6 Lens

The travel of a light field into a camera lens has been studied by Soler et al. [164] who proposed to look at the integration on the sensor at the same time. While this formulation captures the blurring effect of depth-of-field, it does not provide the outgoing light-field. The angular integration on the sensor is

already performed. To get the light-field on the sensor, we may want to look at what happens just after the lens. Figure 3.28 illustrates the behaviour of light ray through a lens.

3.3.6.1 Lens + integration operator

The integration of light field after its travel through a lens is resumed in the following equation :

$$\mathbf{L}_{(f,r)}\big(l(\vec{\delta x})\big) = \int_{\vec{\delta\theta}} v(\vec{\delta x},\vec{\delta\theta}) l(\vec{\delta x},\vec{\delta\theta}) \tag{3.62}$$

Where $v(\vec{\delta x},\vec{\delta\theta})$ is a binary valued function which equals one is the ray in direction $\vec{\delta\theta}$ from the position $\vec{\delta x}$ passes through the lens, and zero otherwise.

We will not derive the Fourier equivalent of this formula as it does not follow the notion of operator we defined. The resulting operator would break the possibility to compose other operators afterwards.

3.3.6.2 Small and Thin Lenses

We look at the effect of lens and travel in the camera on a neighborhood of a ray to obtain a formula compatible with the definition of operator.

From an input local light-field arriving at the lens $l(\vec{\delta x},\vec{\delta\theta},t)$ we want to characterize the local light-field at the sensor position oriented along the central direction of the sensor and lens $l'(\vec{\delta x},\vec{\delta\theta},t)$.

When the light-field passes through the lens, its direction is locally changed due to the curvature of the first interface, then it travels inside the lens, and finally gets out of the lens with another curvature effect. Since the lens is assumed to be thin, travel can be neglected between the two interfaces [60, Chap II.4.1].

Then, the light leaves the lens and travels to the sensor. The light-field is in or out of focus depending on the distance between the lens and the sensor.

$$\boxed{\mathbf{L}_{(d_1,d_2)}\big(l\big)(\vec{\delta x},\vec{\delta\theta}) = l(\vec{\delta x} + d_1(\vec{\delta\theta} + f\vec{\delta x}), \vec{\delta\theta} + f\vec{\delta x})} \tag{3.63}$$

Where f is the focal length of the lens, and d_1 is the distance from the outgoing central position on the lens to the sensor position and d_2 is the distance of the plane in focus from the lens (Figure 3.28).

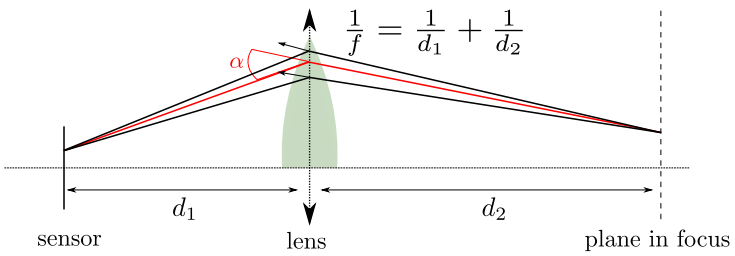


Figure 3.28 *– Light coming from the plane in focus converges at the focal point after passing through the lens.*

The Fourier equivalent of Equation 3.63 is:

$$\boxed{\mathbf{L}_{(d_1,d_2)}\big(\hat{l}\big)(\vec{\Omega_x},\vec{\Omega_\theta}) = \hat{l}(\vec{\Omega_x} - f\vec{\Omega_\theta}, \vec{\Omega_\theta} - d_1(\vec{\Omega_\theta} + f\vec{\Omega_x}))} \tag{3.64}$$

Example We illustrate this analysis in a 2D formalism for clarity (See Figure 3.29). Given a point in focus (with coordinate $[\delta x, \delta\theta]$ in the local light-field), the position on the lens after travel will be:

$$\begin{bmatrix}\delta x_l \\ \delta\theta_l\end{bmatrix} = \begin{bmatrix}\delta x - d_2\delta\theta \\ \delta\theta\end{bmatrix} \tag{3.65}$$

Due to the travel shear. Using Equation 3.63 and Equation 3.65, we can write the position on the sensor as:

$$\begin{aligned}\delta x_s &= \delta x - d_2\delta\theta - d_1\big(\delta\theta + \frac{1}{f}(\delta x - d_2\delta\theta)\big) \\ &= \delta x(1 + \frac{1}{f})\end{aligned} \tag{3.66}$$

The influence of the angle $\delta\theta$ on the final spatial component of the local light-field at the sensor vanishes. This indicates that the point is indeed in focus.

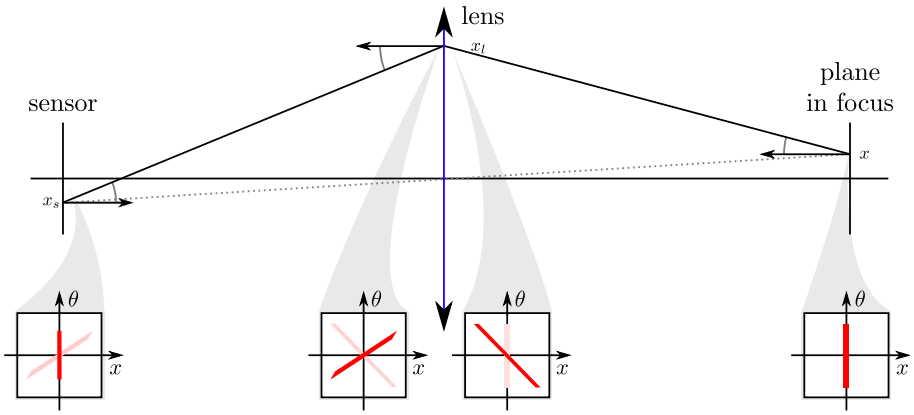


Figure 3.29 *– For an in-focus diffuse point, the resulting angular influence on the position vanishes due to the compensation of the travel shears and the curvature shears.*

3.3.7 Participating media

Participating media such as smoke, liquids, etc, are usually harder to incorporate into ray tracing based rendering engines as the formulation of radiance coming to the camera is no longer defined per surfaces, but in a volume. In this section, we propose operators working on our 2D orthogonal plane parametrization to handle the effect of both attenuation (Section 3.3.7.1) and scattering (Section 3.3.7.2)[4]

[4]Emission can be expressed as an interaction of the medium, but we propose to model emission with light sources and then apply operators.

3.3.7.1 Attenuation

We study the effect of volumetric attenuation of a density function $\sigma(\delta x, \delta y, \delta z)$ along a ray, for a travel distance d. Without loss of generality, we suppose that the ray travels along δz. The attenuated light field is:

$$\mathbf{A}\big(l\big)(\vec{\delta x}, \vec{\delta\theta}) = l(\vec{\delta x}, \vec{\delta\theta}) e^{-\int_0^d \sigma(\vec{\delta x}, u)\mathrm{d}u} \tag{3.67}$$

Homogeneous medium: Considering the case where the attenuation function is constant, in the case of homogeneous media, the integral has an analytical form:

$$\boxed{\mathbf{A}\big(l\big)(\vec{\delta x}, \vec{\delta\theta}) = l(\vec{\delta x}, \vec{\delta\theta}) e^{-\sigma d}} \tag{3.68}$$

The Fourier transform is only affected in amplitude by this constant factor:

$$\boxed{\mathbf{A}\big(\hat{l}\big)(\vec{\Omega_x}, \vec{\Omega_\theta}) = e^{-\sigma d} \hat{l}(\vec{\Omega_x}, \vec{\Omega_\theta})} \tag{3.69}$$

Non-homogeneous medium: When the density of particles is not constant in space, the energy is not uniformly absorbed during the travel. This increases spatial frequencies of the signal, which further propagates to the angular domain because of the travel of light.

We consider ds to be small enough to derive a first order approximation of the attenuation:

$$\boxed{\mathbf{A}\big(l\big)(\vec{\delta x}, \vec{\delta\theta}) = l(\vec{\delta x}, \vec{\delta\theta})(1 - ds\sigma(\vec{\delta x}, \delta z))} \tag{3.70}$$

Let σ_{xy} be the restriction of σ to the (x, y) plane. We adopt the notation $p(\delta x, \delta y) = 1 - ds\sigma_{xy}(\delta x, \delta y)$. In the Fourier domain, we can write:

$$\boxed{\mathbf{A}\big(\hat{l}\big) = \hat{l} \otimes_{\vec{\Omega_x}} \hat{p}} \tag{3.71}$$

In this equation, $\otimes_{\vec{\Omega_x}}$ denotes a convolution over the spatial component only. The effect of attenuation is therefore identical to occlusion, except that the mask $p = 1 - ds\sigma_{xy}$ is a function taking arbitrary values in $[0, 1]$ instead of a binary function.

3.3.7.2 Scattering

We inspire from analytical formulations of beam-beam (or line-beam) integration [168, 88]. We assume having as input local light-field a finite size beam (both spatial and angular), and that the attenuation is negligible within the beam size. Those assumptions are compatible with the infinitesimal analysis. In such case, the scattering is a double integral, one along the outgoing ray, the other with the phase function (Figure 3.30).

With the assumption that the phase function is isotropic, we perform the scattering operator in two steps. First, we convolve the input local light-field with the phase function taking into account the main outgoing direction (Figure 3.31 (b)). Second, we integrate along the outgoing ray (Figure 3.31 (c)).

To integrate over the outgoing direction, it is necessary to have a volumic definition of the local radiance. We will do the following derivations in flatland to avoid too lengthy equations. As for the isotropic Phong BRDF convolution, the convolution is done on one angular dimension only.

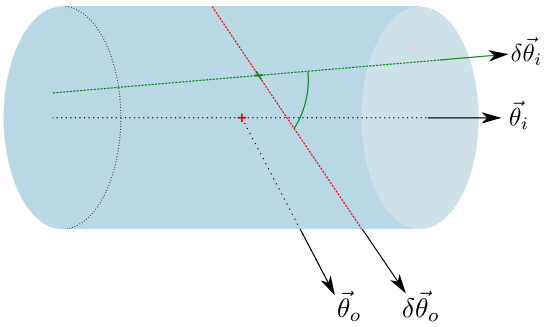
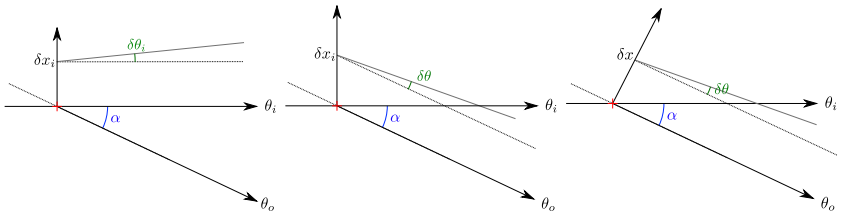


Figure 3.30 *– To compute the outgoing radiance of local direction $\delta\omega_o$ we need to integrate along this ray (in red), for all the incoming local direction $\delta\omega_i$ crossing it.*



(a) Input configuration **(b)** After angular integration with the phase function **(c)** Output configuration

Figure 3.31 *– The input configuration of the scattering operator. The input local light-field (a) is first convolved with the phase function resulting in an intermediate light-field with angular and spatial components unaligned (b). Finaly, we integrate the spatial component in the outgoing frame along the outgoing direction (c).*

Volumic local radiance: We define a volume by considering radiance before and after the central scattering location. We consider this volume infinitesimal in all directions. The local volumic radiance function is then:

$$l_i(\delta x, \delta z, \delta\theta) = l_i(\delta x + \delta z \tan(\delta\theta), \delta\theta)$$

Due to first order assumptions, the volumic radiance is constant with respect to the depth δz:

$$l_i(\delta x, \delta z, \delta\theta) \simeq l_i(\delta x, \delta\theta)$$

We compute the local volumic outgoing radiance by integrating the product of the local volumic incoming radiance with the phase function:

$$\begin{aligned} l_o(\delta x, \delta z, \delta\theta) &= \int l_i(\delta x, \delta z, \delta\theta')\rho_\alpha(\delta\theta', \delta\theta)\mathrm{d}\delta\theta' \\ &= \int l_i(\delta x, \delta\theta')\rho_\alpha(\delta\theta', \delta\theta)\mathrm{d}\delta\theta' \end{aligned} \tag{3.72}$$

ρ_α is a local phase function. The scattering is defined with respect to α, the angle between the two central rays. The local volumic outgoing radiance is still independent of the depth.

Outgoing local radiance: We estimate the outcoming local radiance by evaluating the volumic local radiance in the outgoing local frame and integrating along local rays:

$$l_o(\delta x, \delta\theta) = \int l_i(\delta x \cos(\alpha+\delta\theta)+\delta z \sin(\alpha+\delta\theta), \delta z \sin(\alpha+\delta\theta)-\delta x \sin(\alpha+\delta\theta), \delta\theta)\mathrm{d}\delta z$$

Using the infinitesimal analysis, we can rewrite scale with cosines: $\delta x \cos(\alpha+\delta\theta) = \delta x \cos(\alpha) - \delta x \delta\theta \sin(\alpha) \simeq \delta x \cos(\alpha)$.

$$l_o(\delta x, \delta\theta) = \int l_i(\delta x \cos(\alpha) + \delta z \sin(\alpha), \delta z \cos(\alpha) - \delta x \sin(\alpha), \delta\theta)\mathrm{d}\delta z \quad (3.73)$$

We have to distinguish two cases here. One is the case where $\sin(\alpha) \simeq 0$. In such a case, the depth integration can be neglected as the integrand is constant in the integration domain. On the other case, it cannot be neglected if the incoming and outgoing directions are not almost aligned.

Planar behaviour: In the case of forward scattering and backward scattering ($|\omega_i.\omega_o| \simeq 1$), we can approximate the volumic outgoing radiance as being constant along the δz component. It results in the second integration having no effect on the distribution of energy. In such a case, the outgoing local radiance is then:

$$\boxed{l_o(\delta x, \delta\theta) = \int l_i(\delta x, \delta\theta')\rho_\alpha(\delta\theta, \delta\theta')\mathrm{d}\delta\theta} \quad (3.74)$$

The Fourier equivalent of this equation is:

$$\boxed{\mathbf{S}_{\rho_\alpha}(\hat{l})(\vec{\Omega_x}, \vec{\Omega_{\theta out}}) = \left[\hat{l}(\vec{\Omega_x}, \vec{\Omega_{\theta in}}, \vec{\Omega_{\theta out}}) \otimes_{\vec{\Omega_{\theta in}}} \hat{\rho_\alpha}(\vec{\Omega_{\theta in}}, \vec{\Omega_{\theta out}})\right]_{\vec{\Omega_{\theta in}}=\vec{0}}} \quad (3.75)$$

Non-planar behaviour: For non-forward and non-backward scattering, the integration in depth cannot be avoided. We combine Equation 3.73 and Equation 3.73 to produce the relation between the incoming local light-field and the outgoing local light-field:

$$\boxed{l_o(\delta x, \delta\theta) = \int\int l_i(\delta z, \delta\theta')\rho_\alpha(\delta\theta, \delta\theta')\mathrm{d}\delta\theta\mathrm{d}\delta z} \quad (3.76)$$

The Fourier equivalent of this equation is:

$$\boxed{\mathbf{S}_{\rho_\alpha}(\hat{l})(\vec{\Omega_x}, \vec{\Omega_{\theta out}}) = \left[\hat{l}(0, \Omega_y, \vec{\Omega_{\theta in}}, \vec{\Omega_{\theta out}}) \otimes_{\vec{\Omega_{\theta in}}} \hat{\rho_\alpha}(\vec{\Omega_{\theta in}}, \vec{\Omega_{\theta out}})\right]_{\vec{\Omega_{\theta in}}=\vec{0}}} \quad (3.77)$$

3.3.8 Motion

Motion affects indirectly the local light-field as it has to be coupled with one of the previsouly described operator (e.g., occlusion, reflection, ...). Figure 3.32 present the example of the compound of motion and occlusion. We treat occlusion as a projection of the local frame to a static setting. For example, we will consider the occlusion operator on a frame attached to the occluder, making

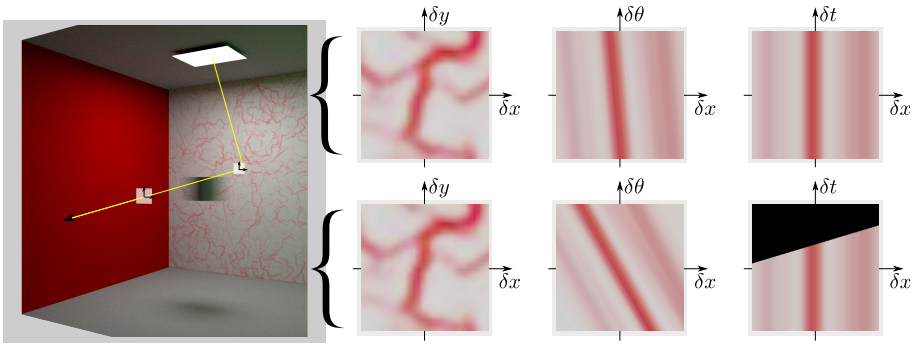


Figure 3.32 *– Motion affects occlusion by shearing its effect on the local light-field. In this example, a cube is translating along the X component of the world coordinates. It partially occlude the local light-field. The effect of motion is visible on the $x-t$ slice. But it is not observable on other slices as there is no occlusion for $t = 0$.*

the operator time independant. For that we add a time projection before and after the operator.

When analysing the influence of motion on a light-field, we need to add the time dimension in the analysis. We keep the infinitesimal analysis formulation for the time dimension. The light-field function becomes $l(\vec{\delta x}, \vec{\delta\theta}, \delta t)$. In this context, we assume that motion is linear. Under this assumption, we can show that the motion operator is a linear shear from the motion dimension (either space or angle) to the time dimension[5]. We illustrate the variables for a flatland configuration in Figure 3.33.

We introduce an intermediate distance: d (Figure 3.33(b)). It is the position of a photon at position $(\delta x, \delta\theta)$ after a travel of δz_t meters with respect to the moving center $\mathbf{o_t}$. d is given by the following formula:

$$d = \delta x + \tan(\delta\theta)\delta z_t - \delta x_t$$

The new position $\delta x'$ is then:

$$\delta x' = d(\cos(\delta\theta_t) + \sin(\delta\theta_t)\tan(\delta\theta'))$$

We simplify this equation using the first order analysis:

$$\delta x' \simeq \delta x - \delta x_t$$

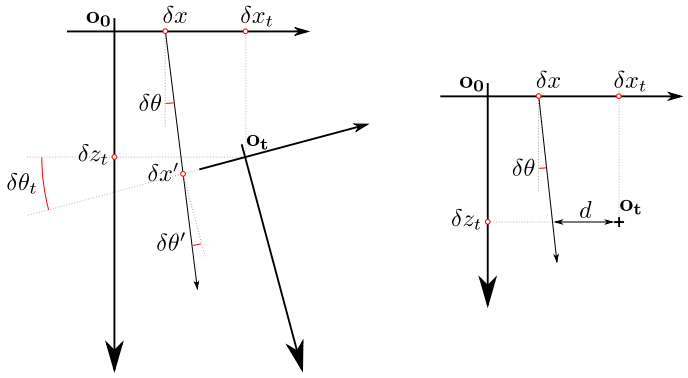
If we rewrite the moving frame position as the spatial velocity times the infinitesimal time variable, $\delta x_t = \vec{v}_x \delta t$ we get the following formula:

$$\boxed{\mathbf{M}_{\vec{v},\vec{r}}(l)(\vec{\delta x}, \vec{\delta\theta}, \delta t) = l(\vec{\delta x} - \vec{v}\delta t, \vec{\delta\theta} - \vec{r}\delta t, \delta t)} \tag{3.78}$$

Given the shear behaviour of this operator, its translation in the frequency domain becomes the symmetrical shear:

$$\boxed{\mathbf{M}_{\vec{v},\vec{r}}(\hat{l})(\vec{\Omega_x}, \vec{\Omega_\theta}, \Omega_t) = l(\vec{\Omega_x}, \vec{\Omega_\theta}, \Omega_t + \vec{v}.\vec{\Omega_x} + \vec{r}.\vec{\Omega_\theta})} \tag{3.79}$$

[5]The notion of sheared transformation for motion has been first proposed by Egan et al. [49] for various special cases of one bounce reflection. During this thesis, we generalized it for any incoming light-field allowing it to be added to the set of operators [47]



(a) Configuration for motion **(b)** Intermediate step

Figure 3.33 *– The effect of motion on positions and angles is illustrated here. We study 2D coordinates $(\delta x, \delta\theta)$ and its transformation after a unit time motion : $(\delta x', \delta\theta')$. The tangential motion of space $\mathbf{o_t}$ with respect to space $\mathbf{o_0}$ is δx_t. The non-tangential motion is δz_t and the angular motion is $\delta\theta_t$ (a). To obtain the value $\delta x'$ we need an intermediate step and compute the distance of the photon at position $(\delta x, \delta\theta)$ after a travel of δz_t meters along the Z axis with respect to the time changing center $\mathbf{o_t}$ (b).*

Non tangential motions are neglected in this approach, due to the second order nature of this phenomenon.

3.4 Comparison with Differential Analysis

In this section we review differential methods for ray tracing. We separate this section from the previous work to provide a better comparison against Fourier analysis of local radiance. We can differentiate two classes of work on differential analysis. The first one looks at the differentials of structures used to carry radiance: *rays* and *light-paths*. The second one analyses the differentials of radiance or irradiance close to a ray, or near a point on a surface.

3.4.1 Comparison with ray differentials

Ray [81] and light-path [24] differentials are based on the analysis specular reflection and transmission. The differential of rays are studied with respect to surface curvature, travel, etc. In a way, those ray differential methods look at variation of a light-path with a constant radiance constraint [83]. This can have benefits for performing analysis on specular paths [24, 83], filtering [81] (when the light-path is traced backward, the input energy is the pixel characteristic function), or estimating density reconstruction kernels [151, 152].

Our local first-order light-field analysis on the other hand allows to keep track of the radiance variations on neighborhood of rays. It encapsulates in a way ray differential methods as iso-values of the radiance function can be analyzed. The other notable difference is that no analysis window is defined in ray (or light-paths) differentials. It follows that the space defined by the differential can become extremely large.

3.4.2 Comparison with radiance and irradiance differentials

3.4.2.1 Irradiance differential

Irradiance differential has been well studied in the case of radiosity. There, mesh segmentation is the main motivation [6, 80]. Holzschuch and Sillion [80] also derived an error bound on the radiosity simulation from this study. Irradiance differential regained interest with the irradiance caching community [186, 102, 90] where it is used to derive the density of needed cache records. Fourier analysis can be used for irradiance analysis. Local irradiance can be obtain by slicing the local frequency spectrum in angle (We show an application in Section 5.3).

3.4.2.2 Radiance differential

Ramamoorthi et al. [141] derived the propagation of differential information (Gradient and Hessian) through operators defined by Durand et al. [47] (e.g., transport, occlusion and reflection). Two major differences are to note compared Fourier analysis.

First, they derived non planar occlusion differential using the occluder curvature. This is not done in any of the Fourier based methods as most assume a planar occluder to constraint occlusion to be a spatial only phenomenon. We showed the derivation of a non planar model, but neglected the curvature.

Second, they provided an analysis of the different terms inside the differential. For example, in the one bounce case, they showed how curvature affects the illumination on the screen in a separable way. This has been used to perform retargeting, enhancement of rendering or even plausible deferred shading rendering based on partial information [179]. Fourier methods are not able to separate influence of previous operators parameters on a local light-field.

While they can provide a separable formulation for one bounce illumination, propagating complex light-paths would make separability intractable. In such case, they could only provide propagation of the gradient vector or the Hessian matrix. This is very similar to the bandwidth vector or to the covariance matrix formulation (Those will be defined in Chapter 4).

4 | Representations of the Local Light-field Spectrum

We saw in Chapter 3 how to express the changes of a local light-field spectrum when it undergoes an operation (e.g., travel, occlusion, reflection, ...). In this chapter, we present tools to evaluate information of the local light-field spectrum after a chain of operations. The evaluation of the entire spectrum is impracticable in our context, so we rely on descriptors. The spatial extent and orientation of the signal's spectrum are of practical interest since they tell how much the signal varies and in which direction.

Occlusion's spectrum is also impracticable to evaluate. We showed in previous chapter that we can estimate an approximate spectrum from the distance, depth and direction to the occluder (Section 3.3.2). We propose a method based on a voxel grid to evaluate those quantities.

In this chapter we present the following contributions:

- A compact representation of the signal's spectrum using the second moments matrix: the *covariance matrix* (Section 4.2). This representation is fully integrated into the frequency analysis theory. We validate the covariance matrix against measurements, and show that it correctly depicts informations about the real local spectrum with comparisons to measured spectra.
- Two data structures to evaluate local occlusion using voxel grids (Section 4.3.2). Those structures are easy to integrate into existing raytracer.

First, we will review previously proposed spectrum representations (Section 4.1). Then, we will introduce and validate the *covariance matrix* (Section 4.2). Finally, we will present and compare structures to estimate local occlusion that are designed to work in a global illumination context (Section 4.3).

4.1 Previous Works

In this section, we present previously used representations of the local spectrum and emphasis that those tools are not fitted for anisotropic and global illumination analysis.

4.1.1 Bandwidth of the Spectrum

We define the bandwidth as a point corresponding to the 99^{th} percentile of the spectrum along its dimensions. It gives information about the spread of the spectrum:

$$\vec{b} = \vec{p} \in \mathbb{R}+^5 \text{ such that } \int_{t=-\vec{p}}^{\vec{p}} f(\vec{t})\mathrm{d}\vec{t} = 0.99 \int_{t=-\infty}^{\infty} f(\vec{t})\mathrm{d}\vec{t} \tag{4.1}$$

Assuming the spectrum has finite support, this point defines a box in this space which contains the spectrum (Figure 4.1). We call this point the *bandwidth vector*[1].

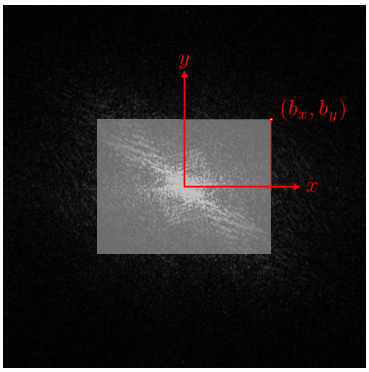


Figure 4.1 *– The bandwidth of the signal is the point corresponding to the 99^{th} percentile of the spectrum. It defines a box containing most of the energy of the spectrum.*

This tool (in a 2D setting[2]) has been used as a prior-analysis in Durand et al.'s bilateral filtering method for rendering [47]. They derived the approximation of the bandwidth after one bounce of infinite frequency angular light source. They predicted the adaptive density for shading in screen space and performed reconstruction on a small set of sample using a bilateral filter.

Bagher et al. [9] used it to estimate both the number of samples required for integrating acquired materials and the number of shading evaluations in screen space. To build the bandwidth buffer, they approximate the bandwidth of the integral with the weighted average of the bandwidths of the incoming rays.

4.1.1.1 Application of the Different Operators

The bandwidth propagation is build on its vector notation. Given a bandwidth vector $\vec{b} = (b_x, b_\theta)^T$ we describe the matrices for the linear operators, or the direct transformation of the vector for non-linear effects. We only describe operators present in the corresponding publication [9].

[1] The matricial derivation of the bandwidth vector is part of the contribution of this thesis (See [9]). But we present it as a previous work to emphasis on the covariance matrix's advantages.

[2] One dimension for space, one dimension for angle. The spectrum is supposed to be isotropic in space and isotropic in angle

Travel After a travel of d meters, the resulting bandwidth vector will be:

$$\vec{b'} = \mathbf{T}_d \, \vec{b}$$

Where the 2D transport matrix is:

$$\mathbf{T}_d = \begin{bmatrix} 1 & 0 \\ -d & 1 \end{bmatrix}$$

Curvature After the virtual projection on a surface with a local maximum curvature κ, the resulting bandwidth vector will be:

$$\vec{b'} = \mathbf{C}_\kappa \, \vec{b}$$

Since the derivation is done in 2D, there is no need for rotation of the signal. We are looking at the maximum bandwidth per dimension. Thus, the curvature matrix becomes a curvature coefficient κ, the maximum of the diagonal elements of the curvature matrix. The 2D curvature matrix operator is:

$$\mathbf{C}_\kappa = \begin{bmatrix} 1 & \kappa \\ 0 & 1 \end{bmatrix}$$

BRDF and texture The resulting bandwidth vector after the reflection of the input local light-field by the surface is:

$$\vec{b'} = \begin{bmatrix} b_x + t \\ \min(b_\theta, m_\theta) \end{bmatrix}$$

Where t is the texture bandwidth and m_θ is the BRDF bandwidth.

Scaling After the scaling (e.g., for projection) of the input spectrum, the bandwidth vector of the new spectrum is:

$$\vec{b'} = \mathbf{P}_\alpha \vec{b}$$

Where the scaling matrix is:

$$\mathbf{P}_\alpha = \begin{bmatrix} \alpha & 0 \\ 0 & 1 \end{bmatrix}$$

4.1.1.2 Discussion

The bandwidth vector has a compact formulation that makes it practical. Furthermore, the matrix formulation is interesting as it allows a clean algebraic definition of the operators.

But the bandwidth vector does not capture anisotropy. The notion of bandwidth of a Monte-Carlo estimate is not theoretically sound (But it works for the case of one bounce without visibility). It is thus not applicable for generic applications with the motion operator.

4.1.2 The Wedge Function

The wedge function was first introduced by Chai et al. [21] for image based rendering using plenoptic sampling, then used by Egan et al. [49, 51, 50] in the study of linear motion of shadows, texture, and occlusion. This tool gives a better intuition of the bounding shape of the spectrum than the bandwidth estimation when the spectrum studied is known to be bounded by two shears of the same 1D input spectrum.

4.1.2.1 Definition of the Wedge Function

We take the example of the travel of the signal in space. We characterize, in the Fourier domain, a geometrical bounding of the spectrum between distance of travel d_1 and d_2 with $d_1 < d_2$. If the input signal is a dirac in angle and a constant in space, we end up with a spectrum enclosed by the wedge function (Figure 4.2).

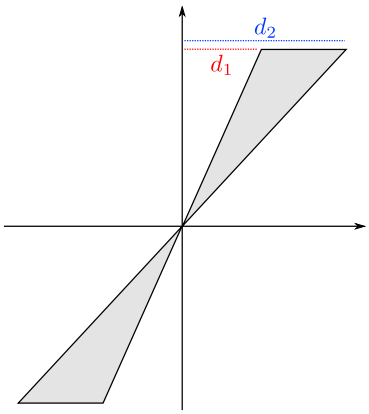


Figure 4.2 *– The wedge function defined as the space covered by a continuous shear from d_1 and d_2. If the input spectrum is one dimensional, before the shear, we can define a tight bound of the possible spectrum.*

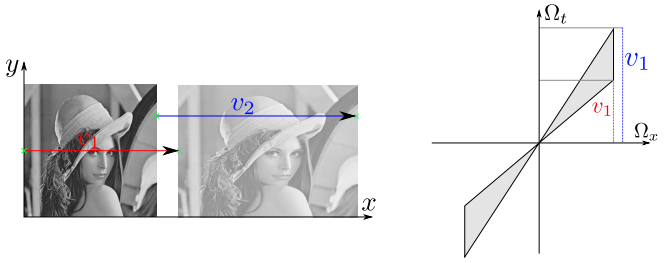
Egan et al. apply this analysis to the study of linear motions such as: texture motion, shadow motion and environment maps rotations [49], and extended this approach to directional occlusion analysis [51, 50]. Each analysis requires a special case formula, but all resulting spectrum share the same wedge enclosing. Having such knowledge of a tight bounding allows to define an optimal packing of samples.

4.1.2.2 Applications

This wedge analysis relies on a particular configuration, namely continuous shears, and cannot be extended to more general cases. This is why, contrarily to other presented tools, we will not derive operators, but cite cases of application.

Motion of a diffuse textured object is studied in the space-time domain $(\delta x, \delta t)$. If $f(\delta x, \delta t_1)$ is the object radiance on screen at the starting position, the translation of the object will result in the sheared radiance function:

$g(\delta x, \delta t) = f(\delta x + v\delta t, \delta t)$. If the motion is not uniform, the resulting spectrum is enclosed inside a wedge area (v_1, v_2) where v_1 is the minimum speed and v_2 is the maximum speed. (Figure 4.3).

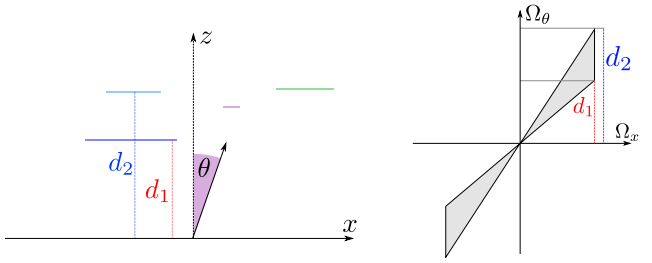


(a) Moving textured object

(b) Associated wedge function

Figure 4.3 *– The wedge function describes the area in which the moving texture object's spectrum lies.*

Occlusion consider the case of multiple planar occluders. If the occluders are bounded in depth where d_1 is the first occluder depth and d_2 is the last occluder's depth the resulting occlusion spectrum is contained inside the wedge (d_1, d_2) in the angle-space domain (Figure 4.4)



(a) A set of planar occluders

(b) Associated wedge function

Figure 4.4 *– The wedge function describes the area in which the occluders spectrum is enclosed.*

4.1.2.3 Discussion

The wedge function is limited to special cases of light transport. In those configurations, it is a powerful tool as its evaluation is simple. But we aim to perform a generic analysis in a context of global illumination.

4.1.3 Density Estimation of the Spectrum

This method evaluates the spectrum using a density estimation approach. Much like the photon mapping method, samples are drawn in the frequency

domain with probability density equals to the input local light-field's spectrum (Figure 4.5). Samples are updated individually for each operation on the light-field.

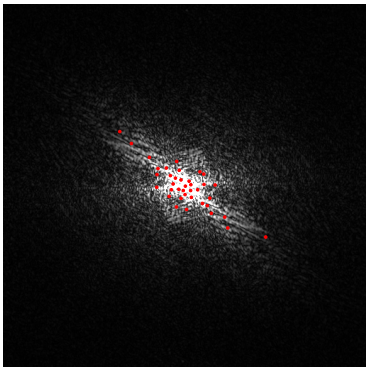


Figure 4.5 – *The spectrum is densely sampled and the bandwidth is estimated from this set of points.*

Soler et al. [164] used this method to estimate the local light-field spectra on the lens and on the sensor for a one bounce reflexion assuming that the incoming light-field before the reflexion was of infinite angular frequency.

4.1.3.1 Application of the Different Operators

The operators are defined for sample positions $\vec{p}_i$ individually. To apply an operator, one has to loop over all the samples and apply the individual operator.

Linear operations Linear operations are applied like for the bandwidth operator. Each position is multiplied by the operator's matrix:

$$\vec{p}_i' = M_{Op}\,\vec{p}_i$$

Where M_{Op} is the matrix of the operator.

Occlusion The convolution with the occluder's window is done by summing the spatial position of the sample with a random position $\vec{o}_\epsilon$ drawn from the PDF of the occluder.

$$\vec{p}_i' = \vec{p}_i + \vec{o}_\epsilon$$

This property comes from the fact that the random variable of the convolution of two PDF is the sum of the individual random variables of the PDFs (Figure 4.6).

BRDF The multiplication with the spectrum of the BRDF is done by clamping samples that are over the angular bandwidth of the BRDF, b_θ. A sample is discarded if the sample angular position $p_{i\theta} > b_\theta$.

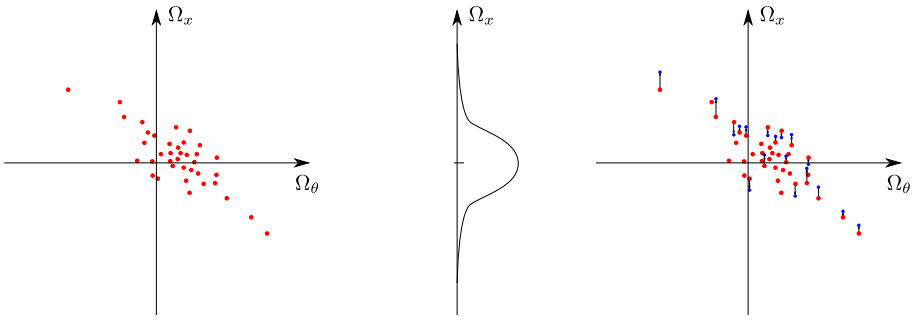


Figure 4.6 *– Convolution is done by moving the input positions (right figure) by random displacements along x (left figure) sampled from the occluder's spectrum (center figure).*

4.1.3.2 Discussion

While it was not described in their paper (as they derived a 2D version of the spectrum), the distribution of points catches the anisotropy of the spectrum. This information is needed for applications using the time operator as the projected direction of motion influences the resulting operator.

But a lot of samples are required to obtain a decent estimator of the spectrum. The more bounces the light-path will make, the more samples it will require since each BRDF operation removes some samples. It is thus not applicable for global illumination algorithms.

4.2 The Covariance Matrix

Our goal is to find a compact and efficient structure to determine the spread and orientation of the spectrum after several bounces. None of the existing methods is able to fulfill all of those criterion. We present a new structure: the *Covariance matrix* that has a compact storage, allows a matricial formulation of the operators, estimates anisotropy, as well as variance, of the local spectrum and is coherent with Monte-Carlo integration.

First, we will introduce the notion of covariance for density functions (Section 4.2.1). We will introduce the covariance matrix and demonstrate that it is the smallest set of information capable of recoverng the covariance (Section 4.2.2 and Section 4.2.3). We will define matricial equivalents for the operators (Section 4.2.4) and validate experimentally the covariance matrix estimate with measured covariance matrix (Section 4.2.5). Finally, we will discuss the relation between the covariance matrix and previous works on Gaussian beams in the physics community (Section 4.2.6).

4.2.1 Covariance of Density Functions : Definition

The covariance of a density function $f : \mathbb{R}^N \to \mathbb{R}$ with zero mean for vector $\vec{x} \in \mathbb{R}^N$ against vector $\vec{y} \in \mathbb{R}^N$ is defined as:

$$\mathrm{cov}_{\vec{x},\vec{y}}(f) = \int_{\vec{t}\in\mathbb{R}^N} \langle \vec{t}, \vec{x} \rangle \langle \vec{t}, \vec{y} \rangle f(\vec{t}) \, \mathrm{d}\vec{t} \tag{4.2}$$

Where $\langle \vec{x}, \vec{y} \rangle$ denotes the dot product between $\vec{x}$ and $\vec{y}$. The covariance is a generalisation of the variance to higher dimensions. When $N = 1$ and $\vec{x} = \vec{y} = 1$, the covariance is the same as the variance.

Prokhorov [139] was the first to introduce the covariance of a measure using the integral form in real space. It was later generalized to probability density functions defined over vector spaces as we can see it in Vakhania et al.'s monograph [174, Chapter III.2].

The covariance of a density function is the extension of the second moment matrix for a random vector of finite dimension. Given a random variable X with probability density function f with null expectation, the second moment matrix (or covariance matrix of X) is defined as:

$$\Sigma_X = E(XX^T) \tag{4.3}$$

An unbiased estimator of Σ_X is:

$$\begin{array}{rcl} \tilde{\Sigma}_X & = & \dfrac{1}{N} \displaystyle\sum_{i=0}^{N} x_i x_i^T \\ & \underset{N=\infty}{\rightarrow} & \mathrm{cov}(f) \end{array}$$

4.2.2 The Covariance Matrix

4.2.2.1 Definition

We are interested in the smallest set of information that defines the covariance of a given function f for any couple of vectors $(\vec{x}, \vec{y})$. This is embodied by the covariance matrix. The covariance matrix is a $N \times N$ matrix defined for a given basis of $\mathbb{R}^N$, $(\vec{e}_i)_{i \in [1..N]}$.

Definition : The covariance matrix of a density function $f : \mathbb{R}^N \rightarrow \mathbb{R}$, with respect to the basis $(\vec{e}_i)_{i \in [1..N]}$ is a $N \times N$ matrix Σ where:

$$\Sigma_{i,j} = \int_{\vec{t} \in \mathbb{R}^N} \langle \vec{t}, \vec{e}_i \rangle \langle \vec{t}, \vec{e}_j \rangle f(\vec{t}) \, \mathrm{d}\vec{t}, \quad \forall (i,j) \in [1..N]^2 \tag{4.4}$$

In the rest of this thesis, we will use the canonical basis for the covariance matrix and avoid talking about the basis used to define it.

4.2.2.2 What is the Covariance Matrix ?

A Frame Our space of study is $\mathbb{R}^N$ which is Hilbertian, the covariance matrix defines (if it is non degenerated) a frame of the space (Figure 4.7). An eigen-decomposition of it results in N eigen-vectors being the principal directions of the density function and of N eigen-values being the variances of the density function along the associated vectors. Furthermore, this frame is orthogonal because the covariance matrix is symmetric positive (Property 1).

A Notion of Entropy The *differential entropy* of a PDF with covariance Σ is bounded by [34]:

$$h(f) \leq \frac{1}{2} \log \left[(2\pi e)^N \det(\Sigma) \right] \tag{4.5}$$

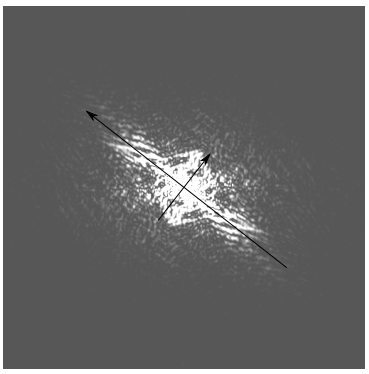


Figure 4.7 *– When the covariance matrix has full rank, it defines a frame of the $\mathbb{R}^N$ space. The length of the frame's vectors are the variances along those vectors. Here we display the eigen-decomposition of the covariance matrix of the amplitude of the Lena picture.*

Where $h(f)$ is the differential entropy, N is the number of dimensions of the input domain of f and Σ its covariance matrix. The differential entropy extends, in information theory, the notion of entropy[3] of random variables to their associated PDF. For example, a high determinant of the covariance matrix will result in a highly varying signal. Samples (light-path associated with this portion of the space) will estimate the average value with a low certainty.

The Hessian matrix The covariance matrix of the amplitude is, up to a sign change, equals to the Hessian of the light-field (Property 3).

4.2.3 Properties

Property 1. *The covariance matrix Σ is symmetric and its diagonal elements are positive.*

It results from the integral definition and the fact that we are looking at the amplitude of the spectrum which has value in $\mathbb{R}+$.

Property 2. *For any couple of vector $\vec{x} \in \mathbb{R}^N$, $\vec{y} \in \mathbb{R}^N$ the covariance of a given function f for $\vec{x}$ against $\vec{y}$ is:*

$$cov_{\vec{x},\vec{y}}(f) = \vec{x}_e^T \, \Sigma \, \vec{y}_e$$

Where Σ is the covariance matrix of f with respect to the canonical basis $(\vec{e}_i)$ and $\vec{x}_e$ is the projection of $\vec{x}$ on it.

This property is important since for any rotation in $\mathbb{R}^N$, we can find the new covariance matrix in the new basis. The definition of the covariance matrix Σ defined with respect to the basis (e_i) in the basis (e'_i) (both are orthogonal and normed) is:

$$\Sigma' = R^T \Sigma R \tag{4.6}$$

Where R is the rotation matrix from (e_i) to (e'_i).

[3]The entropy was defined by Shannon as the uncertainty of information in a message per unit of information [160].

Proof. As $(\vec{e}_i)$ is a basis of $\mathbb{R}^N$ we can define $\vec{x}$, and $\vec{y}$, using its decomposition on this basis:

$$\vec{x} = \sum_i \langle \vec{x}, \vec{e}_i \rangle \vec{e}_i \tag{4.7}$$

$$\begin{aligned}
\mathrm{cov}_{\vec{x},\vec{y}}(f) &= \int_{\vec{t}\in\mathbb{R}^N} \langle \vec{t}, \vec{x} \rangle \langle \vec{t}, \vec{y} \rangle f(\vec{t}) \, \mathrm{d}\vec{t} \\
&= \int_{\vec{t}\in\mathbb{R}^N} \langle \vec{t}, \sum_i x_i \vec{e}_i \rangle \langle \vec{t}, \sum_j y_j \vec{e}_j \rangle f(\vec{t}) \, \mathrm{d}\vec{t} \\
&= \sum_j \sum_i x_i y_j \int_{\vec{t}\in\mathbb{R}^N} \langle \vec{t}, \vec{e}_i \rangle \langle \vec{t}, \vec{e}_j \rangle f(\vec{t}) \, \mathrm{d}\vec{t} \\
&= \sum_j \sum_i x_i y_j \Sigma_{i,j} \\
&= \vec{x}_e^T \Sigma \vec{y}_e
\end{aligned}$$

□

Property 3. *The covariance of a particular direction against itself of the Fourier transform of a function $\hat{f}$ correspond to the second partial derivative of f at the position where the covariance is computed.*

$$cov_{\vec{x},\vec{x}}(\hat{f}) = \frac{d^2 f}{d\vec{x}^2}(\vec{0})$$

Proof. This identity comes from the fact that studying the multiplication of the Fourier transform $\hat{f}(\mu)$ by $i\mu$ is equivalent to study the Fourier transform of the derivative of f. Given the covariance matrix definition (Equation 4.4), we see that it correspond to the value of the second derivative at the central position:

$$\begin{aligned}
\Sigma_{i,j} &= \int_{\vec{\mu}\in\mathbb{R}^N} \mu_i \mu_j \big|\hat{f}\big|(\vec{\mu}) \, \mathrm{d}\vec{\mu} \\
&= \int_{\vec{\mu}\in\mathbb{R}^N} \Big|\mathcal{F}\big[\frac{d^2 f}{dx_i dx_j}\big]\Big|(\vec{\mu}) \, \mathrm{d}\vec{\mu} \\
&= \Big|\frac{d^2 f}{dx_i dx_j}\Big|(\vec{0})
\end{aligned}$$

□

Property 4. *The covariance matrix of the weighted sum of two density functions is the weighted sum of the respective covariance matrices of the density functions.*

$$\Sigma(\alpha f + \beta g) = \alpha' \Sigma(f) + \beta' \Sigma(g)$$

Where $\alpha' = \alpha \dfrac{||f||}{||\alpha f + \beta g||}$ *and* $\beta' = \beta \dfrac{||g||}{||\alpha f + \beta g||}$

Proof.

$$\begin{aligned}\Sigma_{i,j}(\alpha f + \beta g) &= \int_{\vec{t}\in\mathbb{R}^N} \langle \vec{t}, \vec{e}_i\rangle\langle \vec{t}, \vec{e}_j\rangle\big(\alpha f(\vec{t}) + \beta g(\vec{t})\big)\,\mathrm{d}\vec{t}, \quad \forall (i,j) \in [1..N]^2 \\ &= \alpha' \int_{\vec{t}\in\mathbb{R}^N} \langle \vec{t}, \vec{e}_i\rangle\langle \vec{t}, \vec{e}_j\rangle f(\vec{t})\mathrm{d}\vec{t} + \beta' \int_{\vec{t}\in\mathbb{R}^N} \langle \vec{t}, \vec{e}_i\rangle\langle \vec{t}, \vec{e}_j\rangle g(\vec{t})\mathrm{d}\vec{t} \\ &= \alpha'\Sigma_{i,j}(f) + \beta'\Sigma_{i,j}(g)\end{aligned}$$

□

Property 5. *We can build a Monte-Carlo estimate of the covariance matrix using Property 4.*

Proof. Let I be a function resulting from the integration of a positive function f over one of its variable.

$$I(x) = \int_y f(x,y)\mathrm{d}y$$

We would like to estimate the covariance matrix of I, but we only have access to the covariance matrix of f.

$$\Sigma_{i,j}(I) = \int_x x_i x_j \frac{I(x)}{||I||}\mathrm{d}x$$

If we have only access to a Monte-Carlo estimate of the function I, we cannot estimate its covariance matrix. But we can provide a Monte-Carlo estimate of the covariance matrix of I from the Monte-Carlo estimate of I.

$$\begin{aligned}\Sigma_{i,j}(I) &\simeq \int_x x_i x_j \frac{\sum_{y_k} f(x,y_k)}{||I||}\mathrm{d}x \\ &= \sum_{y_k} \frac{1}{||I||}\int_x x_i x_j f(x,y_k)\mathrm{d}x \\ &= \sum_{y_k} \frac{||f_k||}{||I||}\Sigma_{i,j}(f_k)\end{aligned}$$

□

Property 6. *The covariance matrix of the convolution of two independent density functions if the sum of the covariance matrices of the two functions.*

$$\Sigma(f \otimes g) = \Sigma(f) + \Sigma(g) \tag{4.8}$$

Proof. Recall that $\Sigma(f) \simeq \sum_i x_i x_i^T$, when $x \sim f$ (if f has zero mean). For this proof, we use the property that a random variable whose PDF is a convolution can be expressed as a product of uncorrelated random variables:

$$X_{f\otimes g} = Y_f + Z_g$$

If we express the unbiased estimator of $\Sigma(f \otimes g)$, we obtain:

$$\Sigma(f \otimes g) = \sum_{i\in\mathbb{N}} x_i x_i^T$$

But we can express x_i as a draw of two independent random variables y_i and z_i:

$$\begin{aligned}\Sigma(f \otimes g) &= \sum_{i\in\mathbb{N}} (y_i + z_i)(y_i + z_i)^T \\ &= \sum_{i\in\mathbb{N}} y_i y_i^T + z_i z_i^T + y_i z_i^T + z_i y_i^T\end{aligned}$$

The last terms $\sum_{i\in\mathbb{N}} y_i z_i^T$ and $\sum_{i\in\mathbb{N}} z_i y_i^T$ are the correlation terms and are equal to zero. Thus the estimator of the covariance matrix of the convolution is:

$$\begin{aligned}\Sigma(f \otimes g) &= \sum_{i\in\mathbb{N}} y_i y_i^T + \sum_{i\in\mathbb{N}} z_i z_i^T \\ &= \Sigma(f) + \Sigma(g)\end{aligned}$$

□

Property 7. *The covariance matrix of a density function whose input space is linearly transformed is equal to:*

$$\Sigma' = |A| A^T \Sigma A \tag{4.9}$$

Where A is the linear transformation of the input space, and $|A|$ is its determinant.

Proof. Let f_A be the transformed function from f such as $f_A(\vec{x}) = f(A^{-1}\vec{x})$:

$$\begin{aligned}\Sigma_{i,j}(f_A) &= \int_{\vec{t}\in\mathbb{R}^N} t_i t_j f(A^{-1}\vec{t})\mathrm{d}\vec{t} \\ &= |A| \int_{\vec{t}\in\mathbb{R}^N} (A\vec{t})_i (A\vec{t})_j f(\vec{t})\mathrm{d}\vec{t} \\ &= |A| \int_{\vec{t}\in\mathbb{R}^N} \Big(\sum_k A_{k,i} t_i\Big)\Big(\sum_l A_{l,j} t_j\Big) f(\vec{t})\mathrm{d}\vec{t} \\ &= |A| \sum_k \sum_l A_{k,i} A_{l,j} \int_{\vec{t}\in\mathbb{R}^N} t_i t_j f(\vec{t})\mathrm{d}\vec{t} \\ &= |A| \sum_k \sum_l A_{k,i} A_{l,j} \Sigma_{i,j}(f) \\ &= |A| (A^T \Sigma(f) A)_{i,j}\end{aligned}$$

□

Property 8. *The covariance matrix of a partially integrated signal is approximated by the slice of the covariance matrix of the signal un-integrated where the missing dimensions are the integrated dimensions. The sliced covariance matrix along dimension i, $\Sigma_{|i}$ of the covariance matrix Σ is defined as:*

$$\Sigma_{|i} = \left[S_i^T \Sigma^{-1} S_i\right]^{-1} \tag{4.10}$$

Where S_i is the 5×4 matrix composed of vectors of the remaining basis.

$$S_i = \begin{bmatrix} \vec{e}_1 \\ \vdots \\ \vec{e}_{i-1} \\ \vec{e}_{i+1} \\ \vdots \\ \vec{e}_n \end{bmatrix} \tag{4.11}$$

Proof. We build this approximation by looking at the equivalent Gaussian, g_Σ. We apply Fourier's slice theorem (Section 3.2.3.3) on the equivalent Gaussian. The equivalent Gaussian of the integrated signal is:

$$g_{\Sigma_{|i}}(\vec{x}) = g_{\Sigma(}\vec{x}_{|i}) \quad \text{where } (x_{|i})_j = x_j \text{ if } j \neq i \text{ and 0 else} \tag{4.12}$$

This is equivalent to removing the i^{th} column and row of the inverse matrix in the Gaussian formulation:

$$g_\Sigma(\vec{x}) = e^{-\vec{x}^T \Sigma^{-1} \vec{x}} \tag{4.13}$$

But this affects the inverse of the covariance matrix. Thus to obtain the reduced covariance matrix, we need to invert it, remove the i^{th} column and row and invert it again to obtain a sliced covariance matrix. □

This property requires the covariance matrix Σ to be non-degenerate.

4.2.4 Application to Light Transport Operators

4.2.4.1 Travel in Space

The shear is a linear operator. Any linear operator can be represented by its product with the operator matrix (Property 7):

$$\Sigma\big(\mathbf{T}_d(\hat{l})\big) = S_d^T \Sigma\big(\hat{l}\big) S_d \tag{4.14}$$

Where S_d is the transport shear matrix for a distance of d meters. This matrix has the form:

$$S_d = \begin{bmatrix} 1 & 0 & -d & 0 & 0 \\ 0 & 1 & 0 & -d & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.15}$$

4.2.4.2 Occlusion

The covariance matrix of a convolution of two density is the sum of their respective covariance matrices (Property 6). Given the covariance matrix of the visibility spectrum O, the resulting covariance matrix is:

$$\Sigma\big(\mathbf{O}(\hat{l})\big) = \Sigma\big(\hat{l}\big) + \Sigma_O \tag{4.16}$$

For example, under the planar occluder hypothesis, the occluder matrix is:

$$\Sigma_O = \begin{bmatrix} O_{xx} & O_{xy} & 0 & 0 & 0 \\ O_{yx} & O_{yy} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.17}$$

4.2.4.3 Rotation and Scale

Rotation and scale are linear operators (Property 7):

$$\Sigma\big(\mathbf{Rot}_\alpha(\hat{l})\big) = R_\alpha^T \Sigma\big(\hat{l}\big) R_\alpha \tag{4.18}$$

$$\Sigma\big(\mathbf{P}_a(\hat{l})\big) = S_a^T \Sigma\big(\hat{l}\big) S_a \tag{4.19}$$

Where R_α is the rotation matrix of α radian. This matrix has the form:

$$R_\alpha = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 & 0 \\ 0 & 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.20}$$

S_a is the scaling rotation matrix of factor a. This matrix has the form:

$$S_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & a & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.21}$$

4.2.4.4 Cosine term

The cosine term operator is a convolution. We use the covariance convolution formulation (Property 6) between the incoming light-field covariance and the covariance of the spherical Bessel function of the first kind (noted Σ_B):

$$\Sigma\big(\mathbf{C}(\hat{l})\big) = \Sigma\big(\hat{l}\big) + \Sigma_B \tag{4.22}$$

4.2.4.5 Reflection

We present the matrix operator for isotropic Phong parametrized BRDFs. This operator is the multiplication of the input light-field's spectrum and the BRDF's spectrum. We approximate it using the formula for zero-centered Gaussians multiplication. When we multiply two zero-centered Gaussians, the resulting distribution is a Gaussian whose inverse covariance matrix is the sum of input inverse covariance matrices.

We need to take extra care for this operator since the covariance matrices might be low rank when the distribution is a Dirac along one dimension. We use pseudo-inverses instead of inverses in our derivations (noted with a plus sign). We approximate the resulting covariance matrix using:

$$\Sigma(\hat{l} \times \hat{\rho}) \simeq \left[\Sigma(\hat{l})^+ + \Sigma(\hat{\rho})^+\right]^+ \tag{4.23}$$

Where $\Sigma(\hat{\rho})$ is the covariance matrix of the BRDF. For example, the covariance matrix of a Phong lobe is (proof in Appendix B.1):

$$\Sigma(\hat{\rho}_s) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{s}{4\pi^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.24}$$

This formulation is consistent with previous work on frequency analysis of reflection [140].

4.2.4.6 Transmission

Specular transmission is handled by a convolution with the window kernel, followed by a scale of the angular domain. Equation 3.58 is translated in term of covariance matrix by a addition with an angular block matrix followed by a linear transform:

$$\Sigma\big(\mathbf{Tr}(\hat{l})\big) = S^T\Big(\Sigma\big(\hat{l}\big) + \Sigma_W\Big)S \tag{4.25}$$

Where Σ_W, the covariance matrix of angular window is empty everywhere, excepted for the second angular coordinate:

$$\Sigma_W = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w_{\Omega_\phi} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.26}$$

The scale matrix S is defined as:

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{n_1\cos(i_1)}{n_2\cos(i_2)} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.27}$$

Rough refraction is done like the BRDF operator (Equation 4.23).

4.2.4.7 Lens

The two shears formulation (Equation 3.64) is a linear operator:

$$\mathbf{L}_{(d_1,d_2)}\big(\Sigma\big) = L^T\Sigma L \tag{4.28}$$

Where L has the following formulation:

$$L = \begin{bmatrix} 1+d_1d_2 & 0 & d_1 & 0 & 0 \\ 0 & 1+d_1d_2 & 0 & d_1 & 0 \\ d_2 & 0 & 1 & 0 & 0 \\ 0 & d_2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.29}$$

4.2.4.8 Attenuation

Attenuation has no effect on the covariance matrix if the medium is homogeneous. For varying density participating medium, we use the occlusion formula (Equation 4.16).

4.2.4.9 Scattering

The scattering operator behaves like the BRDF operator (Equation 4.23). But it requires the knowledge of the covariance of the phase function. In this thesis, we study the Henyey-Greenstein phase function [75] (noted HG). The definition of the HG function is:

$$\rho_g(\alpha) = \frac{1}{4\pi} \frac{1-g^2}{(1+g^2-2g\cos\alpha)^{\frac{3}{2}}}$$

This function is one dimensional as this scattering model is isotropic. The HG function has been proposed to model the scattering of light by interstellar matter in the galaxy [75]. The reduced dimensionality of the phase function means that the covariance matrix will also be one dimensional:

$$\Sigma(\hat{\rho}_g) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \text{cov}(|\hat{\rho}_g|) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.30}$$

Property 9. *The covariance of the HG phase function is (See Appendix B.2 for the proof):*

$$cov(|\hat{\rho}_g|) = \frac{3}{4\pi} \frac{|g|(1+|g|)}{(1-|g|)^4} \tag{4.31}$$

Figure 4.8 shows the influence of the g parameter on the frequency content of the scattered radiance. Increasing this parameter towards one reduces the low-pass filter effect of scattering.

4.2.4.10 Motion

The motion matrix operator is a linear shear operator. Its matrix is expressed using a diagonal matrix with correlation terms performing the shear:

$$\mathbf{M}_{\vec{v}}(\Sigma) = M_{\vec{v}}^T \Sigma M_{\vec{v}}$$

Where the operator's matrix is:

$$M_{\vec{v}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ v_x & v_y & v_\theta & v_\phi & 1 \end{bmatrix}$$

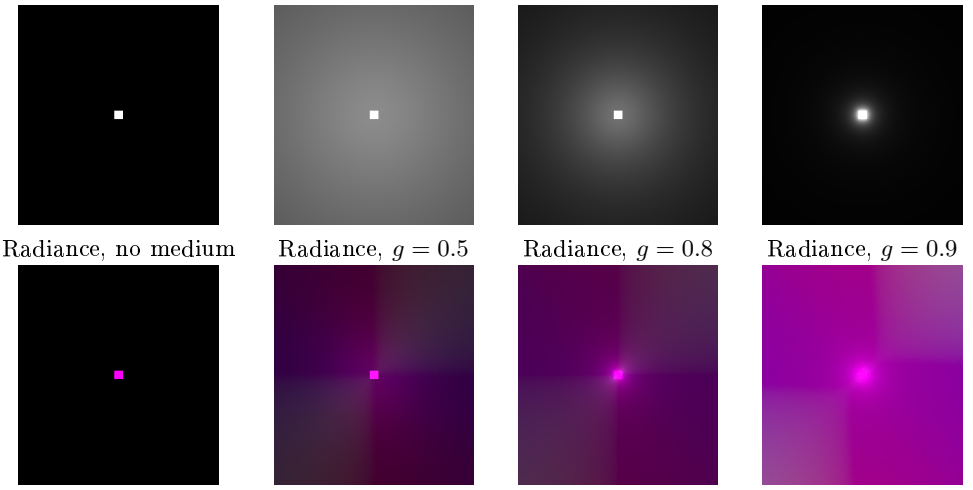


Figure 4.8 *– We illustrate the low-pass effect of scattering. The scene is composed of a square light source that diffuses light in a uniform volume with a Henyey-Greenstein scattering distribution. The camera is placed towards the light to show the extent of the diffusion. The greater the diffusion the less the angular covariance. As we proved, scattering acts as a low pass filter over angular frequencies. The color images show the $\theta\theta$, $\theta\phi$ and $\phi\phi$ covariance values mapped on the RGB channels.*

4.2.5 Validation of the covariance method

In this section, we present validation examples for the our covariance matrix estimate. We designed a 5D light-field ray-tracer and compared the covariance from simulation against our covariance estimate.

The first test scene is composed of a light source directed towards a translating diffuse checkerboard (Figure 4.9). The second test scene is composed of a square light source directed towards a diffuse square receiver partially occluder by a rotated square blocker. Figure 4.10). For this test scene, we used the cone grid to evaluate occlusion (Section 4.3.2.2).

Our covariance prediction are close to the measured covariance. The shear in the time domain is correctly depicted by our prediction (Figure 4.9). The equivalent Gaussian (in dotted) captures the complicated spectrum after the reflection on the sliding reflector. The anisotropy of the signal after occlusion is well estimated (Figure 4.10). The differences between the two matrices are explained by the window function we used to compute the Fourier transforms of the measured light-fields.

4.2.6 Comparison with Gaussian Beams

We emphasis on the novelty of the covariance matrix propagation. To our knowledge, the closest work on matrix propagation of light information is *Gaus-*

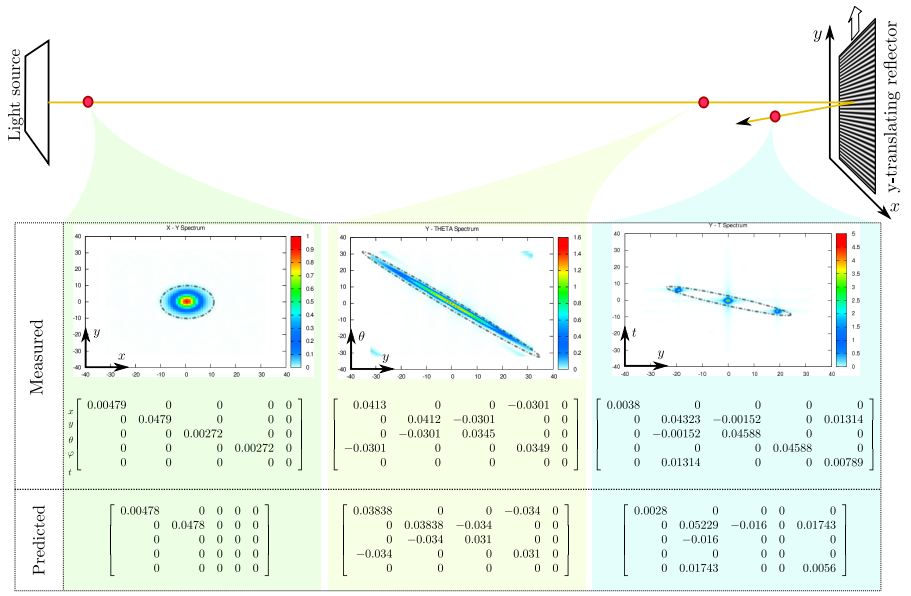


Figure 4.9 *– A Gaussian light source illuminates a moving diffuse reflector with a high frequency texture in the orthogonal direction to the motion. We estimate the covariance at several locations (after the source, before reflection and after reflection) using a light-field ray-tracer and compare the result to our estimated covariance matrix. The results are close and differences can be explained from the window function used before the Fourier transform of the measured data.*

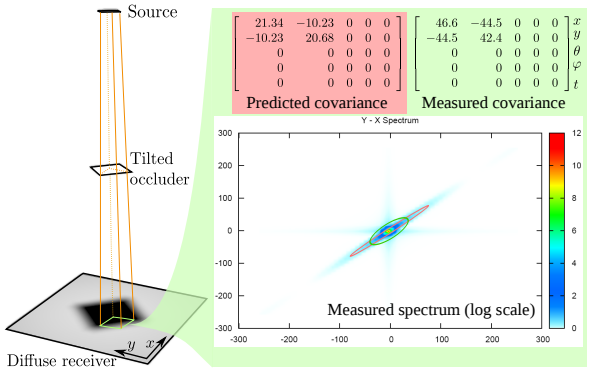


Figure 4.10 *– We analyse a soft shadow casted by a tilted plane. The covariance is estimated after the reflection by the diffuse plane. The resulting spectrum exhibits the same orientation. The measured spectrum is estimated using a windowing of the measured light-field. This windowing increases the frequency content of the spectrum. We can see this difference on the plot (the cross) and on the measured covariance.*

sian beams analysis [2]. In this section, we give a short review of Gaussian beams and highlight the differences between those two tools.

4.2.6.1 Definition of Gaussian beams

To analyse paraxial systems, geometrical optics often approximate the differential irradiance by a Gaussian function [42]: the *Gaussian beams*:

$$\Psi(x,z) = \frac{4}{\pi^2}\sqrt{\frac{e^{-i(\phi(z)-\phi_0)}}{\omega(z)}}e^{-i\frac{kx^2}{2R(z)}-\frac{x^2}{\omega^2(z)}}$$

Where x is the transversal dimension, and z is the axial dimension (in a 2D setting). k describe the wavelength of the beam and $R(z)$, $\phi(z)$, ϕ_0 and $\omega(z)$ describe the behaviour of the beam (dispersion, width and waist position).

The propagation of a Gaussian beam through an optical system can be studied using matrix multiplications [2]. A Gaussian beam propagating through a thin lens keeps its Gaussian nature [43]. Those properties are interesting to model and test optical systems.

4.2.6.2 Comparison with Covariance

Covariance matrices exhibit a richer information than Gaussian beams. The former characterizes the spectrum of the radiance function while the later characterizes the spatial distribution of irradiance.

Covariance matrix allows to analyse paraxial systems as the matricial operators such as refraction and travel are dual of the paraxial operators [60]. Complex optical systems can be formulated using the set of operators as well multiple inputs systems thanks to the additivity of covariance. The later is not possible with the definition of Gaussian beams.

4.3 Occluder Spectrum Evaluation

We saw various methods to represent or approximate the frequency spectrum of the light-field function. Yet the occlusion operator requires the corresponding representation of its frequency spectrum. We need to evaluate it during the construction of the light-path.

We first review existing approaches of occlusion evaluation (Section 4.3.1). Then we propose two new data structure to obtain local occlusion information without altering the core of the ray-tracing engine (Section 4.3.2).

4.3.1 Previous Works

Previously proposed approaches require either sampling of visibility (Section 4.3.1.1) or its rasterization (Section 4.3.1.2).

4.3.1.1 Sampling Occlusion

Durand et al. [47] used a sampling of the visibility to estimate the occlusion for directions on the hemisphere of a particular hit-point with the scene.

This method requires an appropriate sampling of the hemisphere It has to be done for each ray (but it can be precomputed for eye bounce). This is not suited for a global illumination setting as visibility evaluation is a bottleneck.

4.3.1.2 Depth Map Discontinuities

Soler et al. [164] evaluated the visibility using a depth map. The visibility is rasterized which is cheaper than Durand et al. [47]. The evaluation of the directionality (when needed) is made using the gradient direction in the tile used to detect the discontinuity (for example a 3×3 pixels tile).

This method is not adapted for a general purpose, such as global illumination. Gathering local occlusion information for a given ray requires to rasterize occluders.

4.3.2 Voxel Grids

We incorporate occlusion information into a spatial structure to evaluate it during a marching step. The data structures are two voxel grids where voxels store a distribution information of the nearby occlusion.

The voxel grid has been developed for algorithms that look at multiple reflections analysis or when a depth map cannot be evaluated (for environment maps for example). This spatial structure subdivide the 3D scene into voxels in which we compute a conservative approximation of the occlusion frequency spectrum with the given spectrum representation.

During the covariance computation step, for a given ray, we ray march into this grid, using [4], to estimate the minimum occlusion window.

4.3.2.1 Normal voxel Grid

Description: The normal voxel grid stores a Gaussian representation of the normal density distribution (noted NDF) inside the voxels.

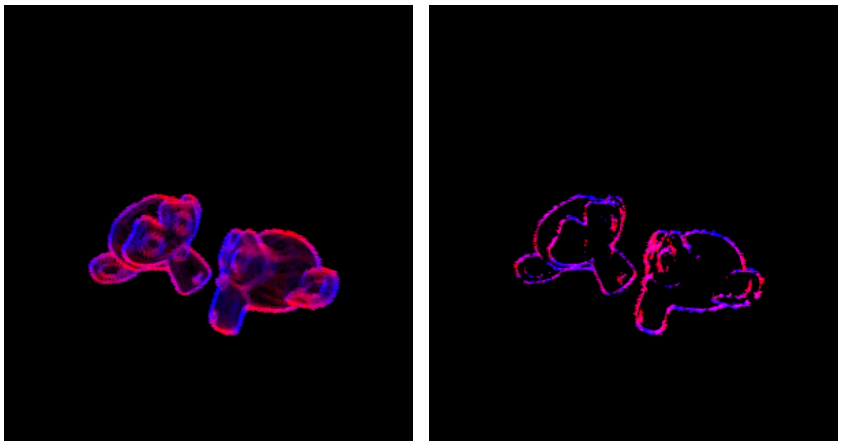
Construction: For each object inside the voxel grid, we sample positions according to the area of objects. To each position we compute the associated normal. We define its covariance matrix using a dirac function in the direction of the normal. To obtain the covariance matrix of the normal distribution, we sum the individual covariance matrices of the normals. We also store the distance to the closest sample point with the center of the voxel.

Evaluation of covariance is done by slicing from the 3D covariance matrix of normals using the tangent plane of the ray's direction. The matrix is inverted and rotated to align the ray's direction with the third component of the matrix. Then, the 2D submatrix of the tangent plane is extracted and inverted to obtain the covariance matrix of the occluder. From the normals' covariance matrix Σ_N, and a ray's direction $\vec{d}$, we define the occluder's covariance matrix Σ_O as:

$$\Sigma_O^{-1} = \left[R_{\vec{d}}^T {\Sigma_N}^{-1} R_{\vec{d}}\right]_{1,2} \tag{4.32}$$

This slicing is not equivalent to an integration, as we store a Gaussian representation of the NDF and not a spectrum. It extracts the distribution of direction inside the tangent plane of the ray.

Frequency leaking: Using a covariance matrix to represent the distribution of normals smoothes it. This introduces leaking of frequency when evaluating the covariance for ray pointing in directions close to the normal of a surface. This leaking creates an undesirable effect: auto-occlusion of surfaces (See Figure 4.11(a)). This auto-occlusion is not problematic for one bounce illumination applications but in a global illumination setting, the auto-occlusion accumulates leading to overestimation of occlusion. To avoid this effect, we introduce another voxel grid: the *cone voxel grid*.



(a) Using a covariance voxel grid **(b)** Using a cone voxel grid

Figure 4.11 *– Auto-occlusion (a) produces an overestimate of the local frequency. Using a cone voxel-grid (b) allows to remove this issue.*

4.3.2.2 Cone Voxel Grid

The cone grid stores the cone in which the NDF lies. This is coarser representation of the NDF, but in our experiences, it gives a cleaner frequency estimate of the occlusion for the auto-occlusion issue.

Definition: The cone grid stores a bi-cone in each voxel. A bi-cone is defined with a mean direction and an aperture. To test if a ray is occluded by a nearby cone, we test for the intersection between the cone and the orthogonal plane of the ray. It allows to get rid of some auto-occlusion issues (Figure 4.11(b)).

Construction: For each object inside the voxel grid, we sample normals. We first construct the mean direction of the cone by computing the mean normal and then calculate the extent of the cone using the normal with the smallest dot product to the mean direction. We also store the distance to the closest sample point with the center of the voxel.

Evaluation of covariance is done first by testing for intersection between the orthogonal plane and the cone (Figure 4.12), then the mean direction is

projected onto the tangent plane to estimate the direction of occlusion. Finally, we weight this matrix by the distance factor.

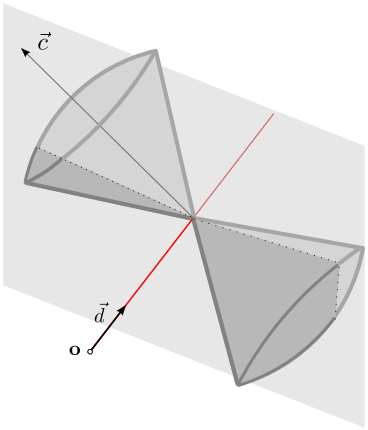
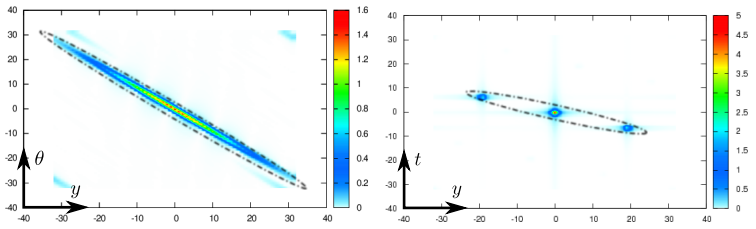


Figure 4.12 *– We test the occlusion of a cone by intersecting the tangent plane of the ray with the cone. The main direction $\vec{c}$ is then projected on the tangent plane to evaluate the direction of occlusion.*

4.4 Notes on Uni-modality

All representations of the spectrum presented in this chapter estimate a mono-modal spectrum. It means that a spectrum consisting of several distinct components will be treated as a whole (Figure 4.13). This is not a problem in practice as we are interested in the complete spectrum. But it could be interesting to separate near-diffuse and specular components of a signal for example.



(a) Mono-modal spectrum **(b)** Multi-modal spectrum

Figure 4.13 *– Our analysis is mono-modal. We obtain the information about the complete spectrum using a covariance analysis for example (a). But a spectrum can be composed of distinct frequency elements on which a multi-modal analysis could be beneficial (b). In the case presented here, there is a low frequency component around the DC and an anisotropic high frequency component with a close to Gabor type.*

It is possible to perform a multi-modal analysis of the spectrum with covariance, assuming that different modes corresponds to different light-paths

(and that we have the knowledge of which mode a light-path correspond to). The additivity of covariance allows to extract the different modes. But such method would require an extra storage cost since multiple covariance matrices would be required to estimate one spectrum.

5 | Applications of Frequency Analysis of Light Transport

We showed how to derive local frequency content around a light-path using a set of operators on 5D radiance functions (Chapter 3). We adapted this analysis into a set of computationally tractable operations on a generic representation, the *covariance matrix* (Chapter 4). In this chapter, we look at applications of the covariance matrix. We show its usefulness for both Monte-Carlo integration and kernel density estimation algorithms and prove that it can handle various light phenomena like depth-of-field, motion blur and participating media.

In this chapter we present the following contributions:

We present applications of the covariance analysis. Those applications intend to accelerate the convergence of either Monte-Carlo integration or kernel density estimation methods. More precisely, we present:

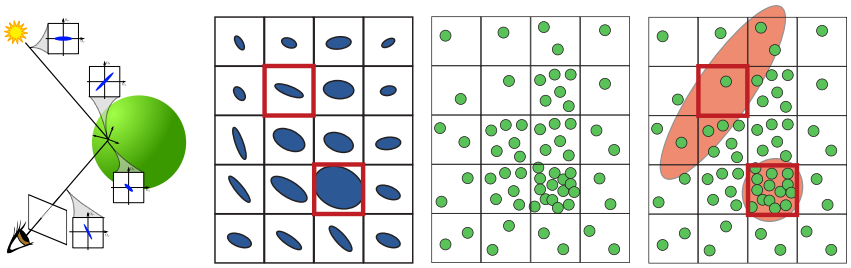
- A new image space adaptive sampling and filtered reconstruction algorithm that takes anisotropic variations of the integrand into account (Section 5.1).
- Methods to perform filtering in object space based on kernel density estimation (Section 5.2).
- Methods to perform adaptive sampling of crepuscular rays and adaptive filtering of photon beams using a volume irradiance spectrum estimate (Section 5.3).

5.1 Image Space Applications

To prove its usefulness, we use the covariance matrix (Section 4.2) in an adaptive sampling and sparse reconstruction using gather filters method. We benefit from the spatial, angular and time analysis by filtering distributed effects (e.g., depth-of-field, motion blur, and soft shadows).

5.1.1 Algorithm

Our algorithm decomposes into four steps: First, we trace a small amount of light-paths from the light to the camera and estimate the associated 5D covariance for each pixel of the image (Figure 5.1a). We store the resulting covariance



(a) Estimation of covariance along a path **(b)** The covariance buffer **(c)** Samples, based on estimated density **(d)** Anisotropic filtering using the covariance matrix

Figure 5.1 *– We estimate covariance matrices from a set of light-paths (a). We store them in a covariance buffer of the size of the screen containing one covariance matrix per pixel (b). This covariance buffer predicts a required density of samples (c) and the associated filter (d). The number of required density of samples can be below* 1.

matrices in a *covariance buffer*. Then, we evaluate the covariance of the 2D image signal's spectrum from the 5D covariance matrices (Figure 5.1b). For each pixel, we estimate a sampling rate from the 5D covariance and a reconstruction filter from the 2D covariance. We sample according to the estimated density (Figure 5.1c). Finally, we reconstruct the final image with a gather algorithm and the estimated filter (Figure 5.1d).

5.1.1.1 The covariance buffer

We store covariance in a 2D buffer with the same resolution as the picture. This buffer contains a single covariance matrix per pixel. In a first pass, we accumulate a fixed number of covariance matrices per pixel (in the covariance buffer) by sampling light-paths and estimating the associated covariance matrix at the sensor position using our forward analysis (Figure 5.1a). We average the matrices in the same pixel using the Monte-Carlo estimator of the covariance matrix (Property 5).

5.1.1.2 Preintegration

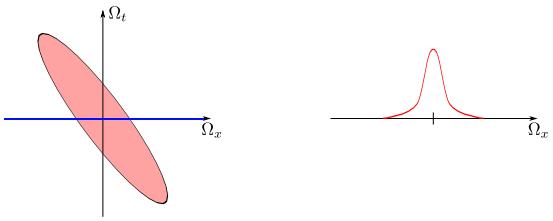
We derive filters in the 2D space of the image to avoid higher dimensional data structures for the samples. For that we express the covariance matrix of the signal after integration with time and lens (Figure 5.2). In the primal, this is done by integrating the input signal multiplied by the time shutter and lens window:

$$l(x,y) = \int_{u,v,t} l(x,y,u,v,t) ws(u,v,t) \mathrm{d}u\mathrm{d}v\mathrm{d}t \tag{5.1}$$

Where ws is the product of the lens window and the camera time shutter.

The equivalent in Fourier is a convolution with the window spectrum evaluated at the DC:

$$\hat{l}(\Omega_x, \Omega_y) = \left[\hat{l} \otimes ws\right](0,0,0) \tag{5.2}$$



(a) Input spectrum (b) Sliced spectrum

Figure 5.2 *– We want to estimate the spectrum of the signal (here in space-time) along the spatial axis, after integration of the time domain ((a) in blue). We apply the slice theorem and evaluate the covariance matrix based on a Gaussian approximation.*

Given the covariance matrix of the 5D input light-field, the covariance matrices of the resulting image on the sensor is estimated by summing the input covariance matrices with the window covariance matrix and slicing them to keep only the spatial components:

$$\Sigma_{i,j} = \left[\Sigma + \Sigma_{ws}\right]_{0_\theta, 0_\phi, 0_t} \tag{5.3}$$

5.1.1.3 Sampling rate and Filter

Once the buffer is populated with 5D and 2D covariance matrices, we estimate the required number of samples and the associated filter. We derive the number of samples of a pixel using the associated 5D covariance matrix and derive the filter using the 2D covariance matrix. Using the 2D covariance matrix to derive sampling rate would require to integrate the time and lens for each image sample. As we want to distribute the integration of those effects, we use the 5D sampling rate.

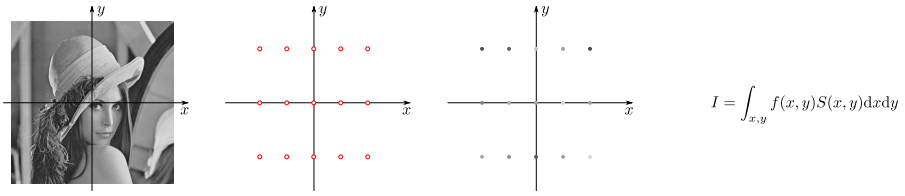
Sampling rate: To express Monte-Carlo integration in the Fourier domain, we reformulate it as a two step process working on continuous functions [46]. Given an integrand f and a distribution of samples S, we rewrite the Monte-Carlo integration as:

$$I_n = \sum_{x_i \in S_n} w_i f(x_i) = \int_{x \in A} S_n(x) f(x) \mathrm{d}x \tag{5.4}$$

Where w_i is the weight associated with the sample x_i, and S_n is either the set of samples or the associated function defined as:

$$S_n(x) = \sum_{x_i \in S_n} w_i \delta(x - x_i)$$

Figure 5.3 presents this concept for the integration of the Lena image. This example assumes however that the sampling function is an infinite dirac comb. To simplify the demonstration, we keep this hypothesis for the rest of the demonstration.



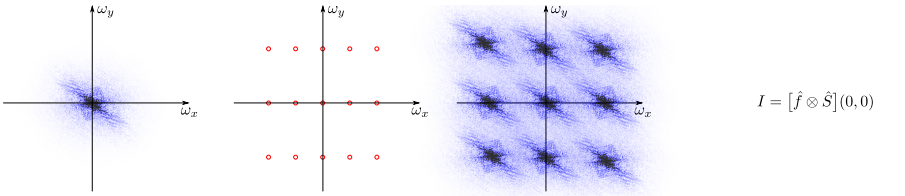
(a) Lena image (b) Sampling pattern (c) Multiplication (d) Integration

Figure 5.3 *– Integration can be seen as a two step operation. Given an integrand (a), given a sampling pattern (b), the resulting Monte-Carlo integral is the integral of the product of those two functions. This continuous formulation allows us to express it in the Fourier domain.*

The Fourier equivalent of Equation 5.4 is the convolution of the two spectrum evaluated at the central position (also called DC term):

$$I_n = \left[\hat{f} \otimes \hat{S}_n\right](0) \tag{5.5}$$

We can interpret aliasing as pollution of the DC term by replicas of the spectrum. This will happen if the samples are not close enough to capture the frequency of the integrand (Figure 5.4).



(b) Integrand spectrum (b) Sampling spectrum (b) Convolution (d) Integration in Fourier

Figure 5.4 *– In the Fourier space, the integration becomes the convolution between the spectrum of the integrand and the spectrum of the sampling function evaluated at the DC.*

We follow the idea of Egan et al. [49]: We assume that the sampling distribution is a dirac comb with a given distance a between the samples. We need to find the optimal a such that the number of samples is minimal and we avoid aliasing (Figure 5.5).

For a given dimension, we need to have our dirac comb (in Fourier) to be separated by the frequency width of the integrand. The covariance gives us this information as it provides the variance of the spectrum along any axis. We use this property to define a sampling rate per unit hypercube from the covariance matrix. The total density of samples in a pixel times the shutter speed times the lens size is given by the square root of the determinant of the covariance matrix multiplied by a constant which converts the standard deviation into spread:

$$\boxed{N = k\sqrt{|\Sigma|}} \tag{5.6}$$

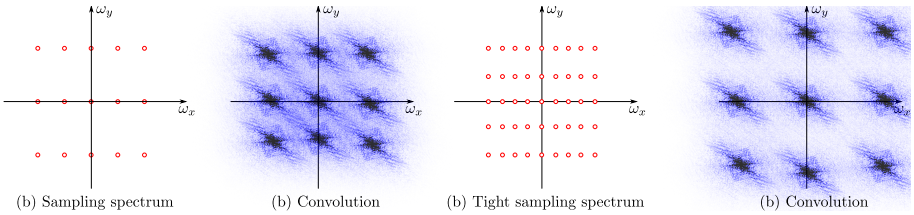


Figure 5.5 *– From the analysis of Monte-Carlo integration in the Fourier domain, we can see two constraints. Assuming we are sampling a dirac comb, the distance between the samples affects the integration. Is the distance is below the frequency extent of the integrand, we create aliasing. But we need to keep the distance between samples as large as possible to reduce costs.*

Since we separate our samples by the spread of the spectrum, the resulting packed spectra overlap each other. But replicas do not overlap the DC term. That is the necessary condition to avoid aliasing.

Filters: Using the 2D covariance matrices, we develop filters that smooth the samples based on the estimated local frequency of the final image. We use Gaussian kernels to smooth our samples because they allow simple derivation of the filters. We derive a filter with same spread and orientation than the signal. The intuition behind this is that we filter the samples in regions where the signal do not vary much. For high variation regions, we rely on adaptive sampling to reduce aliasing.

We use filters with the same covariance matrix than the signal. Gaussians can be formulated from the Mahalanobis distance with respect to the covariance matrix. Furthermore the Fourier transform of a Gaussian is a Gaussian which covariance matrix is inversed:

$$\begin{aligned} g(\vec{p}) &= e^{-d(\vec{p},\Sigma)} \quad \text{where } d(\Sigma,\vec{p}) = \vec{p}^T\Sigma^{-1}\vec{p} \\ \hat{g}(\vec{\mu}) &= \frac{1}{|\Sigma|}e^{-\pi d(\Sigma^{-1},\vec{p})} \end{aligned}$$

$d(\Sigma,\vec{p})$ is called the squared Mahalanobis distance of vector $\vec{p}$, with zero mean, to the matrix Σ. We assume that the covariance matrix C is invertible. It is not always the case, but we have found in practice that adding small quantities to the diagonal almost always leads to an invertible matrix.

The resulting filter for pixel with covariance matrix Σ is then:

$$h(\vec{p}) = Ae^{-\pi\vec{p}^T\Sigma\vec{p}} \tag{5.7}$$

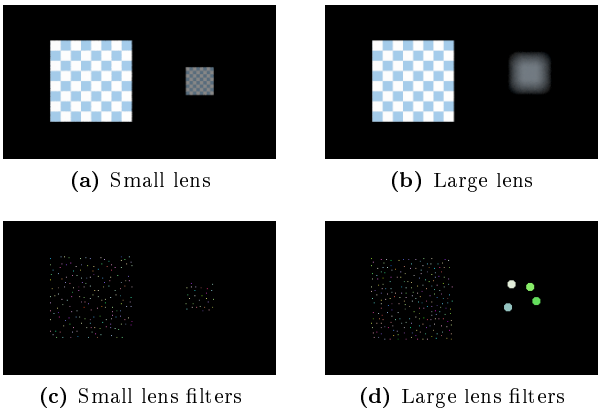
We used a weighted average of the samples using the filter. It allows to avoid the computation of constant A (present in the numerator and denominator):

$$I = \frac{1}{\sum_i h(\vec{p_i})}\sum_i h(\vec{p_i})l(\vec{p}) \tag{5.8}$$

5.1.2 Test Cases

We first validate this method using a set of simples cases emphasizing specific behaviours such as change of curvature, shininess, etc.

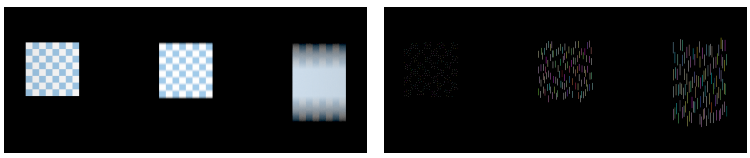
In a first set of examples (Figure 5.6 and Figure 5.7), we present how the filtering process behave in the presence of motion-blur and depth-of-field. The filters adapt to the local smoothness of the image thanks to the frequency estimation. Because of the combination of correlation and slicing, the resulting 2D filters from the 5D covariance matrix adapt to the effect. This could not be presented without an anisotropic analysis.



(a) Small lens **(b)** Large lens

(c) Small lens filters **(d)** Large lens filters

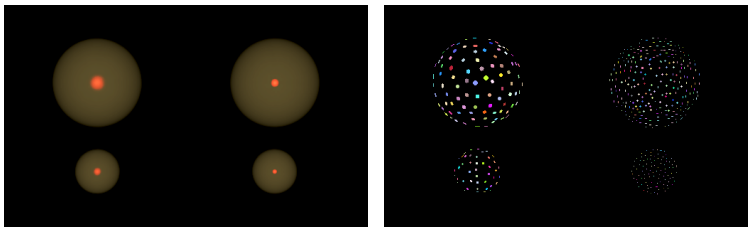
Figure 5.6 *– We compare the resulting image space filters for the same scene of diffuse checkerboard with different lens size. A large lens creates a bigger blurring of the out-of-focus region. We modify our code to enforce the texture to have the maximum image space frequency in order to emphasis on the lens effect.*



(a) Scene **(b)** Filters

Figure 5.7 *– We compare the resulting image space filters for a scene composed of diffuse checkerboard with motion along the y axis of the image. Large motion will produce more stretched filters.*

Contrary to other image space techniques [145, 129], we do not need to compare the samples between each other. Since the number of covariance paths is smaller than the number of radiance samples (by an order of magnitude), we avoid computational expenses during the reconstruction step. But at the expense of a prior-analysis on light-paths.



(a) Scene **(b)** Filters

Figure 5.8 *– We compare the effect of shininess on filters spread on spheres with material being the combination of a diffuse term and of a glossy lobe. For a nearly specular lobe (b), the filters are small and do not overblur the image. in the contrary, for more smooth BRDF, the filters spread more and diffuse samples according to the light, curvature and BRDF (a).*

5.1.3 Results

We compared our algorithm to a standard path-tracer. The snooker scene (Figure 5.9) presents a traditional snooker table under a sky environment map. The scene exhibits both low and high frequency materials (diffuse, glossy and specular). Frequency information is computed using 15 covariance samples per pixel. We limited the maximum number of primary rays per pixel to 100 for our algorithm. We used a 200 voxels wide voxel grid for the occlusion detection. We performed all computations on a Xeon W3520 at 2.66 GHz with 8GB of RAM. Our algorithm takes advantage of parallel computing, with OpenMP, for sampling both covariance rays and radiance rays.

(a) Equal time reference using 512 samples per pixel (25utes minutes) **(b)** Our algorithm (25 minutes) **(c)** Reference with same maximum number of samples per pixel (3000 sample per pixel, 2 hours 25 minutes)

Figure 5.9 *– The snooker scene rendered using our algorithm at the center, with a path tracer using the maximum number of samples in a pixel used in our algorithm at the right (same quality), and using the same amounts of time as our algorithm at the left.*

We further compare the snooker scene in Figure 5.10 using insets of the image. While the traditional ray-tracer still exhibits noise, our insets look converged (But we are incorporating some bias in our solution due to the

filtering approach). The green inset shows that for complicated cumulated effects such as combination of depth-of-field and highly varying shininess, our algorithm was not able to filter completely the noise out. This is because we use a threshold on the number of samples per pixel. The analysis tells us that more samples would be needed here. This is extensible with a progressive algorithm that would resample high frequency regions on demand.

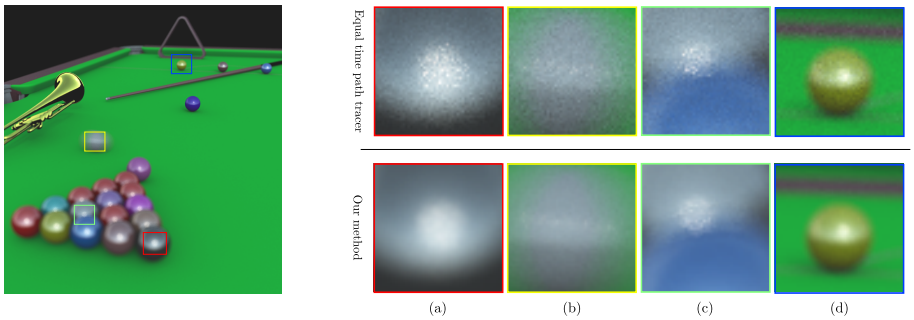


Figure 5.10 *– We compare insets of the rendering of the snooker scene from our method and from a path tracer using the same amount of time. We show here that using filters allow to get a noise-less image.*

The helicopter scene (Figure 5.11) shows a toy lit by a square light source. The rotor of the helicopter is rotating around its axis creating motion-blur, while the textured background of the scene exhibits depth-of-field effect. We used 10 light paths per pixel to estimate the covariance and a maximum of 200 samples for the reconstruction. Again, we compare our results with a path traced image computed in the same amount of time.

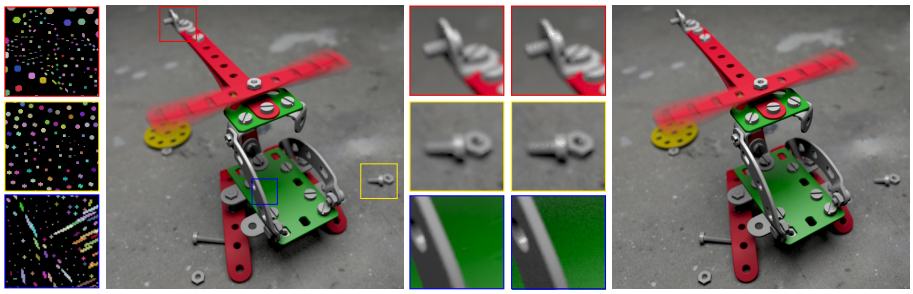


Figure 5.11 *– The helicopter scene rendered using our algorithm (a) and compared with an equal time path traced rendering (b). We used 10 light-paths per pixel to estimate the covariance and used a maximum of 200 converged samples for the reconstruction.*

These scenes demonstrate that our method save computation time for the low frequency parts of the scene. We report timings in Table 5.1.

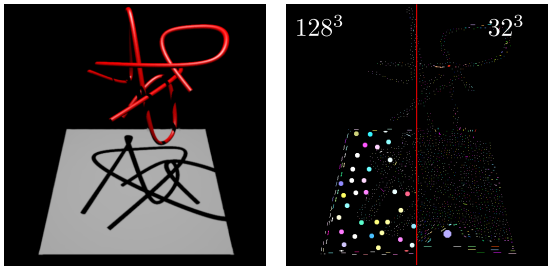
Scene	Our (covariance tracing / reconstruction)	Reference
Snooker	25m (2m36 / 16s)	2h25m
Helicopter	29m (2m / 16s)	x

Table 5.1 *– Timing comparison between our algorithm and our reference tracer for the snooker (Figure 5.9) and for the helicopter scene (Figure 5.11). The first column shows the time needed by our algorithm. Inside the brackets we show the covariance acquisition and the reconstruction timings. For the helicopter scene, we don't report the path tracer timing since we are doing equal time comparison.*

5.1.4 Limitations

Using non locally defined covariances of BRDF's spectrum over-estimates the materials frequency. This is visible in Figure 5.8 where only the reflection lobe should be at the maximum frequency. With more complex materials' covariances, we could correctly estimate the low frequency part of the snooker ball's BRDFs.

We are limited in this application by the use of a covariance grid. The resolution of the space compared to features size can lead to poor estimator of the visibility. In such a case, our algorithm will over-estimate the occlusion effect. This will influence the sampling rate and the reconstruction (Figure 5.12).



(a) Scene (b) Filters

Figure 5.12 *– We compare different resolutions of the voxel grid for a complex occluder. Note how the coarse resolution grid leads to small reconstruction filters in most of the image.*

Furthermore, the voxel grid adds cost to our covariance tracing step since we have to perform ray marching into this structure. We developed this structure to prove that it is possible to add the covariance analysis into a standard path tracer without altering the inner code.

5.2 Object Space Application

In this section, we use covariance information to estimate kernels in photon mapping (and progressive photon mapping) applications. We aim to reduce noise in low variation regions. Using larger kernels in such regions improves the quality of the estimate as more photons are gathered by the kernel while preserving a low bias.

5.2.1 Estimation of the Kernel size for Photon Mapping

Photon mapping [91] and progressive photon mapping [65, 98] (PPM) are applications of kernel based density estimation [163]. Our idea is to take advantage of information brought by the covariance matrix to define the kernel size. Kernels should not be smaller than the frequency content of the density they are reconstructing. We detail a method to estimate kernel radii based on covariance matrices.

Other methods have looked at giving better estimates of high frequency part of the photon map [151, 165, 169]. Those methods are complementary to our work, we filter the photon map according to its local frequency. Hierarchical methods [166] are close to our work, but rely on the eye-ray footprint. Our method can estimate kernel larger than the eye footprint. Diffusion methods [150] perform photon diffusion respecting edges. They rely on the photon statistic to define the diffusion gradients whereas our frequency estimate is a feature of photons.

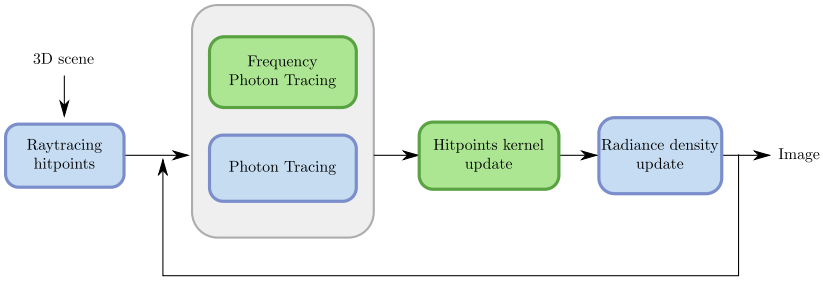


Figure 5.13 *– We modify the classical progressive photon mapping pipeline (in blue) and add a particular photon mapping step where photons carry covariance information (in green). The kernels are updated using this frequency information.*

We propose to look at an implementation of the Progressive Photon Mapping algorithm [65] (Figure 5.13 presents our pipeline). The derived radius can be used in a classical Photon Mapping algorithm.

5.2.2 Error based kernel size estimation

In this section we propose to use the approximation of kernel based density estimation's error described in Silverman's monograph [163, Chapter 4.3.1]:

$$\epsilon(x) \simeq \frac{1}{2} r^2 \alpha \nabla^2(f)(x) \tag{5.9}$$

We use the Property 3 of the covariance matrix to estimate the Laplacian of a density function. We want an error below a given threshold error ϵ_0. Given the knowledge of the local Laplacian, we can infer an approximate optimal radius of collection r_0:

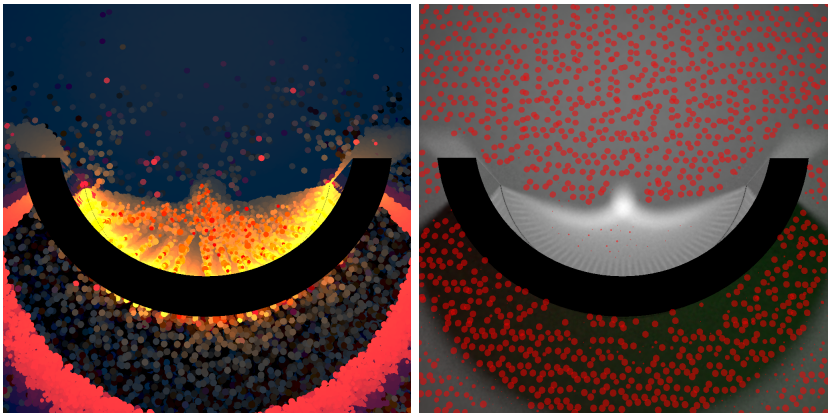
$$r_0 \simeq \sqrt{\frac{2\epsilon_0}{\alpha \nabla^2(f)(x)}} \tag{5.10}$$

Given the equation of the radius (Equation 5.11), and the fact that we can estimate the Laplacian of f at p from its covariance matrix (Property 3),

we derive the following equation, giving the radius based on the covariance matrix and an error coefficient ϵ_0 allowing for user control over the bias of the reconstruction.

$$r_0 \simeq \sqrt{\frac{2\epsilon_0}{\alpha\Big(\Sigma_{x,x}(f)(x) + \Sigma_{y,y}(f)(x)\Big)}} \tag{5.11}$$

For this application, we used 2D covariance matrices assuming that the signal is isotropic in space and isotropic in angle, with no motion. This permits to keep covariance computation times short enough to compare the algorithms at equivalent passes.



(a) Covariance accumulated in the hit points **(b)** Estimated filters using the heuristic

Figure 5.14 *– Covariance can be visualised as hit points map to pixels. We show here the accumulated covariance at the surface of objects (a). The scene is composed of a curved mirror on a diffuse surface. The light creates a caustic and a soft shadow. Aside (b), we display estimated filters drawn on top of the final image. The filters are small is the high frequency regions (caustic and shadow).*

5.2.3 Results

As shown in the Figure 5.15 our algorithm converges faster for low frequency parts such as diffuse non-occluded regions or indirectly lit diffuse regions.

The frequency analysis permits to detect specific lighting situations, such as a caustic focus point, which corresponds to a frequency spectrum with no energy along the angular axis. As expected, the high frequency content in the spatial domain at the focus point causes smaller reconstruction kernels (Figure 5.16). In the caustic scene, the lower part of the caustic has high frequency due to the facetting of the model.

For the equal amount of time, our algorithm exhibits less noise in diffuse low frequency regions (Figure 5.17).

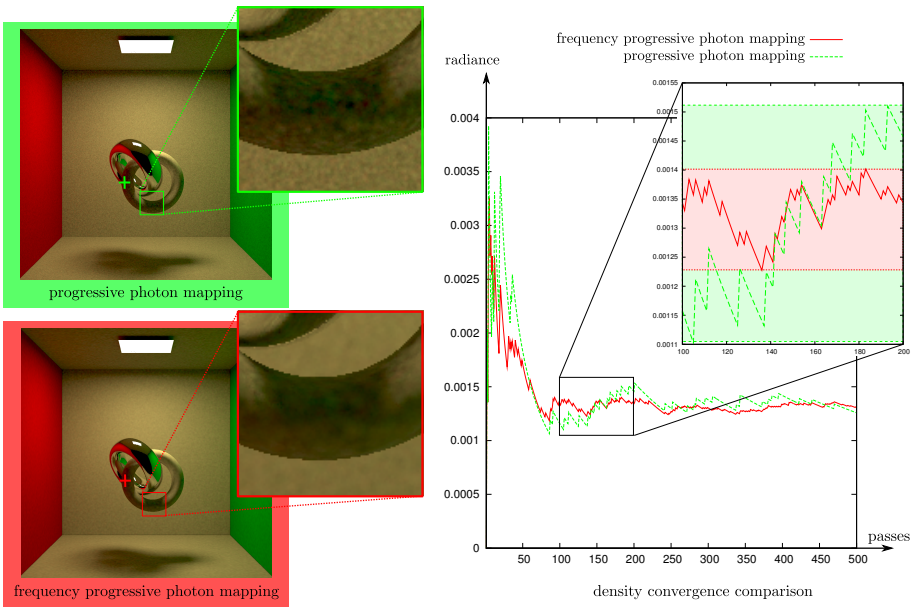


Figure 5.15 *– We compare against progressive photon mapping with our algorithm for the convergence of an indirectly lit part of the scene. In the closeup, we show that our algorithm produces a lower varying estimate at an earlier stage of its execution.The images where produced using* 100.000 *photons per pass and* 25% *of frequency photons to make timing comparable.*

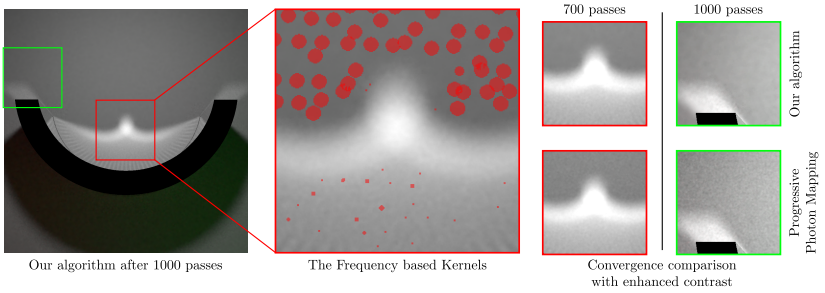
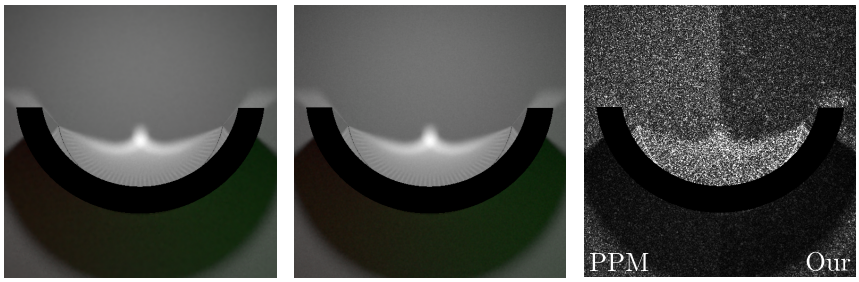


Figure 5.16 *– We compare the convergence of two regions of the caustic scene. Those regions contain both high frequency and low frequency content. The low frequency regions exhibit less noise than the high frequency ones.*

5.3 Participating media

In this section, we present results for the application of our covariance analysis for the integration of volumetric effects in the case of participating media (e.g., fog, clouds). First, we will present a data structure that reduces the cost of covariance tracing by reusing covariance information from previous traced light-paths: the *covariance grid* (Section 5.3.1). Then, we present three use of the covariance grid: adaptive sampling and filtering in image space (Section 5.3.2), adaptive sampling along a ray for one scatter illumination (Section 5.3.3), and an improvement of the progressive photon beams algorithm [89] (Section 5.3.4).



(a) Our **(b)** PPM **(c)** L2 Comparison

Figure 5.17 *– We compare the convergence for the caustic scene using the L2 norm for two equal time rendering. While the caustic region exhibits the same amount of noise, the diffuse parts are converging faster.*

5.3.1 Covariance grid

The covariance grid is a voxel grid that stores covariance information of the irradiance inside a participating medium. We do not store covariance information of radiance since this latter quantity is a directional information and would require a quantization of directions. Such quantization would dramatically increase the memory footprint of the covariance grid, making it less usable for resolution questions for example. Instead, we propose to store 3D covariance of irradiance.

5.3.1.1 Definition

In each voxel $\mathbf{p}$ of the grid, we store a 3×3 covariance matrix $\Sigma_{\mathbf{p}}$, where entry (i, j) is the ij-covariance of the Fourier transform of the local irradiance in the neighborhood of $\mathbf{p}$:

$$(\Sigma_{\mathbf{p}})_{ij} = \int_{\omega} \omega_i \omega_j \mathcal{F}\big[I\big](\omega) d\omega \tag{5.12}$$

The irradiance I is defined as:

$$I(\mathbf{p}) = \int_{\vec{d} \in S^2} l(\mathbf{x}, \vec{d})$$

5.3.1.2 From Covariance of Radiance to Covariance of Irradiance

For a light-path and a position on this light-path, we can compute the covariance of the local radiance. We show here how from a covariance matrix of radiance we can obtain the spatial matrix of irradiance.

Let Σ be the covariance matrix at position $\mathbf{p}$ of a light-path l. We can compute the 2D covariance of irradiance on the orthogonal plane to the central ray by looking at the integration over angles of the radiance, which is in the Fourier domain a slice of the spectrum.

$$\Sigma_{\vec{x}_{loc}, \vec{y}_{loc}} = \Sigma|_{1,2} \tag{5.13}$$

We assume the local irradiance to be constant along $\vec{z}_{loc}$, as infinitesimal analysis allows to do (the shear due to travel will be second order in this case).

The final 3D local covariance matrix of the irradiance's spectrum is then:

$$\Sigma_{loc} = \begin{bmatrix} \Sigma|_{1,2} & 0 \\ 0^T & 0 \end{bmatrix} \tag{5.14}$$

Note that this covariance matrix is defined in a local frame $(\vec{x}_{loc}, \vec{y}_{loc}, \vec{z}_{loc})$. We rotate it to express the local covariance in the global frame $(\vec{x}, \vec{y}, \vec{z})$.

5.3.1.3 Accumulating Local Irradiance's Covariance Matrices

The last step is to accumulate different covariance matrices of the local irradiance to get an estimate of the global covariance matrix of irradiance. This is possible as the covariance is a property of an incoming light-path a thus can be integrated over the light-path space.

$$\Sigma_{\mathbf{p}} = \sum_{\vec{d} \in S^2} w_{\vec{d}} \Sigma_{\mathbf{p}}(\vec{d}) \tag{5.15}$$

Where $w_{\vec{d}}$ is the light intensity weight.

5.3.1.4 An example: A caustic from a glass sphere

The covariance grid stores the spatial covariance of irradiance. As such, directional high frequency regions are not represented in the orthogonal plane, but in the global plane. As an example, Figure 5.18 shows the equivalent Gaussian of the spatial covariance for a selection of point in a caustic created by a glass sphere lit by a spot light.

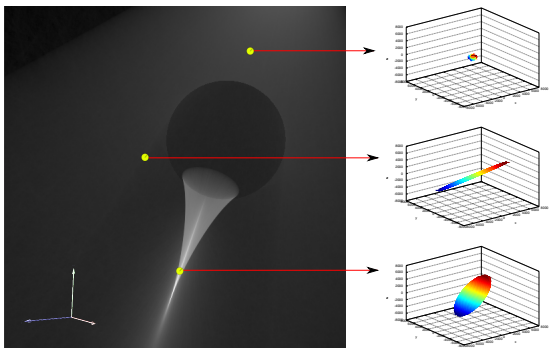


Figure 5.18 *– A caustic is created using a sphere and a spot light. We analyze three positions in the covariance grid and show the equivalent Gaussians on the right. The upper figure is a diffuse part of the scene, we covariance is the minimum covariance due to the grid analysis. The middle inset shows the spatial covariance of a shadow. The elongated peak is in the direction of the visibility. Last is a point inside the caustic. The caustic is a high frequency region where different directions will accumulate high frequency. As such, the equivalent Gaussian expands in all three directions.*

5.3.2 Image Space Adaptive Sampling and Reconstruction using the Covariance Grid

In this application, we accumulate covariance matrices on the image plane by ray marching the covariance grid. The 3D covariance matrices are sliced to extract a 2D covariance matrix in the local frame of the image. We add the eye path attenuation and occlusion spatial covariance matrix before accumulating on the screen.

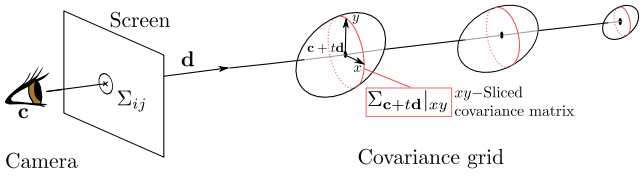
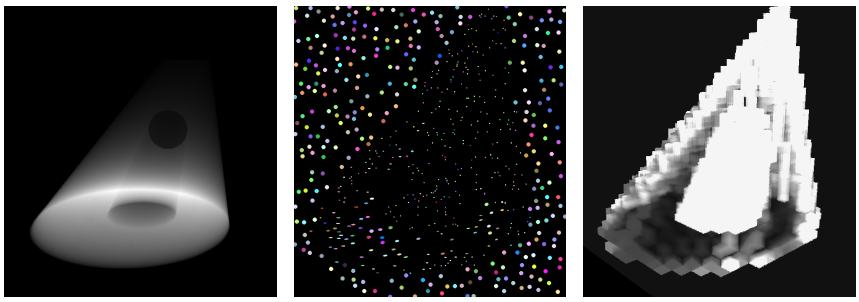


Figure 5.19 *– We accumulate spatial covariance matrices from the covariance grid by ray marching. In each voxel, we slice the covariance of the irradiance's spectrum and add the eye path attenuation and occlusion to it. The resulting 2D matrices are averaged using intensity weights.*

We illustrate the accumulation using Figure 5.19. Given a camera shooting eye rays with starting position $\mathbf{p}$ and direction $\mathbf{d}$, we construct the accumulated covariance matrix $\Sigma_{i,j}$ with a weighted average of slices of spatial covariance matrices along the ray $\Sigma_{\mathbf{c}+t\mathbf{d}|x,y}$, $t \in [0, d_{hit}]$ to which we add the covariance of the attenuation to the eye $\Sigma_{A(t)}$.



(a) Reconstruction **(b)** Estimated filters **(c)** Estimated density

Figure 5.20 *– A shaft created by a sphere lit by a spot light in an inhomogeneous volume with an exponentially varying density along the height. The covariance grid's size is* 32^3.

The Figure 5.20 shows how the effect of a shaft is handled by our adaptive method. Our method adapts the samples to favor crepuscular regions (such as the shaft of the sphere). The border of the spot light creates a high frequency regions.

5.3.3 Adaptive sampling along a ray

To integrate the effect of the participating media into a ray tracer, we need to add another integral dimension per ray bounce (counting eye rays as a

ray "bounce"). This integral accounts for the scattering of of light from the scene sources along the ray. It is usually done by ray marching along the ray, connecting the sampled position on the ray to light sources and adding this contribution to the integral with correct attenuation factors.

We use frequency information to drive the sampling of positions along the ray. We illustrate our method by the Figure 5.22. This method is similar to the method proposed by Engelhardt and Dachsbacher [52] where god rays are adaptively sampled with respect to the integrand discontinuity. Instead, we look at the variance along the ray. This way we capture the discontinuity due to occlusion (as it generates high variance spectra), we capture the variation due to changing density and other effects such as convergence of photons in the case of a caustic.

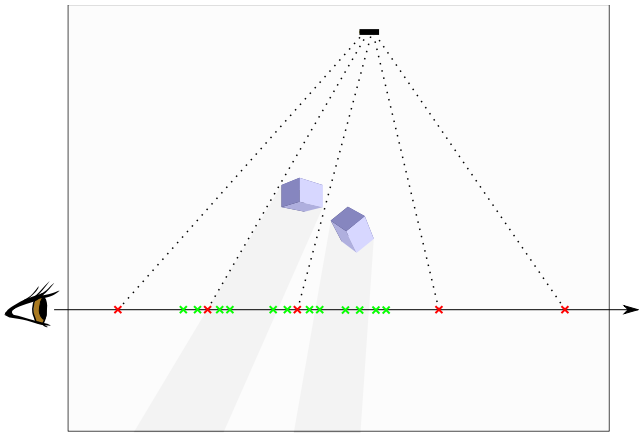


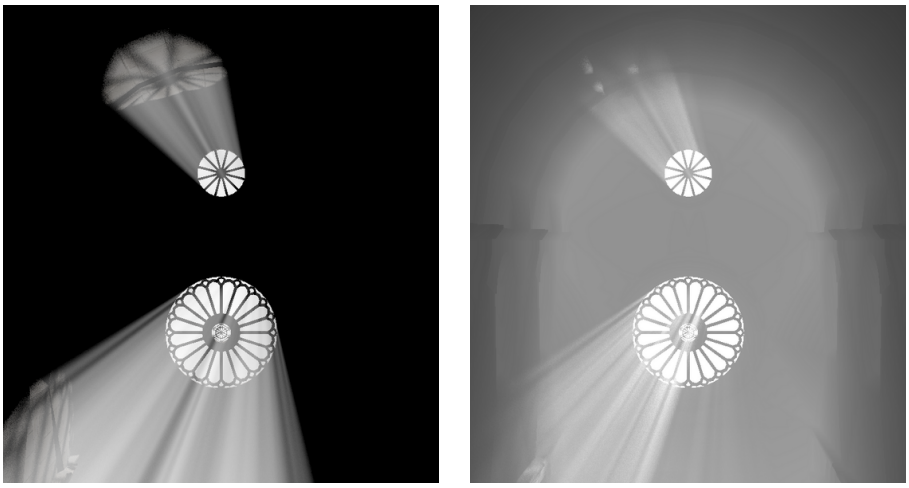
Figure 5.21 *– We perform adaptive sampling along the eye-ray by resampling regions of high variance. In a first pass, we estimate the covariance matrix and scattered radiance for a sparse set of samples. Then, from the covariance matrix, we estimate the variance of the incoming radiance along the eye-ray to resample regions with high variation.*

Since our algorithm takes advantage of adaptive sampling on the shadows boundaries, we are able to reduce the aliasing of light shafts caused by undersampling the high frequency regions. Figure 5.22 shows our results on the Sibenik cathedral model where the rose windows are used to cast highly varying shafts due to the fine geometry of the rose.

Our adaptive sampling strategy is based on the variance of the illumination. Traditional algorithms [189] are based on the maximum variance of the density in the medium along a light path. Therefore we avoid oversampling regions with too-low energy.

5.3.4 Frequency Progressive Photon Beams

We build upon the existing work of Jarosz et al.'s *progressive photon beams* [89] (referred as *PPB*) to illustrate the benefits of the frequency analysis. In the progressive photon beam algorithm, photons are traced in the scene containing a participating medium and the paths of propagation (called beams) are stored. Then, rays are shot from the camera for each pixel, and the density of beams



Adaptive sampling along eye path | Total number of samples per pixel

Figure 5.22 *– We present the result of integrating the shaft casted by the rose windows in the Sibenik cathedral (modeled by Marko Dabrovic). We also show the total number of samples used per pixel for our algorithm.*

along the ray is estimated using a 1D kernel (Figure 5.23). This is repeated while decreasing kernel size until convergence is satisfactory.

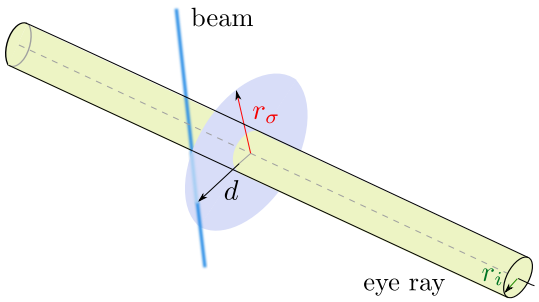


Figure 5.23 *– Given a camera ray (in green), and a beam, we use the radius, r_σ, estimated by the covariance analysis, instead of the radius, r_i of the progressive photon mapping when r_i is smaller. The effect is to gather more beams in low frequency regions to decrease the variance of the estimate in those regions.*

5.3.4.1 Gathering photon beams

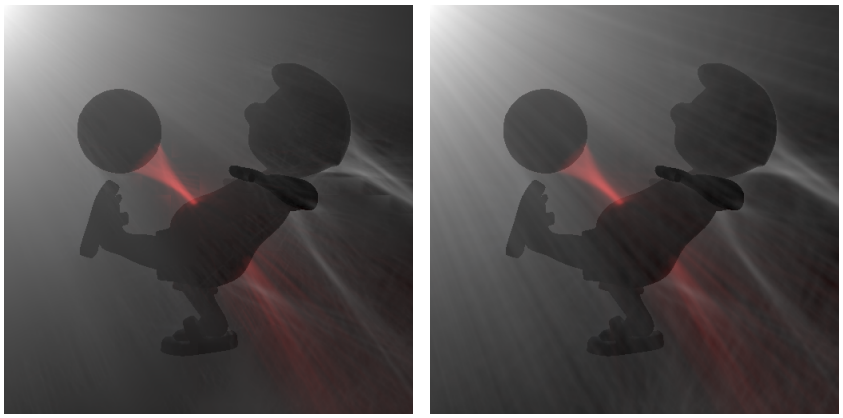
During the gathering pass, for each eye ray, we test its distance d to the beams stored (See Figure 5.23). At the closest point to each beam along the ray, we look into the covariance matrix, and estimate the ideal gathering radius r_σ using the error formula (Equation 5.11) but in a 1D setting. We gather that

beam only if:

$$d < \max(r_i, r_\sigma)$$

Where, r_i is the radius given by the photon beam method for pass #i. In other words, we replace the gathering radius of progressive photon mapping by a specific radius for each pair (eye-ray, photon beam) adapted to the local variations of the signal. This adaptive radius computation prevents us from decreasing the radius in regions of low bandwidth, and therefore reduces variance, while controlling the bias. We implemented this process in CUDA, which allows us to compare our results to the implementation of PPB by Jarosz et al. [89].

We validate our covariance computation step using classical test scenes such as a caustic produced by a glass sphere (Figure 5.25) and the soccer boy (Figure 5.24) to illustrate particular high frequency setups such as occlusion or light concentration. In both cases, our covariance framework correctly estimates the high frequency regions. Note that we do not follow specular eye path to gather beam, this explains the inner look of the soccer boy.



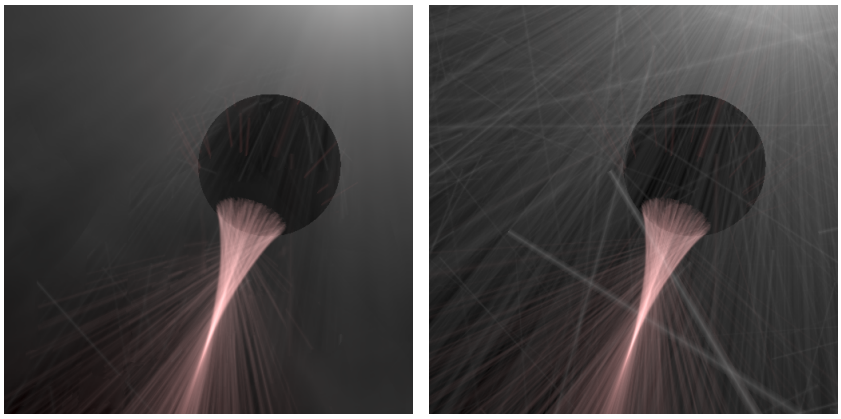
(a) Our method, 740s **(b)** Equal time PPB

Figure 5.24 *– Complicated caustics are produced by a glass soccer boy figurine (From Sun et al. [168]). Progressive photon beam is very effective to produce converged volumetric caustics but is rather slow at generating smooth diffusion. Our covariance grid algorithm is able to diffuse more beams in low frequency parts of the screen, allowing to meet the stop criterion sooner there.*

At equal computation time, we achieve a much better convergence in smoother regions of the image, while we keep the equal convergence in high frequency regions such as the caustic.

5.3.4.2 Discussion

We do not need to precompute the covariance grid. As for the frequency progressive photon mapping algorithm, we update it while tracing photon beams. For each photon tracing pass, we allocate a fixed proportion of photons to carry a covariance matrix (For our scenes, we chose 10% percent as the proportion).



(a) Our method, 342 **(b)** Equal time PPB

Figure 5.25 *– A glass sphere is lit by a diffuse point light. This setup creates a caustic in the axis of the light and sphere centers.*

The matrix is updated as the photon is reflected, refracted, and scattered and the grid is updated using ray marching.

The main strength of the covariance estimation of kernel radii is to stop the radius reduction when it passes below a frequency criterion. This allows the estimate to converge faster in low-frequency regions. For high frequency region, our estimation do not blur the beams resulting in the same convergence than the classical progressive photon mapping algorithm.

Keeping a large radius slows down the selection process, when an acceleration structure such as a KD-tree [168] used. As the areas are bigger, it increases the branching factor of the KD-tree search. In our CUDA implementation which follows the method described by Jarosz et al., there is no penalty as there is no acceleration structure.

6 | Conclusion

This dissertation explored a way to optimize the generation of synthetic photo-realistic images. From a set of various light phenomenons (e.g., depth-of-field, motion blur, scattering in fog), we derived the Fourier transform of the mathematical formulation using the framework proposed by Durand et al. [47]. We showed how to depict blurry regions and adapt the calculation for various light transport algorithms in order to obtain faster convergences. In this chapter, we review the presented contributions and propose future directions of research.

6.1 Summary of Contributions

We presented the most complete, to date, analysis of light transport from the perspective of local light-fields spectrum. This analysis expresses the variations changes in intensity of local neighboring rays with respect to a given light-path. We expressed this analysis in a united form, using a composition formulation that chains the operators to define the final operator on a light-path. We generalized previously defined operators:

- *Occlusion* for non planar objects. We showed that the wedge spectrum [107] estimated from multiples slices of the occluder can be explained by a conservative non planar occluder.
- *Reflection,* more precisely the integration with the BRDF. It is expressed as a convolution in a six dimensional space followed by a reduction of the number of dimensions. We showed that for one dimensional BRDF the resulting operator is a multiplication in a four dimensional space as previously stated [47, 140].
- *Lens* for thin lenses. We presented an operator that allows to express a chain of lenses. This operator defines the transport of light throught a small lens using shears. This is simpler than the previous method which used a Bessel multiplication [164].
- *Motion.* We provided a generalization of the motion operator (previously defined for special cases [49]). We modelized motion as a change of referential before local operations (i.e., reflection, refraction, ...).

We enriched the analysis with two new operators to handle the case of non-thick participating media (e.g., smoke, air, water, ...) and rough refractive surfaces:

- *Attenuation.* It models the absorption and out-scattering of light when photons travel inside a non empty medium. We showed that this operator is related to the occlusion operator. Furthermore, the non opque visiblity can be approximated using the density gradient.

- *Scattering.* It models the in-scattering of light from a known direction. We showed that, for the case of an isotropic phase function, the operator is a convolution in angle. If the main scattering angle is not close to zero, the spatial frequency along the X axis vanishes.

- *Rough refraction.* It models the transmission of light inside a constant index medium. We showed how to define Walter's refraction model [184] in a local light-field setting and proposed to approximate the BTDF's spectrum amplitude with Gaussian lobes.

We provided a new versatile tool to express the spectrum and estimate the anisotropic variance of the spectrum in a ray-tracing application: the *covariance matrix*. We build a Monte-Carlo estimate of this tool for complex lighting effects. We proposed to use voxel grids to estimate the partial occlusion of the local light-field. Those structures enable to perform frequency analysis for global illumination effects.

We introduced the *covariance grid*, a tool to distribute a prior covariance analysis for participating media. The covariance grid stores the covariance matrix of the irradiance's spectrum. We showed the benefits of using it in various application ranging from image space adaptive sampling and reconstruction to filtering photon beam kernels.

We provided examples of use of the covariance analysis. Several improvements on classical algorithm where demonstrated:

- *Adaptive sampling and reconstruction.* Covariance information permits to do sparse adaptive sampling in the image space and to filter distributed effects such as soft shadows, motion blur and depth-of-field.

- *Kernel filtering.* Photon mapping and its derivatives can benefit from the covariance analysis as an optimal radius of gather can be derived.

- *Adaptive sampling in light-path space.* Non uniform sampling along eye paths to integrate shafts can be done using the covariance grid. This can be extended to other highly varying light effects in volume such as caustics.

6.2 Perspectives

Towards completeness We did not completely succed in our goal to provide a complete frequency analysis of light transport. Some lighting effects present in current ray-tracers are missing:

- *Phase function spectrums.* We describe the spectrum of two phase functions: the *Phong* BRDF and the *Heynyey-Greenstein* phase function.

A large body of phase functions exits. To be complete, the frequency analysis has to provide a spectrum for each one of them.

- *BSSRDFs.* Diffusion of light on surfaces is not part of the operators. Yet, BSSRDFs are well studied in computer graphics. It is known that scattering in such thick media produces a spatial and angular blur of the signal [93, 41].

- *Wave optics effects* such as diffraction are not embodied inside our analysis. They are not part of the geometrical optics model. Yet, recent works inside the graphics community approximate wave optics phenomenons [148, 35].

Our representation of the spectrum misses phase information. Phase of the Fourier transform is rich in terms of information (Figure 3.4, and Figure 3.27). While the lack of phase in our covariance analysis leads to an over estimate of the spectrum, our experiments showed that we can provide a gain.

Applications of Covariance We did not cover all the possible applications of the covariance matrix. Its genericity makes it a versatile tool to work with light-paths. Other possible applications are:

- *Irradiance caching* is a promising area. The covariance matrix can bring an estimate of the Hessian matrix. The second derivatives are used to predict the density of radiance (or irradiance) caches [90].

- *Information theory* has application in computer graphics [159, 149]. The covariance matrix give a local bound to the entropy of the signal which has to be infered from the samples in previous works.

- *Driving light-paths generation* for photon mapping to draw photons based on the frequency content of the integrand. In combinaison to our reconstruction kernels, this would result in a better estimate of the high frequency regions while preserving a good convergence of the low frequency regions.

A limitation of covariance is its mono-modal form. A spectrum consisting of differents separated modes (such as a diffuse term and a separated specular term) will be treated as a whole. Multi-modality analysis can be beneficial but remain a hard task. It requires to keep track of a matrix per mode. This can become untracktable as the number of bounces augment. Further research in this direction would require to analyze the different type of multi-modal signals present in rendering and to build a tool that account for the different modes.

Bibliography

[1] Sameer Agarwal, Ravi Ramamoorthi, Serge Belongie, and Henrik Wann Jensen. Structured importance sampling of environment maps. *ACM Transactions on Graphics*, 22(3):605–613, July 2003. doi: 10.1145/882262.882314.

[2] Javier Alda. Laser and Gaussian Beam Propagation and Transformation. *Optical Engineering*, 110(10 Pt 1):999–1013, 2003. doi: 10.1081/E-EOE.

[3] John Amanatides. Ray tracing with cones. *ACM Siggraph Computer Graphics*, 18(3):129–135, January 1984. doi: 10.1145/964965.808589.

[4] John Amanatides and Andrew Woo. A Fast Voxel Traversal Algorithm for Ray Tracing. Technical report, University of Toronto, 1987.

[5] James Arvo. Backward Ray Tracing. *ACM SIGGRAPH 86 Course Notes Developments in Ray Tracing*, 12:259–263, 1986. doi: 10.1145/10.1.1.31.581.

[6] James Arvo. The irradiance Jacobian for partially occluded polyhedral sources. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques - SIGGRAPH '94*, pages 343–350. ACM Press, July 1994. doi: 10.1145/192161.192250.

[7] Michael Ashikhmin and Peter Shirley. An Anisotropic Phong BRDF Model. *Journal of Graphics Tools*, 5(1):25–32, 2000. doi: 10.1080/10867651.2000.10487522.

[8] Mahdi Bagher, Cyril Soler, and Nicolas Holzschuch. Accurate fitting of measured reflectances using a Shifted Gamma micro-facet distribution. *Computer Graphics Forum*, 31(4), June 2012.

[9] Mahdi M. Bagher, Cyril Soler, Kartic Subr, Laurent Belcour, and Nicolas Holzschuch. Interactive rendering of acquired materials on dynamic geometry using bandwidth prediction. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '12, pages 127–134. ACM, 2012. doi: 10.1145/2159616.2159637.

[10] Kavita Bala, Bruce Walter, and Donald P Greenberg. Combining edges and points for interactive high-quality rendering. *ACM Transactions on Graphics*, 22(3):631–640, 2003. doi: 10.1145/882262.882318.

[11] Laurent Belcour and Cyril Soler. Frequency based kernel estimation for progressive photon mapping. In *SIGGRAPH Asia 2011 Posters*, SA '11, pages 47:1–47:1. ACM, 2011. ISBN 978-1-4503-1137-3. doi: 10.1145/2073304.2073357.

[12] Laurent Belcour, Cyril Soler, and Kavita Bala. Frequency analysis of participating media. Technical report, INRIA and Cornell University, 2012.

[13] Laurent Belcour, Cyril Soler, Kartic Subr, Frédo Durand, and Nicolas Holzschuch. 5D Covariance Tracing for Efficient Depth of Field and Motion Blur. *ACM Transactions of Graphics*, 201X. Accepted with minor modifications.

[14] Mark R. Bolin and Gary W. Meyer. A frequency based ray tracer. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 409–418. ACM, 1995. doi: 10.1145/218380.218497.

[15] Mark R. Bolin and Gary W. Meyer. A perceptually based adaptive sampling algorithm. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 299–309. ACM, 1998. doi: 10.1145/280814.280924.

[16] Antoine Boudet, Mathias Paulin, Paul Pitot, and David Pratmarty. Low Memory Spectral Photon Mapping. Technical report, Université Paul Sabatier, Toulouse, 2004.

[17] Antoine Boudet, Paul Pitot, David Pratmarty, and Mathias Paulin. Photon Splatting for participating media. In *Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, GRAPHITE '05, page 197. ACM Press, November 2005. doi: 10.1145/1101389.1101430.

[18] Eric Bruneton and Fabrice Neyret. A Survey of Non-linear Pre-filtering Methods for Efficient and Accurate Surface Shading. *IEEE Transactions on Visualization and Computer Graphics*, 18(2):242–260, February 2012. doi: 10.1109/TVCG.2011.81.

[19] A. Buades, B. Coll, and J. M. Morel. A Review of Image Denoising Algorithms, with a New One. *Multiscale Modeling & Simulation*, 4(2):490, November 2005. URL `http://link.aip.org/link/?MMSUBT/4/490/1`.

[20] David Burke, Abhijeet Ghosh, and Wolfgang Heidrich. Bidirectional Importance Sampling for Direct Illumination. In *Rendering Techniques*, pages 147–156, 2005. URL `http://dblp.uni-trier.de/db/conf/rt/rt2005.html#BurkeGH05`.

[21] Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. Plenoptic sampling. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 307–318. ACM, 2000. doi: 10.1145/344779.344932.

[22] Jiating Chen, Bin Wang, Yuxiang Wang, Ryan S Overbeck, Jun-Hai Yong, and Wenping Wang. Efficient Depth-of-Field Rendering with Adaptive Sampling and Multiscale Reconstruction. *Computer Graphics Forum*, 30(6):1667–1680, 2011. doi: 10.1111/j.1467-8659.2011.01854.x.

[23] Jiating Chen, Bin Wang, and Jun-Hai Yong. Improved Stochastic Progressive Photon Mapping with Metropolis Sampling. *Computer Graphics Forum*, 30(4):1205–1213, 2011. doi: 10.1111/j.1467-8659.2011.01979.x.

[24] Min Chen and James Arvo. Theory and application of specular path perturbation. *ACM Transactions on Graphics*, 19(4):246–278, October 2000. doi: 10.1145/380666.380670.

[25] Shencahng Eric Chen, Holly E. Rushmeier, Gavin Miller, and Douglas Turner. A progressive multi-pass method for global illumination. *ACM SIGGRAPH Computer Graphics*, 25(4):165–174, July 1991. doi: 10.1145/127719.122737.

[26] Ying-Chieh Chen, Su Ian Eugene Lei, and Chun-Fa Chang. Spatio-Temporal Filtering of Indirect Lighting for Interactive Global Illumination. *Computer Graphics Forum*, 31(1):189–201, February 2012. doi: 10.1145/2283296.2283300.

[27] Petrik Clarberg, Wojciech Jarosz, Tomas Akenine-Möller, and Henrik Wann Jensen. Wavelet importance sampling. *ACM Transactions on Graphics*, 24(3):1166, July 2005. doi: 10.11451186822.1073328.

[28] Luc Claustres, Mathias Paulin, and Yannick Boucher. BRDF Measurement Modelling using Wavelets for Efficient Path Tracing. *Computer Graphics Forum*, 22(4):1–16, 2003.

[29] Luc Claustres, Mathias Paulin, and Yannick Boucher. A wavelet-based framework for acquired radiometric quantity representation and accurate physical rendering. *The Visual Computer*, 22(4):221–237, April 2006. doi: 10.1007/s00371-006-0001-y.

[30] David Cline and PK Egbert. Two stage importance sampling for direct lighting. In Tomas Akenine-Möller and Wolfgang Heidrich, editors, *Rendering Techniques 2006 (Eurographics Symposium on Rendering)*, pages 103–113, Nicosia, Cyprus, 2006. Eurographics Association.

[31] David Cline, Justin Talbot, and Parris Egbert. Energy redistribution path tracing. *ACM Transactions on Graphics*, 24(3), 2005. doi: 10.1145/1186822.1073330.

[32] C. C. Constantinou and C. Demetrescu. Physical interpretation of virtual rays. *Journal of the Optical Society of America A*, 14(8):1799, August 1997. ISSN 1084-7529. URL `http://josaa.osa.org/abstract.cfm?URI=josaa-14-8-1799`.

[33] Robert L. Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics*, 6(1), January 1986.

[34] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*, chapter 9, pages 224–238. John Wiley & Sons, Inc, 1991. ISBN 0-471-06259-6.

[35] Tom Cuypers, Se Baek Oh, Tom Haber, Philippe Bekaert, and Ramesh Raskar. Ray-Based Reflectance Model for Diffraction. *ACM Transactions on Graphics*, 2012. pre-print.

[36] Holger Dammertz, Daniel Sewtz, Johannes Hanika, and Hendrik P. A. Lensch. Edge-avoiding À-Trous wavelet transform for fast global illumination filtering. In M. Doggett, S. Laine, and W. Hunt, editors, *HPG '10 Proceedings of the Conference on High Performance Graphics*, pages 67–75, Saarbrücken, Germany, June 2010. Eurographics Association. doi: 10.1145/1921479.1921491.

[37] Gaston Darboux. *Leçons sur la théorie générale des surfaces et les applications géométriques du calcul infinitésimal*. Gauthier-Villars, Paris, 1887. URL `http://gallica.bnf.fr/ark:/12148/bpt6k77831k`.

[38] Tomáš Davidovič, Jaroslav Křivánek, Miloš Hašan, Philipp Slusallek, and Kavita Bala. Combining global and local virtual lights for detailed glossy illumination. *ACM Transactions on Graphics*, 29(6):1, December 2010. doi: 10.1145/1882261.1866169.

[39] Charles de Rousiers, Adrien Bousseau, Kartic Subr, Nicolas Holzschuch, and Ravi Ramamoorthi. Real-time rough refraction. In *Symposium on Interactive 3D Graphics and Games on - I3D '11*, page 111. ACM Press, February 2011. doi: 10.1145/1944745.1944764.

[40] Paul Debevec. A median cut algorithm for light probe sampling. In *ACM SIGGRAPH 2005 Posters on - SIGGRAPH '05*, page 66. ACM Press, July 2005. doi: 10.1145/1186954.1187029.

[41] Eugene D'Eon and Geoffrey Irving. A quantized-diffusion model for rendering translucent materials. *ACM Transactions on Graphics*, 30(4):1, July 2011. doi: 10.1145/2010324.1964951.

[42] G. A. Deschamps and P. E. Mast. Beam Tracing and Applications. *Quasi-Optics; Proceedings of the Symposium on Quasi-Optics*, 14, 1964. URL `http://adsabs.harvard.edu/abs/1964quop.conf..379D`.

[43] G.A. Deschamps. Gaussian beam as a bundle of complex rays. *Electronics Letters*, 7(23):684, 1971. doi: 10.1049/el:19710467.

[44] Mark A. Z. Dippé and Erling Henry Wold. Antialiasing through stochastic sampling. *ACM SIGGRAPH Computer Graphics*, 19(3):69–78, 1985. doi: 10.1145/325165.325182.

[45] Michael Donikian, Bruce Walter, Sebastian Fernandez, Kavita Bala, and Donald P Greenberg. Accurate Direct Illumination Using Iterative Adaptive Sampling. *IEEE Transactions on Visualization and Computer Graphics*, 12(3):353–364, 2006. doi: 10.1109/TVCG.2006.41.

[46] Frédo Durand. A frequency analysis of monte-carlo and other numerical integration schemes. Technical report, MIT, 2011. URL `http://dspace.mit.edu/handle/1721.1/67677`.

[47] Frédo Durand, Nicolas Holzschuch, Cyril Soler, Eric Chan, and François X. Sillion. A frequency analysis of light transport. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 1115–1126. ACM, 2005. doi: 10.1145/1186822.1073320.

[48] Philip Dutré, Philippe Bekaert, and Kavita Bala. *Advanced Global Illumination*. A.K. Peters / CRC Press, 2003. ISBN 9781568811772.

[49] Kevin Egan, Yu-Ting Tseng, Nicolas Holzschuch, Frédo Durand, and Ravi Ramamoorthi. Frequency analysis and sheared reconstruction for rendering motion blur. *ACM Transactions on Graphics*, 28(3), 2009. doi: 10.1145/1576246.1531399.

[50] Kevin Egan, Frédo Durand, and Ravi Ramamoorthi. Practical filtering for efficient ray-traced directional occlusion. *ACM Transactions on Graphics*, 30, December 2011. doi: 10.1145/2024156.2024214.

[51] Kevin Egan, Florian Hecht, Frédo Durand, and Ravi Ramamoorthi. Frequency analysis and sheared filtering for shadow light fields of complex occluders. *ACM Transactions on Graphics*, 30(2):9:1–9:13, April 2011. doi: 10.1145/1944846.1944849.

[52] Thomas Engelhardt and Carsten Dachsbacher. Epipolar sampling for shadows and crepuscular rays in participating media with single scattering. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, I3D '10, pages 119–125. ACM, 2010. doi: 10.1145/1730804.1730823.

[53] Manfred Ernst, Marc Stamminger, and Gunther Greiner. Filter Importance Sampling. *2006 IEEE Symposium on Interactive Ray Tracing*, pages 125–132, 2006. doi: 10.1109/RT.2006.280223.

[54] Shaohua Fan, Stephen Chenney, and Yu-chi Lai. Metropolis Photon Sampling with Optional User Guidance. In *Rendering Techniques '05 (Proceedings of the 16th Eurographics Symposium on Rendering)*, pages 127–138. Eurographics Association, 2005.

[55] Jean-Philippe Farrugia and Bernard Péroche. A progressive rendering algorithm using an adaptive perceptually based image metric. *Computer Graphics Forum*, 23:605–614, 2004. doi: 10.1.1.104.2302.

[56] Jean Baptiste Joseph Fourier. *Théorie du mouvement de la chaleur dans les corps solides*, volume IV, pages 153–246. 1824. URL `http://gallica.bnf.fr/ark:/12148/bpt6k33707/f6`.

[57] Jeppe Revall Frisvad. Importance sampling the Rayleigh phase function. *Journal of the Optical Society of America A*, 28(12):2436, November 2011.

[58] Claude Gasquet and Patrick Witomski. *Analyse de Fourier et applications : filtrage, calcul numérique, ondelettes*. Sciences sup. Dunod, Paris, 2000. ISBN 2-10-005018-4. Autre tirage : 2001.

[59] Iliyan Georgiev, Krivanek Jaroslav, Davidovic Tomas, and Philipp Slusallek. Bidirectional Light Transport with Vertex Connection and Merging. In *Proceedings of SIGGRAPH Asia 2012*, 2012.

[60] A. Gerrad and J.M. Bruch. *Introduction to Matrix Methods in Optics*. Dover, 1975. ISBN 978-0486680446.

[61] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. *ACM SIGGRAPH Computer Graphics*, 18(3):213–222, July 1984. doi: 10.1145/964965.808601.

[62] Baining Guo. Progressive radiance evaluation using directional coherence maps. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques - SIGGRAPH '98*, pages 255–266. ACM Press, July 1998. doi: 10.1145/280814.280888.

[63] Toshiya Hachisuka and Henrik Wann Jensen. Stochastic progressive photon mapping. *ACM Transactions on Graphics*, 28(5):1, December 2009. ISSN 0730-0301. doi: 10.1145/1618452.1618487.

[64] Toshiya Hachisuka, Wojciech Jarosz, Richard Peter Weistroffer, Kevin Dale, Greg Humphreys, Matthias Zwicker, and Henrik Wann Jensen. Multidimensional adaptive sampling and reconstruction for ray tracing. *ACM Transactions on Graphics*, 27(3), 2008. doi: 10.1145/1360612.1360632.

[65] Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. Progressive photon mapping. *ACM Transactions on Graphics*, pages 130:1–130:8, 2008. doi: 10.1145/1457515.1409083.

[66] Toshyia Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. A Path Space Extension for Robust Light Transport Simulation. Technical report, NVIDIA, 2012. URL `http://cs.au.dk/~toshiya/ups_tr.pdf`.

[67] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. ISSN 00063444. URL `http://www.jstor.org/stable/2334940`.

[68] Vlastimil Havran, Jiří Bittner, and Hans-Peter Seidel. Ray maps for global illumination. In *ACM SIGGRAPH 2004 Sketches on - SIGGRAPH '04*, page 77. ACM Press, August 2004. doi: 10.1145/1186223.1186319.

[69] Vlastimil Havran, Robert Herzog, and Hans-Peter Seidel. Fast Final Gathering via Reverse Photon Mapping. *Computer Graphics Forum*, 24 (3):323–333, 2005.

[70] Miloš Hašan, Fabio Pellacini, and Kavita Bala. Matrix row-column sampling for the many-light problem. *ACM Transactions on Graphics*, 26 (3):26, August 2007. doi: 10.1145/1275808.1276410.

[71] Miloš Hašan, Jaroslav Křivánek, Bruce Walter, and Kavita Bala. Virtual spherical lights for many-light rendering of glossy scenes. *ACM Transactions on Graphics*, 28(5):1, December 2009. doi: 10.1145/1661412.1618489.

[72] Paul S Heckbert. Survey of texture mapping. *IEEE Computer Graphics and Applications*, 6(11):56–67, November 1986. ISSN 0272-1716. doi: 10.1109/MCG.1986.276672.

[73] Paul S. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. *ACM SIGGRAPH Computer Graphics*, 24(4):145–154, September 1990. doi: 10.1145/97880.97895.

[74] Eric Heitz and Fabrice Neyret. Representing Appearance and Pre-filtering Subpixel Data in Sparse Voxel Octrees. In *High Performance Graphics*, June 2012.

[75] L. C. Henyey and J. L. Greenstein. Diffuse radiation in the Galaxy. *The Astrophysical Journal*, 93:70, January 1941. doi: 10.1086/144246.

[76] Christophe Hery and Ravi Ramamoorthi. Importance Sampling of Reflection from Hair Fibers. *Journal of Computer Graphics Techniques (JCGT)*, 1(1):1–17, June 2012. URL `http://jcgt.org/published/0001/01/01/`.

[77] Robert Herzog, Havran Vlastimil, Kinuwaki Shinichi, Myszkowski Karol, and Seidel Hans-Peter. Global Illumination using Photon Ray Splatting. *Computer Graphics Forum*, 26:503 – 513, 2007.

[78] Robert Herzog, Karol Myszkowski, and Hans-Peter Seidel. Anisotropic Radiance-Cache Splatting for Efficiently Computing High-Quality Global Illumination with Lightcuts. *Computer Graphics Forum*, 28(2):259–268, 2009.

[79] Heinrich Hey and Werner Purgathofer. Advanced radiance Estimation for Photon Map Global Illumination. *Computer Graphics Forum*, 21(3), 2002. URL `http://www.cg.tuwien.ac.at/research/publications/2002/Hey-2002-Adv/`.

[80] Nicolas Holzschuch and François X. Sillion. Accurate Computation of the Radiosity Gradient with Constant and Linear Emitters. In *Rendering Techniques 1995 (Eurographics Symposium on Rendering)*, pages 186–195, Dublin, Ireland, June 1995.

[81] Homan Igehy. Tracing ray differentials. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH '99*, pages 179–186. ACM Press, July 1999. doi: 10.1145/311535.311555.

[82] Akira Ishimaru. *Wave Propagation and Scattering in Random Media*. Academic Press, New York, 1978. ISBN 978-0-7803-4717-5.

[83] Wenzel Jakob and Steve Marschner. Manifold exploration: A markov chain monte carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics*, 51, July 2012. doi: 10.1145/2185520.2185554.

[84] Wenzel Jakob, Adam Arbree, Jonathan T. Moon, Kavita Bala, and Steve Marschner. A radiative transfer framework for rendering materials with anisotropic structure. *ACM Transactions on Graphics*, 29(4):1, July 2010. doi: 10.1145/1778765.1778790.

[85] Wenzel Jakob, Christian Regg, and Wojciech Jarosz. Progressive expectation–maximization for hierarchical volumetric photon mapping. *Computer Graphics Forum*, 30(4), June 2011.

[86] Wojciech Jarosz, Matthias Zwicker, and Henrik Wann Jensen. The beam radiance estimate for volumetric photon mapping. *Computer Graphics Forum (Proceedings of Eurographics 2008)*, 27(2):557–566, April 2008.

[87] Wojciech Jarosz, Matthias Zwicker, and Henrik Wann Jensen. Irradiance Gradients in the Presence of Participating Media and Occlusions. *Computer Graphics Forum (Proceedings of EGSR 2008)*, 27(4):1087–1096, June 2008.

[88] Wojciech Jarosz, Derek Nowrouzezahrai, Iman Sadeghi, and Henrik Wann Jensen. A comprehensive theory of volumetric radiance estimation using photon points and beams. *ACM Transactions on Graphics*, 30(1):1–19, 2011. doi: 10.1145/1899404.1899409.

[89] Wojciech Jarosz, Derek Nowrouzezahrai, Robert Thomas, Peter-Pike Sloan, and Matthias Zwicker. Progressive photon beams. *ACM Transactions on Graphics*, 30(6):1, December 2011. doi: 10.1145/2070781.2024215.

[90] Wojciech Jarosz, Volker Schönefeld, Leif Kobbelt, and Henrik Wann Jensen. Theory, Analysis and Applications of 2D Global Illumination. *ACM Transactions on Graphics*, 31, May 2012. to-appear.

[91] Henrik Wann Jensen and Niels JÃÿrgen Christensen. Photon maps in bidirectional monte carlo ray tracing of complex objects. *Computers Graphics*, 19(2):215 – 224, 1995. doi: 10.1016/0097-8493(94)00145-O.

[92] Henrik Wann Jensen and Per H. Christensen. Efficient simulation of light transport in scences with participating media using photon maps. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques - SIGGRAPH '98*, pages 311–320. ACM Press, July 1998. doi: 10.1145/280814.280925.

[93] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*, pages 511–518. ACM Press, August 2001. doi: 10.1145/383259.383319.

[94] H Kahn and A W Marshall. Methods of reducing sample size in Monte Carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278, 1953. URL `http://www.jstor.org/stable/166789`.

[95] James T. Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '86, pages 143–150. ACM, 1986. doi: 10.1145/15922.15902.

[96] Csaba Kelemen, Laszlo Szirmay-Kalos, Gyorgy Antal, and Ferenc Csonka. A Simple and Robust Mutation Strategy for the Metropolis Light Transport Algorithm. *Computer Graphics Forum*, 21(3):531–540, 2002. doi: 10.1111/1467-8659.00703.

[97] Alexander Keller. Instant radiosity. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 49–56. ACM, August 1997. doi: 10.1145/258734.258769.

[98] Claude Knaus and Matthias Zwicker. Progressive photon mapping: A probabilistic approach. *ACM Transactions on Graphics*, 30:25:1–25:13, May 2011. doi: 10.1145/1966394.1966404.

[99] Thomas Kollig and Alexander Keller. Efficient illumination by high dynamic range images. In Per H. Christensen, Daniel Cohen-Or, and Stephen N. Spencer, editors, *Proceedings of the 14th Eurographics Workshop on Rendering Techniques, Leuven, Belgium, June 25-27, 2003*, volume 44 of *ACM International Conference Proceeding Series*, pages 45–51. Eurographics Association, 2003.

[100] Christopher Kulla and Marcos Fajardo. Importance Sampling Techniques for Path Tracing in Participating Media. *Computer Graphics Forum*, 31 (4):1519—-1528, 2012. doi: 10.1111/j.1467-8659.2012.03148.x.

[101] Jaroslav Křivánek, Pascal Gautron, Sumanta Pattanaik, and Kadi Bouatouch. Radiance caching for efficient global illumination computation. *IEEE transactions on visualization and computer graphics*, 11(5):550–61, January 2005. doi: 10.1145/1079840.1080029.

[102] Jaroslav Křvánek, Pascal Gautron, Kadi Bouatouch, and Sumanta Pattanaik. Improved radiance gradient computation. In *Proceedings of the 21st spring conference on Computer graphics - SCCG '05*, page 155. ACM Press, May 2005. doi: 10.1145/1090122.1090148.

[103] Eric P. Lafortune and Yves D. Willems. Bi-Directional Path Tracing. *Proceedings of Compugraphics âĂŹ93*, pages 145–153, 1993.

[104] Eric P Lafortune and Yves D Willems. A 5D Tree to Reduce the Variance of Monte Carlo Ray Tracing. In *Proceedings of the 6th Eurographics Workshop on Rendering*, pages 11–20, Dublin, Ireland, 1995. doi: 10.1.1.23.2520.

[105] Eric P. Lafortune and Yves D. Willems. Rendering Participating Media with Bidirectional Path Tracing. In Xavier Pueyo and Peter Schroder, editors, *Rendering Techniques '96, Proceedings of the Eurographics*, pages 91–100, Porto, 1996. Springer.

[106] Paul Lalonde and Alain Fournier. Generating Reflected Directions from BRDF Data. *Computer Graphics Forum*, 16(3):293–300, 1997. doi: 10.1.1.57.5153.

[107] Douglas Lanman, Ramesh Raskar, Amit Agrawal, and Gabriel Taubin. Shield fields: modeling and capturing 3D occluders. *ACM Transactions on Graphics*, 27(5):131:1–131:10, December 2008. doi: 10.1145/1457515.1409084.

[108] M Lastra, C Ureña, J. Revelles, and R. Montes. A particle-path based method for Monte-Carlo density estimation. In P. Debevec and S. Gibson, editors, *Thirteenth Eurographics Workshop on Rendering (2002)*, Pisa, Italy, 2002. Eurographics Association. URL `http://lsi.ugr.es/rosana/investigacion/papers/lastra-egrw02.pdf`.

[109] Fabien Lavignotte and Mathias Paulin. Scalable photon splatting for global illumination. In *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Austalasia and South East Asia - GRAPHITE '03*, page 203. ACM Press, February 2003. doi: 10.1145/604471.604511.

[110] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Efficient BRDF importance sampling using a factored representation. *ACM Transactions on Graphics*, 23(3):496, August 2004. doi: 10.1145/1015706.1015751.

[111] Jaakko Lehtinen, Timo Aila, Jiawen Chen, Samuli Laine, and Frédo Durand. Temporal light field reconstruction for rendering distribution effects. *ACM Transactions on Graphics*, 30(4), 2011.

[112] Jaakko Lehtinen, Timo Aila, Samuli Laine, and Frédo Durand. Reconstructing the indirect light field for global illumination. *ACM Transaction on Graphics*, 31(4), 2012.

[113] Stephen R. Marschner, Henrik Wann Jensen, Mike Cammarano, Steve Worley, and Pat Hanrahan. Light scattering from human hair fibers. *ACM Transactions on Graphics*, 22(3):780, July 2003. doi: 10.1145/882262.882345.

[114] Wojciech Matusik. *A Data-Driven Reflectance Model*. PhD thesis, Massachusetts Institute of Technology, 2003. URL `http://people.csail.mit.edu/wojciech/pubs/phdthesis.pdf`.

[115] Michael D. McCool. Anisotropic diffusion for Monte Carlo noise reduction. *ACM Transactions on Graphics*, 18(2):171–194, April 1999. doi: 10.1145/318009.318015.

[116] Nicholas Metropolis. The begining of the Monte Carlo method. *Los Alamos Science*, (Special Issue), 1987.

[117] Mark Meyer and John Anderson. Statistical acceleration for animated global illumination. *ACM Transactions on Graphics*, 25(3):1075, July 2006. doi: 10.1145/1141911.1141996.

[118] Don P Mitchell. Generating antialiased images at low sampling densities. *ACM SIGGRAPH Computer Graphics*, 21(4):65–72, 1987. doi: 10.1145/37401.37410.

[119] Don P. Mitchell. Consequences of stratified sampling in graphics. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, pages 277–280. ACM Press, 1996. doi: 10.1145/237170.237265.

[120] Don P. Mitchell and Arun N. Netravali. Reconstruction filters in computer-graphics. *ACM SIGGRAPH Computer Graphics*, 22(4):221–228, August 1988. doi: 10.1145/378456.378514.

[121] Rosana Montes, Carlos Ure, and Miguel Lastra. Generic BRDF Sampling: a Sampling Method for Global Illumination and Monte-Carlo. In *Proceedings of the Third International Conference on Computer Graphics Theory and Applications - GRAPP 2008*, pages 191–198, Funchal, Madeira, Portugal, 2008. Springer.

[122] Rosana Montes Soldado and Carlos Ureña Almagro. An Overview of BRDF Models. Technical report, University of Granada, Granada, 2012. URL `http://digibug.ugr.es/handle/10481/19751`.

[123] Mukesh C. Motwani, Mukesh C Motwani, Mukesh C Gadiya, Rakhi C Motwani, Frederick C Harris, and Jr. Survey of image denoising techniques. In *Proceedings of Global Signal Processing Expo (GSPx)*, 2004. doi: 10.1.1.107.1823.

[124] Fabrice Neyret. Modeling Animating and Rendering Complex Scenes using Volumetric Textures. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):55–70, 1998. doi: 10.1109/2945.675652.

[125] Fred E Nicodemus. Directional Reflectance and Emissivity of an Opaque Surface. *Applied Optics*, 4(7):767–773, July 1965. doi: 10.1364/AO.4.000767.

[126] Jan Novák, Derek Nowrouzezahrai, Carsten Dachsbacher, and Wojciech Jarosz. Virtual ray lights for rendering scenes with participating media. *ACM Transactions on Graphics*, 31(4):1–11, July 2012. ISSN 07300301. doi: 10.1145/2185520.2185556.

[127] Jan Novák, Derek Nowrouzezahrai, Carsten Dachsbacher, and Wojciech Jarosz. Progressive Virtual Beam Lights. *Computer Graphics Forum*, 31(4):1407–1413, June 2012. ISSN 01677055. doi: 10.1111/j.1467-8659.2012.03136.x.

[128] Victor Ostromoukhov, Charles Donohue, and Pierre-Marc Jodoin. Fast Hierarchical Importance Sampling with Blue Noise Properties. *ACM Transactions on Graphics*, 23(3):488–495, 2004.

[129] Ryan S. Overbeck, Craig Donner, and Ravi Ramamoorthi. Adaptive wavelet rendering. *ACM Transactions on Graphics*, 28(5):140:1–140:12, December 2009. doi: 10.1145/1618452.1618486.

[130] Art Owen and Yi Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000. URL `http://www.jstor.org/stable/10.2307/2669533`.

[131] Romain Pacanowski, Oliver Salazar Celis, Christophe Schlick, Xavier Granier, Pierre Poulin, and Annie Cuyt. Rational BRDF. *IEEE transactions on visualization and computer graphics*, February 2012. URL `http://hal.inria.fr/hal-00678885/en`.

[132] James Painter and Kenneth Sloan. Antialiased ray tracing by adaptive progressive refinement. In *SIGGRAPH '89 Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 281–288, 1989. doi: 10.1145/74334.74362.

[133] Anthony Pajot, Loïc Barthe, Mathias Paulin, and Pierre Poulin. Representativity for robust and adaptive multiple importance sampling. *IEEE transactions on visualization and computer graphics*, 17(8):1108–21, August 2011. doi: 10.1145/1999161.1999196.

[134] S. N. Pattanaik and S. P. Mudur. Computation of global illumination in a participating medium by monte carlo simulation. *The Journal of Visualization and Computer Animation*, 4(3):133–152, July 1993. URL `10.1002/vis.4340040303`.

[135] Mark Pauly, Thomas Kollig, and Alexander Keller. Metropolis Light Transport for Participating Media. *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*, pages 11–22, 2000.

[136] Matt Pharr and Greg Humphreys. *Physically Based Rendering, Second Edition: From Theory To Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2010. ISBN 0123750792, 9780123750792.

[137] Michel Plancherel and Mittag Leffler. Contribution à l'Étude de la représentation dâĂŹune fonction arbitraire par des inégrales définies. *Rendiconti del Circolo Matematico di Palermo (1884 - 1940)*, 30:289–335, 1910. URL `http://dx.doi.org/10.1007/BF03014877`. 10.1007/BF03014877.

[138] Simon Premože, Michael Ashikhmin, and Peter Shirley. Path integration for light transport in volumes. In Philip Dutré, Frank Suykens, Per H. Christensen, and Daniel Cohen-Or, editors, *14th Eurographics Symposium on Rendering*, pages 52–63, Leuven, Belgium, June 2003. Eurographics Association. ISBN 3-905673-03-7.

[139] Yu. V. Prokhorov. Convergence of Random Processes and Limit Theorems in Probability Theory. *Theory of Probability and its Applications*, 1(2):157, November 1956. doi: 10.1137/1101016. URL `http://link.aip.org/link/?TPRBAU/1/157/1`.

[140] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for reflection. *ACM Transactions on Graphics*, 23(4):1004–1042, October 2004. doi: 10.1145/1027411.1027416.

[141] Ravi Ramamoorthi, Dhruv Mahajan, and Peter Belhumeur. A first-order analysis of lighting, shading, and shadows. *ACM Transactions on Graphics*, 26(1), January 2007. doi: 10.1145/1189762.1189764.

[142] Mickaël Ribardière, Samuel Carré, and Kadi Bouatouch. Adaptive records for volume irradiance caching. *The Visual Computer*, 27(6-8): 655–664, April 2011. doi: 10.1007/s00371-011-0573-z.

[143] T. Ritschel, T. Engelhardt, T. Grosch, H.-P. Seidel, J. Kautz, and C. Dachsbacher. Micro-rendering for scalable, parallel final gathering. *ACM Transactions on Graphics*, 28(5):1, December 2009. URL `10.1145/1618452.1618478`.

[144] Fabrice Rousselle, Petrik Clarberg, Luc Leblanc, Victor Ostromoukhov, and Pierre Poulin. Efficient product sampling using hierarchical thresholding. *The Visual Computer*, 24(7-9):465–474, June 2008. doi: 10.1007/s00371-008-0227-y.

[145] Fabrice Rousselle, Claude Knaus, and Matthias Zwicker. Adaptive Sampling and Reconstruction using Greedy Error Minimization. *ACM Transactions on Graphics (TOG)*, 30(6):1—-12, 2011.

[146] Holly E. Rushmeier and Gregory J. Ward. Energy preserving non-linear filters. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques - SIGGRAPH '94*, pages 131–138. ACM Press, July 1994. doi: 10.1145/192161.192189.

[147] Szymon Rusinkiewicz. A New Change of Variables for Efficient BRDF Representation. In G Drettakis and N Max, editors, *Rendering Techniques '98*, Eurographics Book Series. Springer-Verlag, Wien, Austria, 1998.

[148] Iman Sadeghi, Adolfo Munoz, Philip Laven, Wojciech Jarosz, Francisco Seron, Diego Gutierrez, and Henrik Wann Jensen. Physically-based simulation of rainbows. *ACM Transactions on Graphics*, 31(1):1–12, January 2012. ISSN 07300301. doi: 10.1145/2077341.2077344.

[149] Mateu Sbert, Miquel Feixas, Ivan Viola, Jaume Rigau, and Miguel Chover. Information theory in computer graphics and visualization. In *SIGGRAPH Asia 2011 Courses on - SA '11*, pages 1–58. ACM Press, December 2011. doi: 10.1145/2077434.2077443.

[150] L. Schjøth, J. Sporring, and O. Fogh Olsen. Diffusion Based Photon Mapping. *Computer Graphics Forum*, 27(8):2114–2127, December 2008. doi: 10.1111/j.1467-8659.2008.01196.x.

[151] Lars Schjøth, Jeppe Revall Frisvad, Kenny Erleben, and Jon Sporring. Photon differentials. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, GRAPHITE '07, pages 179–186. ACM, 2007. doi: 10.1145/1321261.1321293.

[152] Lars Schjøth, Jeppe Revall Frisvad, Kenny Erleben, and Jon Sporring. Photon Differentials in Space and Time. In *Computer Vision, Imaging and Computer Graphics*, pages 274–286, 2011. URL `http://www.imm.dtu.dk/English/Research/Image_Analysis_and_Computer_Graphics/Publications.aspx?lg=showcommon&id=9da1fdfe-4134-4eaf-84b9-95beaf8f7370`.

[153] Roland Schregle. Bias Compensation for Photon Maps. *Computer Graphics Forum*, 22(4):729–742, December 2003. doi: 10.1111/j.1467-8659.2003.00720.x.

[154] Laurent Schwartz. Théorie des distributions et transformation de Fourier. Technical Report 23, Annales de l'université de Grenoble, 1947. URL `http://www.numdam.org/item?id=AUG_1947-1948__23__7_0`.

[155] B Segovia, J C Iehl, and B Péroche. Coherent Metropolis Light Transport with Multiple-Try Mutations. Technical report, LIRIS, 2007.

[156] Benjamin Segovia, Jean-Claude Iehl, and Bernard Péroche. Metropolis Instant Radiosity. *Computer Graphics Forum*, 26(3):425–434, 2007.

[157] Pradeep Sen and Soheil Darabi. Compressive estimation for signal integration in rendering. *Computer Graphics Forum*, 29(4):1355–1363, August 2010. doi: 10.1111/j.1467-8659.2010.01731.x.

[158] Pradeep Sen and Soheil Darabi. Compressive rendering: a rendering application of compressed sensing. *IEEE transactions on visualization and computer graphics*, 17(4):487–99, April 2011. doi: 10.1145/1957382.1957602.

[159] Pradeep Sen and Soheil Darabi. On filtering the noise from the random parameters in Monte Carlo rendering. *ACM Transactions on Graphics*, 31(3):1–15, May 2012. doi: 10.1145/2167076.2167083.

[160] Claude E. Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27, July 1948.

[161] Peter Shirley, Bretton Wade, Philip M Hubbard, David Zareski, Bruce Walter, and Donald P Greenberg. Global Illumination via Density-Estimation. In *Proceedings of the 6th Workshop on Rendering*, pages 219 – 230, 1995.

[162] Peter Shirley, Timo Aila, Jonathan Cohen, Eric Enderton, Samuli Laine, David Luebke, and Morgan McGuire. A local image reconstruction algorithm for stochastic rendering. In *Symposium on Interactive 3D Graphics and Games on - I3D '11*, page 9. ACM Press, February 2011. doi: 10.1145/1944745.1944747.

[163] B W Silverman. *Kernel density estimation techniques for statistics and data analysis*. Chapman and Hall, 1986. ISBN 978-0412246203.

[164] Cyril Soler, Kartic Subr, Frédo Durand, Nicolas Holzschuch, and François X. Sillion. Fourier depth of field. *ACM Transactions on Graphics*, 28(2):18:1–18:12, May 2009. ISSN 0730-0301. doi: 10.1145/1516522.1516529.

[165] B. Spencer and M.W. Jones. Into the Blue: Better Caustics through Photon Relaxation. *Computer Graphics Forum*, 28(2):319–328, April 2009. doi: 10.1111/j.1467-8659.2009.01371.x.

[166] Ben Spencer and Mark W Jones. Hierarchical photon mapping. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):49–61, 2009. URL `http://hdl.handle.net/10512/99999177`.

[167] Jos Stam. Multiple scattering as a diffusion process. In P. M. Hanrahan and W. Purgathofer, editors, *Proceedings of the 6th Eurographics Workshop on Rendering*, pages 41–50. Springer Verlag, June 1995.

[168] Xin Sun, Kun Zhou, Stephen Lin, and Baining Guo. Line space gathering for single scattering in large scenes. *ACM Transactions on Graphics*, pages 54:1–54:8, 2010. doi: 10.1145/1833349.1778791.

[169] Frank Suykens and Yves D. Willems. Density Control for Photon Maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*, pages 23–34, June 2000. doi: 10.1145/647652.732120.

[170] Eric Tabellion and Arnauld Lamorlette. An approximate global illumination system for computer generated films. *ACM Transactions on Graphics*, 23(3):469, August 2004. doi: 10.1145/1015706.1015748.

[171] Justin F Talbot, David Cline, and Parris Egbert. Importance Resampling for Global Illumination. In Kavita Bala and Philip Dutré, editors, *Rendering Techniques 2005 Eurographics Symposium on Rendering*, pages 139–146, Konstanz, Germany, 2005. Eurographics Association.

[172] J. Tessendorf. Radiative transfer as a sum over paths. *Physical Review A*, 35(2):872–878, January 1987. URL `http://pra.aps.org/abstract/PRA/v35/i2/p872_1`.

[173] Robert F. Tobler and Stefan Maierhofer. Improved Illumination Estimation for Photon Maps in Architectural Scenes. In *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG '06)*, pages 257–261, 2006.

[174] Nikolai Nikolaevich Vakhania, Vazha Izemovich Tarieladze, and Sergei Akopovich Chobanian. *Probability distributions on Banach spaces*. Springer Science & Business, 1987. ISBN 9027724962. URL `http://books.google.com/books?hl=en&lr=&id=unJ5T5aa2fEC&pgis=1`.

[175] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, December 1997. URL `http://graphics.stanford.edu/papers/veach_thesis/`.

[176] Eric Veach and Leonidas Guibas. Bidirectional Estimators for Light Transport. In *Proceedings of the Fifth Eurographics Workshop on Rendering*, pages 147–162. Eurographics, June 1994.

[177] Eric Veach and Leonidas J. Guibas. Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* -

SIGGRAPH '95, pages 419–428. ACM Press, September 1995. doi: 10.1145/218380.218498.

[178] Eric Veach and Leonidas J. Guibas. Metropolis light transport. In *Proceedings of the 24th Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '97*, 1997. doi: 10.1145/258734.258775.

[179] Romain Vergne, Pascal Barla, Roland Fleming, and Xavier Granier. Surface Flows for Image-based Shading Design. *ACM Transactions on Graphics*, 31(4):94:1–94:9, August 2012. doi: 10.1145/2185520.2185590.

[180] Kuntee Viriyothai and Paul Debevec. Variance minimization light probe sampling. In *SIGGRAPH '09: Posters on - SIGGRAPH '09*, pages 1–1. ACM Press, August 2009. doi: 10.1145/1599301.1599393.

[181] Bruce Walter, Philip M. Hubbard, Peter Shirley, and Donald P. Greenberg. Global illumination using local linear density estimation. *ACM Transactions on Graphics*, 16(3):217–259, July 1997. doi: 10.1145/256157.256158.

[182] Bruce Walter, Sebastian Fernandez, Adam Arbree, Kavita Bala, Michael Donikian, and Donald P. Greenberg. Lightcuts: a scalable approach to illumination. *ACM Transactions on Graphics*, 24(3):1098–1107, July 2005. doi: 10.1145/1073204.1073318.

[183] Bruce Walter, Adam Arbree, Kavita Bala, and Donald P. Greenberg. Multidimensional lightcuts. *ACM Transactions on Graphics*, 25(3):1081, July 2006. doi: 10.1145/1141911.1141997.

[184] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the Eurographics Symposium on Rendering Techniques, Grenoble, France, 2007*, pages 195–206, 2007.

[185] Rui Wang and Oskar Åkerlund. Bidirectional Importance Sampling for Unstructured Direct Illumination. *Computer Graphics Forum*, 28(2):269–278, 2009. doi: 10.1111/j.1467-8659.2009.01366.x.

[186] Gregory J. Ward and Paul S. Heckbert. Irradiance gradients. In *Proceedings of the 2nd Eurographics Workshop on Rendering*, page 1. ACM Press, 1992. doi: 10.1145/1401132.1401225.

[187] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. A ray tracing solution for diffuse interreflection. *ACM SIGGRAPH Computer Graphics*, 22(4):85–92, August 1988. doi: 10.1145/378456.378490.

[188] Markus Weber, Marco Milch, Karol Myszkowski, Kirill Dmitriev, Przemyslaw Rokita, and Hans-Peter Seidel. Spatio-Temporal Photon Density Estimation Using Bilateral Filtering. In Daniel Cohen-Or, Lakhmi Jain, and Nadia Magnenat-Thalmann, editors, *Computer Graphics International (CGI 2004)*, pages 120–127, Crete, Greece, 2004. IEEE. URL `http://www.mpi-inf.mpg.de/resources/anim/STdenest/index.htm`.

[189] Yonghao Yue, Kei Iwasaki, Bing-Yu Chen, Yoshinori Dobashi, and Tomoyuki Nishita. Unbiased, adaptive stochastic sampling for rendering inhomogeneous participating media. *ACM Transactions on Graphics*, 29 (6):177:1–177:8, December 2010. doi: 10.1145/1882261.1866199.

[190] Arno Zinke and Andreas Weber. Efficient ray based global illumination using photon maps. In L. Kobbelt, T. Kuhlen, T. Aach, and R. Westermann, editors, *Vision, Modeling, and Visualization 2006 (VMV 2006)*, pages 113–120. Akademische Verlagsgesellschaft Aka GmbH, Berlin, November 2006.

A | Detailed Proofs for Operators

A.1 Non-Planar Visibility Spectrum

We start from the definition of the partial visibility functions (Section 3.3.2):

$$r_{t,d}(\delta x, \delta\theta) = \begin{array}{l} 1 \text{ if } \delta x + t\delta y > -d \\ 0 \text{ else} \end{array}$$

The Fourier transform of a partial visibility function is (In this case $r_{t_{min},d}$):

$$\begin{aligned}
\hat{r}_{t_{min},d}(\Omega_x, \Omega_\theta) &= \int_{\delta\theta}\int_{\delta x} r_{t,d}(\delta x, \delta\theta) e^{-2i\pi\delta\theta\Omega_\theta} e^{-2i\pi\delta x\Omega_x} \mathrm{d}\delta x \mathrm{d}\delta\theta \\
&= \int_{\delta x}\int_{\delta\theta=t_{min}\delta\theta-d}^{+\infty} e^{-2i\pi\delta\theta\Omega_\theta} e^{-2i\pi\delta x\Omega_x} \mathrm{d}\delta x \mathrm{d}\delta\theta \\
&= \int_{\delta\theta} e^{-2i\pi\delta\theta\Omega_\theta} \left[\frac{e^{-2i\pi\delta x\Omega_x}}{-2i\pi\Omega_x}\right]_{t_{min}\delta\theta-d}^{+\infty} \mathrm{d}\delta\theta \\
&= \frac{e^{2i\pi d\Omega_x}}{2i\pi\Omega_x} \int_{\delta\theta} e^{-2i\pi(t_{min}\delta\theta\Omega_x + \delta\theta\Omega_\theta)} \mathrm{d}\delta\theta
\end{aligned}$$

It gives us the following formula for the spectrum of the partial visibility functions:

$$\boxed{\hat{r}_{t_{min},d}(\Omega_x, \Omega_\theta) = \frac{e^{2i\pi d\Omega_x}}{2i\pi\Omega_x}\delta(t_{min}\Omega_x + \Omega_\theta)} \tag{A.1}$$

$$\boxed{\hat{r}_{t_{max},d}(\Omega_x, \Omega_\theta) = \frac{e^{2i\pi d\Omega_x}}{2i\pi\Omega_x}\delta(t_{max}\Omega_x + \Omega_\theta)} \tag{A.2}$$

We calculate then the convolution of those two spectrum to obtain the Fourier spectrum of the visibility function:

$$\begin{aligned}
\hat{r}_{t_{max},d} \otimes \hat{r}_{t_{min},d}(\Omega_x, \Omega_\theta) &= \int_{\Omega'_x}\int_{\Omega'_\theta} \hat{r}_{t_{min},d}(\Omega'_x, \Omega'_\theta)\hat{r}_{t_{max},d}(\Omega_x - \Omega'_x, \Omega_\theta - \Omega'_\theta)\mathrm{d}\Omega'_\theta\mathrm{d}\Omega'_x \\
&= \int_{\Omega'_x}\int_{\Omega'_\theta} \left[\tfrac{e^{2i\pi d\Omega'_x}}{2i\pi\Omega'_x}\delta(t_{min}\Omega'_x + \Omega'_\theta)\right]\left[\tfrac{e^{2i\pi d(\Omega_x-\Omega'_x)}}{2i\pi(\Omega_x-\Omega'_x)}\delta(t_{max}(\Omega_x-\Omega'_x)+(\Omega_\theta-\Omega'_\theta))\right]\mathrm{d}\Omega'_\theta\mathrm{d}\Omega'_x
\end{aligned}$$

To evaluate the nested integrals, we have to solve the following equation system:

$$\begin{cases} t_{min}\Omega'_x + \Omega'_\theta = 0 \\ t_{max}(\Omega_x - \Omega'_x) + (\Omega_\theta - \Omega'_\theta) = 0 \end{cases} \tag{A.3}$$

It lead to the following equalities:

$$\begin{cases} \Omega'_\theta = -t_{min}\Omega'_x \\ \Omega'_x = \dfrac{t_{max}\Omega_x + \Omega_\theta}{t_{max} - t_{min}} \end{cases} \tag{A.4}$$

By evaluating the integrand at the position obtained from the system, we obtain the formula of the convolution:

$$\boxed{\hat{r}_{t_{max},d} \otimes \hat{r}_{t_{min},d}(\Omega_x, \Omega_\theta) = \frac{e^{2i\pi d\Omega_x}}{4\pi^2 \dfrac{(t_{max}\Omega_x + \Omega_\theta)(t_{min}\Omega_x + \Omega_\theta)}{(t_{max} - t_{min})^2}}} \tag{A.5}$$

A.2 Reparametrization onto Another Plane

We assume that the incoming local light-field and the outcoming local light-field are aligned. The problem of projection boils down to a flat-land problem (Figure A.1).

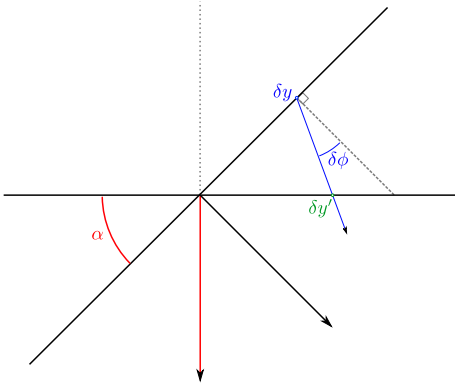


Figure A.1 *– The projection problem can be express in flat-land. Given an incoming light-field $\delta y, \delta\phi$, the formulation of the new position $\delta y'$ on the projection plane is approximated by a scale.*

To compute the value $\delta y'$ based on the inputs $(\delta y, \delta\phi, \alpha)$, we add intermediate steps: d, q and t (Figure A.2). First, we calculate length d:

$$d = \frac{\delta y \sin(\alpha)}{\cos(\alpha - \delta\phi)}$$

We chain the calculation to find q and t, based on intermediate variables expansion:

$$\begin{aligned} q &= \sin(\delta\phi)d \\ &= \delta y \sin(\delta\phi)\frac{\sin(\alpha)}{\cos(\alpha - \delta\phi)} \\ t &= \frac{q}{\cos(\alpha)} \\ &= \delta y \sin(\delta\phi)\frac{\tan(\alpha)}{\cos(\alpha - \delta\phi)} \end{aligned}$$

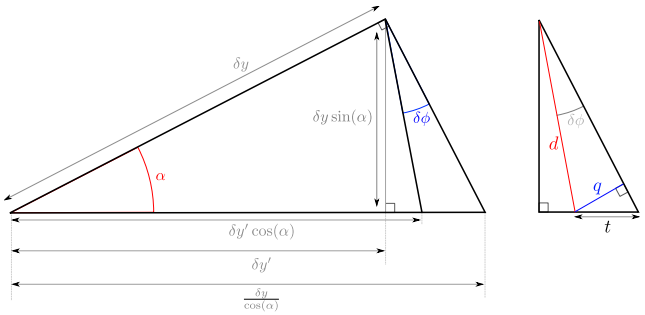


Figure A.2 – *We add intermediate steps to evaluate the length $\delta y'$. We will use a chain of right triangles to calculate d, q and then t based on the input of our problem.*

It gives us:

$$\delta y' = \delta y\Big(\frac{1}{\cos(\alpha)} - \sin(\delta\phi)\frac{\tan(\alpha)}{\cos(\alpha - \delta\phi)}\Big)$$

Which can be linearized:

$$\boxed{\delta y' = \frac{\delta y}{\cos(\alpha)}} \tag{A.6}$$

Equation A.6 is only valid for α not close to $\frac{\pi}{2}$. In such a context, the linearization is not possible. But we can keep the division by a cosine approach as the closer we get to $\frac{\pi}{2}$, the more the signal will be stretched.

B | Covariances of Scattering

In this appendice, we estimate the covariance of scattering functions. We neglect here the windowing term of phase functions. This results in an overestimate of the local frequency content.

B.1 Phong BRDF

The Phong BRDF is defined with respect to the reflected direction. As defined in 3.3.4.5, we can use the Fourier transform operator as a multiplication. The Phong BRDF has the following formulation:

$$\rho_s(\theta) = \frac{s+1}{2\pi} cos(\theta)^s \tag{B.1}$$

Where s is the *shininess* of the material.

This function is one dimensional, we will only provide the σ^2 of its Fourier transform. Note that since the covariance is define over *pdf*s, we have to normalize the Fourier transform. We define here $\mathcal{F}'[\rho_s]$ as the normalized Fourier transform. Assuming that $s \geq 2$, we can write:

$$\begin{aligned}
\sigma^2 &= \int_{\mu\in\mathbb{R}} \mu^2 \mathcal{F}'[\rho_s](\mu)\mathrm{d}\mu \\
&= \frac{1}{|\mathcal{F}[\rho_s]|} \int_{\mu\in\mathbb{R}} \mu^2 \int_{\theta\in\mathbb{R}} \rho_s(\theta) e^{-2i\pi\mu\theta}\mathrm{d}\theta\mathrm{d}\mu \\
&= \frac{1}{|\mathcal{F}[\rho_s]|} \int_{\theta\in\mathbb{R}} \rho_s(\theta) \int_{\mu\in\mathbb{R}} \mu^2 e^{-2i\pi\mu\theta}\mathrm{d}\mu\mathrm{d}\theta \\
&= \frac{1}{|\mathcal{F}[\rho_s]|} \int_{\theta\in\mathbb{R}} \rho_s(\theta)\delta^{(2)}(\theta)\mathrm{d}\theta \\
&= \frac{1}{|\mathcal{F}[\rho_s]|} \int_{\theta\in\mathbb{R}} \left(\frac{i}{2\pi}\right)^2 \rho_s^{(2)}(\theta)\delta(\theta)\mathrm{d}\theta \\
&= \frac{1}{|\mathcal{F}[\rho_s]|} - \frac{1}{4\pi^2}\rho_s^{(2)}(0) \\
&= \frac{1}{|\mathcal{F}[\rho_s]|} \frac{s(s+1)}{8\pi^3}
\end{aligned} \tag{B.2}$$

We can obtain the norm of the Fourier transform of ρ_s using again the

Fubini double integral theorem:

$$\begin{aligned}
\left|\mathcal{F}\left[\rho_s\right]\right| &= \int_{\mu\in\mathbb{R}} \mathcal{F}\left[\rho_s\right](\mu)\mathrm{d}\mu \\
&= \int_{\mu\in\mathbb{R}}\int_{\theta\in\mathbb{R}} \rho_s(\theta)e^{-2i\pi\mu\theta}\mathrm{d}\theta\mathrm{d}\mu \\
&= \int_{\theta\in\mathbb{R}} \rho_s(\theta)\int_{\mu\in\mathbb{R}} e^{-2i\pi\mu\theta}\mathrm{d}\mu\mathrm{d}\theta \\
&= \int_{\theta\in\mathbb{R}} \rho_s(\theta)\delta(\theta)\mathrm{d}\theta \\
&= \frac{s+1}{2\pi} \qquad \text{(B.3)} \\
& \qquad \text{(B.4)}
\end{aligned}$$

Combining Equation B.2 and Equation B.4, we obtain the following covariance for the Phong BRDF:

$$\boxed{\mathrm{cov}_{\theta,\theta}(\rho_s) = \frac{s}{4\pi^2}} \qquad \text{(B.5)}$$

B.2 Henyey-Greenstein Phase Function

We are interested in the covariance of the amplitude of the Fourier transform of the HG function ρ_g. This function being a purely real even function, its Fourier transform is purely real:

$$\mathrm{cov}(|\hat{\rho}_g|) = \mathrm{cov}(\hat{\rho}_g) = \int_{-\infty}^{\infty} \omega^2\mathcal{F}\left[\rho_g\right](\omega)\mathrm{d}\omega$$

We use Fubini theorem on integration order change:

$$\begin{aligned}
\mathrm{cov}(|\hat{\rho}_g|) &= \int_{\omega=-\infty}^{+\infty}\int_{\theta=-\infty}^{+\infty} \omega^2\rho_g(\theta)e^{-2i\pi\omega\theta}\mathrm{d}\omega\mathrm{d}\theta \\
&= \int_{\theta=-\infty}^{+\infty} \rho_g(\theta)\int_{\omega=-\infty}^{+\infty} \omega^2 e^{-2i\pi\omega\theta}\mathrm{d}\omega\mathrm{d}\theta
\end{aligned}$$

The Fourier transform of the power function x^n is the n^{th} derivative of the delta distribution:

$$\int_{f=-\infty}^{+\infty} \omega^2 e^{-2i\pi\omega\theta}d\omega = \frac{-1}{4\pi^2}\delta^{(2)}(\theta)$$

we have:

$$\begin{aligned}
\mathrm{cov}(|\hat{\rho}_g|) &= \int_{\theta=-\infty}^{+\infty} \rho_g(\theta)\frac{-1}{4\pi^2}\delta^{(2)}(\theta)d\theta \\
&= \frac{-1}{4\pi^2}\frac{d\rho_g(\theta)}{d\theta^2}(0) \\
&= \frac{3}{4\pi}\frac{|g|(1+|g|)}{(1-|g|)^4}
\end{aligned}$$