Eigenvalue estimation of light transport using Monte Carlo algorithms

Ronak Molazem

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique
Spécialité : Mathématiques et Informatique
Unité de recherche : Laboratoire Jean Kuntzmann

Estimation des valeurs propres du transport de la lumière à l'aide d'algorithmes de Monte Carlo

Eigenvalue estimation of light transport using Monte Carlo algorithms

Présentée par :

Ronak MOLAZEM

Direction de thèse :

Cyril SOLER — Directeur de thèse
CHARGE DE RECHERCHE, Université Grenoble Alpes

Rapporteurs :

DEREK NOWROUZEZAHAI
Professeur associé, McGill University
BRUNO LEVY
Directeur de recherche, INRIA CENTRE NANCY-GRAND-EST

Thèse soutenue publiquement le **30 septembre 2022**, devant le jury composé de :

CYRIL SOLER Chargé de recherche HDR, INRIA CENTRE GRENOBLE-RHONE-ALPES	Directeur de thèse
DEREK NOWROUZEZAHAI Professeur associé, McGill University	Rapporteur
BRUNO LEVY Directeur de recherche, INRIA CENTRE NANCY-GRAND-EST	Rapporteur
MATHIAS PAULIN Professeur des Universités, UNIVERSITE TOULOUSE 3 - PAUL SABATIER	Examinateur
VALERIE PERRIER Professeur des Universités, GRENOBLE INP	Présidente
CHRISTIAN LESSIG Professeur assistant, Otto-Von-Guericke Universität Magdeburg	Examinateur

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Cyril Soler for the continuous support of my Ph.D study, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor.
Besides, I would like to thank my thesis committee: Mr Bruno Lévy, Derek Nowrouzezahrai, Valérie Perrier, Mathias Paulin, and Christian Lessig, for their insightful comments and encouragement.
I would also like to thank my family for their continuous support during the time.

Eigenvalue estimation of light transport using Monte Carlo algorithms

Ronak Molazem

Abstract

Constructing solutions for the discrete form of rendering equation has been extensively studied, due to the fact that digital computations are restricted to finite representations, i.e. matrices. The disadvantage of such representations is their computational cost for accurate estimations. On the other hand, the behavior of an operator acting on an infinite-dimensional space may not be the same as its matrix form in some states. In this study, we develop methods to estimate dominant eigenvalues of light transport based on the continuous form of light transport. We commence this investigation by formulating light transport as an integral operator defined on an infinite-dimensional Hilbert space. Applying all the tools offered by functional analysis formalism, we are able to discover properties such as continuity, closedness, compactness, and Hilbert-Schmidt characteristic. We also demonstrate the effect of the compactness property of light transport to compromise the accuracy of the finite and infinite representation of light transport. Furthermore, we study the resolvent of light transport and discuss obstacles on the way to building resolvent-based tools for the spectral study of light transport. The resolvent study is highly connected to the Fredholm theory. In this framework, operators are kernel operators. Fredholm derived an infinite series expansion for his famous determinant. We illustrate how it is possible to transform computations (related to Fredholm determinant) for light transport, when light transport is Hilbert-Schmidt, in terms of path tracing techniques. Finally, we design a few methods that leverage path tracing techniques normally used in the production of images, in order to numerically compute approximations of the dominant eigenvalues of the operator.

Estimation des valeurs propres du transport de la lumière à l'aide d'algorithmes de Monte Carlo

Ronak Molazem

Résumé

La construction de methodes discrètes d'approximation de l'équation de rendu a été largement étudiée. Celles-ci impliquent des calculs numériques utilisant des représentations de dimension finie (des matrices), dont la taille suit le besoin en precision du resultat. La convergence de ces methodes est intrinsequement liee a la compacite de l'operateur de transport lumineux. Dans cette these, nous développons des méthodes pour estimer les valeurs propres dominantes de l'operateur de transport de la lumière, basées sur la forme continue de l'operateur. Nous formulons le transport de la lumière comme un opérateur intégral défini sur un espace de Hilbert de dimension infinie. Grace aux outils offerts par l'analyse fonctionnelle, nous analysons tout d'abord des propriétés telles que la continuité, la fermeture, la compacité et la caractéristique de Hilbert-Schmidt de l'operateur. Nous démontrons également l'impact de la compacité du transport de la lumière sur la précision des approximations finies de l'equation du rendu. De plus, nous étudions le résolvent de l'operateur de transport et discutons des possibilites qu'il offre pour l'étude spectrale du transport de la lumière. L'étude du résolvent est fortement liée à la théorie de Fredholm, qui propose une extension en série infinie pour son célèbre déterminant. Nous illustrons comment il est possible de transformer les calculs (liés au déterminant de Fredholm) pour le transport de la lumière, lorsque l'operateur de transport de lumière est Hilbert-Schmidt, sous forme d'integrales d'ensembles de chemin lumineux. Enfin, nous proposons plusieurs méthodes qui tirent parti des techniques de Path Tracing, normalement utilisées dans la production d'images, afin de calculer numériquement des approximations des valeurs propres dominantes de l'opérateur de transport.

Contents

List of Figures

List of Tables

Global notations

S	Surfaces of the scene
Ω	Half-sphere of outgoing directions (local frame)
ℓ	The ray space $S \times \Omega$
$\mathcal{L}_2$	The space of square–integrable functions
$\mathbb{H}$	The space of light distributions
$\mathcal{B}(\mathbb{H})$	The space of linear bounded operators from $\mathbb{H}$ to itself
$r(\cdot)$	The spectral radius of an operator (Def. 5.1.10)
$<,>_{\mathbb{H}}$	Dot product weighted by cosine (Eq. (3.30))
$\|\cdot\|_{\mathbb{H}}$	Norm induced by $\langle , \rangle_{\mathbb{H}}$
$L : \ell \to \mathbb{R}$	Light distribution function. $L \in \mathbb{H}$
$K : \mathbb{H} \to \mathbb{H}$	Global reflectance operator
$K_{\mathbf{x}} : L(\Omega_{\mathbf{x}}) \to L(\Omega_{\mathbf{x}})$	Local reflectance operator at $\mathbf{x}$
$G(\mathbf{x}, \mathbf{y})$	Geometric term between points $\mathbf{x}$ and $\mathbf{y}$
$\mathbf{G} : \mathbb{H} \to \mathbb{H}$	reparametrization operator
$T : \mathbb{H} \to \mathbb{H}$	Light transport operator ($T = K\mathbf{G}$)
$T_b : \mathcal{B} \to \mathcal{B}$	Radiant exitance transport operator
$\rho_d(\mathbf{x})$	Albedo at $\mathbf{x} \in S$
$\rho(\mathbf{x}, \omega_o, \omega_i)$	BRDF at $\mathbf{x}$ in directions ω_o and ω_i
$V(\mathbf{x}, \mathbf{y})$	Visibility function between points $\mathbf{x}$ and $\mathbf{y}$
$\kappa(\mathbf{x}, \mathbf{y})$	Integral kernel of $\mathbf{T}_b$
$p : \ell \to S$	Function mapping $(\mathbf{x}, \omega_i)$ to the point (and direction) seen from $\mathbf{x}$ in direction ω_i, with $p^2(\mathbf{x}, \omega_i) = (\mathbf{x}, \omega_i)$

Figure 1: Notations used in this manuscript.

Chapter 1

Introduction

During the past few centuries, mathematical models simulating the physical behavior of light propagation and its interaction with surfaces are developed by physicians. With the aid of computers, it is possible to evaluate these models numerically, given the material properties and the description of the environment. The process of creating images using numerical methods is called image synthesis and has become an active research area in computer graphics since 1970. Early applications of shading models come from the pivotal effort of Phong [Pho75], Bouknight [BK70], and Gouraud [Gou71]. All these models were *local* i.e. illumination caused by other surfaces was not considered. It has been established that considering illumination from other surfaces, *the global model* achieves greater realism. This model essentially solves the equilibrium of light energy in a scene by determining the visible surfaces.
The central purpose of any global illumination algorithm is to provide a solution for the so–called rendering equation. For example, Kajiya [Kaj86] illustrated that distribution ray tracing provides a solution to the rendering equation by the Monte Carlo algorithm, but it was Arvo [Arv95b] who developed the operator formalism for the rendering concept. Based on the same foundation, Veach [Vea98] built estimators for this equation using path tracing and the Markov chain concept, leading to constructing the Markov chain Monte Carlo estimator. For more restricted cases, considering purely diffuse materials, the radiosity method, derived by Gloral et al. [GTGB84], is a finite element method. For this special case, the type of solution is achieved by discretization, while Monte Carlo estimators build estimations without it. The cost of discretization increases dramatically for non-diffuse scenes, since not only patch elements but also directions are discretized. Therefore, building a light transport simulator without discretization requirement is beneficial from a practical point of view.
All these approaches approximate the solution of the rendering equation. Evaluation of these approximations can be analyzed by studying the spectral properties of the integral operator, contained in the rendering equation, that transforms the incident light to the reflected one. This operator is called *the light transport operator*.

Functional analysis is a well–known tool to study operators. Applying the knowledge developed in this framework, it is possible to discover the true nature of this operator as an operator acting on an infinite–dimensional space (the space of light distributions). Observing the light transport operator from this window will help answering the questions challenging the solutions based on finite dimensional representations (such as the solution obtained by radiosity methods) associated to an operator which is not necessarily a matrix.

1.1 The eigenvalue problem of light transport

Path tracing algorithms provide the solution by generating paths between light sources and the surface points in the scene. These algorithms can be applied for any material model and geometry. In general, path tracing algorithms with acceleration structures (such as kd–trees, and octrees) build an image with high–quality illumination effects faster. Despite the advent of GPUs to reduce the computation cost, the determination of the visibility between point surfaces still remains costly. Therefore, there are other families of approaches developed based on recorded data for interactive rendering, such as Precomputed Radiance Transfer (PRT) that records the transport matrix to handle complex illumination effects. PRT algorithms assume that a static scene is lit by infinitely far light sources i.e. environment maps. In the preprocessing step, the transfer functions are generated to transform the incident light into the output radiance. These transfer functions are usually represented in terms of some basis functions, such as Spherical Harmonics (SH), wavelet, spectral mesh basis, etc.
Dimension. From the computational point of view, studying the concept of dimension is crucial to constructing any light transport simulator. Dimensionality might be also expressed as the number of necessary eigenvalues required to represent the solution in terms of the eigendecomposition of the operator (under the assumption that this decomposition exists). It has been shown that the local light transport operator is a low–dimensional operator.
Considering PRT methods, the number of basis elements, required to represent the solution, might be defined as the dimension of local light transport. For example, Mahajan et al. [MSRB07] defined the dimension as the number of principal components required to approximate local light transport with a low dimensional subspace offered by PCA.
In application, studying the behavior of the scene when changing the lighting conditions is possible by the spectral study of the transport matrix, provided by discretization. Mahajan et al. [MSRB07] developed the concept of locally low dimensional light transport by searching for the eigenvalues of the covariance matrix of the canonical cases.
Considering all these approaches which attempt to study the light transport locally, there is a common requirement which is discretization. On the other hand, discretization is usually computationally expensive (even for Lambertian scenes).

1.2 Our contribution

In comparison to path tracing methods, the techniques based on discretization, such as radiosity methods, provide the solution in low rank approximations. However, the light transport operator is intrinsically an operator acting on an infinite–dimensional space, and may not be necessarily finite rank. Our main goal in this study is to develop the methods that compute eigenvalues without the need of discretization.
To achieve this goal, some necessary information about the spectrum of the operator, i.e. the spectral study, is indispensable. The spectral study of the global light transport provides the knowledge for understanding the behavior of this operator on a global scale since the eigenpair of the operator does not depend on specific parameters such as the location of light sources (or lighting condition) or the camera position. However, it may depend on the geometry of the scene and the material model. [1]

1.2.1 Spectral study of the light transport operator

Based on the continuous representation of the light transport operator, we apply functional analysis to obtain the necessary information on the spectrum of light transport, which will guide us to design our methods. In the following, we address some important facts about the light transport operator that are proved using functional analysis.
Eigenvalues of the light transport operator. When implementing the algorithms to extract eigenvalues, knowing that eigenvalues are real or complex, helps to simplify the complexity. For example, in the methods based on polynomials (Chapter 8), choosing an appropriate algorithm to extract the roots is highly dependent on such information. Therefore, similar to the proof of Baranoski et al. [BBR97] [2], we prove that the light transport operator in Lambertian scenes has real eigenvalues, Theorem 4.A.3.
Compactness of the light transport operator. Facing an operator acting over an infinite–dimensional space, the first question rises to the head is that to what extent does the operator behave like a matrix? From a mathematical point of view, this question is answered by checking the compactness property of the operator. Using this property, one may establish whether an estimation of the solution, provided by an algorithm, based on the discrete representation, converges to the actual solution. Considering light transport operator, we prove that the operator is not compact given an arbitrary geometry, Chapter 4. However, once the edges removed from the geometry, the operator becomes compact. In other words, the lack of compactness explains the failure of bounding the approximation error of radiosity methods for arbitrary lighting conditions of a scene.

[1] It should be emphasized here that the eigenpair of the global light transport is different from the eigenpair of the local light transport, which is obtained from matrices provided by discretization.

[2] However, the radiosity matrix in this paper is $I - T$.

Resolvent study. The resolvent formalism of an operator provides a framework to apply complex analysis to obtain information about the spectrum of an operator on Banach spaces [3]. The integral of the resolvent defines a projection onto the related eigenspace. We study the resolvent of the light transport operator and discuss the obstacles that prevent designing any resolvent–based algorithm to extract eigenvalues.

1.2.2 Methods

To design the methods for obtaining eigenvalues, we seek to avoid discretization. Therefore, it is possible to generalize the methods to non–Lambertian scenes. However, for each method, it is required to check that the necessary (and theoretical) conditions hold for that specific method, which is not an easy task. Therefore, to simplify the task, we design and validate our methods for Lambertian scenes.
Searching for eigenvalues of an operator is a well–known problem. This task becomes more challenging when the operator acts on an infinite–dimensional space, i.e. it is not a matrix. Therefore, inspired by mathematical approaches that support the generalization of the eigenvalue problem to the operators over infinite–dimensional spaces, we build algorithms that estimate the dominant eigenvalues using Monte Carlo approaches.

Fredholm theory. Fredholm theory studies Fredholm integral equations. The existence and the uniqueness of the solution of these integral operators are equivalent to obtaining a non–zero values for the so–called Fredholm determinant. Depending on the characteristics of the kernel and the integration domain, several approaches are offered by mathematicians. The formalism of the Fredholm determinant is also generalized to some specific class of compact operators, such as Hilbert–Schmidt and trace class operators.
Considering the light transport operator for Lambertian scenes, we are able to compute the regularized determinant, which is the extension of the Fredholm determinant to the class of Hilbert–Schmidt operators, for a specific geometry (i.e., a geometry without edges).

Regression method. Inspired by regression methods, we estimate the characteristic polynomial of the light transport operator. Because of the sensitivity of extracting roots of polynomials to the coefficients, we repeat the process and obtain the dominant eigenvalues from histograms.

Power iteration method. Similar to the matrix, under some specific assumptions on the spectrum of the operator, it is possible to reach the first dominant eigenvalue by acting the operator on the output vector repeatedly.

1.3 Thesis organization

Chapter 3 reviews the basics of the transport theory, and includes the mathematical formulation of the operators included in the rendering equations. Based

[3] A Banach space is a complete normed vector space.

on the operator formulation of light transport, introduced by Arvo [Arv95b], in Chapter 4, we first review the functional properties of this operator, which were also studied by Arvo [Arv95b] and Veach [Vea98]. In addition to these primary properties, we discuss the compactness property of this operator. In Chapter 5, we discuss the resolvent of the light transport operator and demonstrate how the method is dependent on extra information about the eigenvalues and the set of eigenvectors. In Chapter 6, we perform the power iteration method by Metropolis approach to extract the dominant eigenvalue of the light transport operator. Chapter 7 represents specific coefficients that all participate to calculate the regularized determinant of the light transport operator. In Chapter 8, we build histograms of the roots of a polynomial, which is constructed from the data provided by path tracing.

Chapter 2

Related work

The low–dimensionality of the light transport operator has been widely studied (e.g. Mahajan et al. [MSRB07] analyzed the dependency of a local light transport on the block and cluster size) and used in many global illumination algorithms. The class of Precomputed Radiance Transfer (PRT) methods is a typical example of the application of this property. However, there are only a few studies on the spectrum of the light transport operator. Baranoski et al. [BBR97] illustrated that the eigenvalues of a finite rank approximation of $I - T$ are real and positive for Lambertian scenes. Ashdown et al. [Ash01] interprets the eigenvectors of a radiosity matrix as the eigenvectors of a conductance matrix.
Building on the continuous representation of light transport (operator formalism), which was introduced by Arvo [Arv95b], and similar to the functional study of this operator by Veach [Vea98], we start reviewing the very basic functional properties of the operator. Next, we go further to analyze *the compactness property* of this operator.
In Section 2.1, we address a few references used for the functional study of the light transport operator. Section 2.2 lists some applications of low–dimensionality of the local light transport. In Section 2.3, we review a few applications based on *sparseness property* of the light transport operator. Existence of this property permits exploring the structure of the transport matrix using a small number of rows and columns.

2.1 Spectrum of operators on infinite–dimensional vector spaces

Spectral theory refers to a wide range of studies regarding the eigenpairs of an operator which may not be necessarily a matrix. Therefore, many studies have been trying to find situations in which the spectral properties of the operator are known. Most of these theorems are developed based on essential properties, such as boundedness (otherwise, at least to be a closed operator), compactness

and self–adjoint (or at least a normal operator).
Unfortunately, the light transport operator does not have these *nice* properties (e.g. as a kernel operator, its kernel is not continuous or even bounded, it is neither compact nor normal). These facts about the light transport operator limit the previous work studies on operators we can exploit. However, we still benefit from the boundedness and the (local) low–dimensionality property of this operator.

Regarding the functional properties of the light transport operator, we rely upon the knowledge provided by Cheverry et al. [CR19] as a handbook of spectral theory to build possible foundations of our methods. Dealing with a non self–adjoint operator, we also refer to the book by Gohberg et al. [GK78] to study the Schmidt representation of non self–adjoint operators.
The spectral study of an operator is connected to the resolvent of that operator, see Kato et al.'s book [Kat13] for the resolvent theory of a finite rank operator and Dimov et al. [DAK01] for the application of the resolvent to the eigenvalue problem.
Resolvent theory is highly related to Fredholm theory [1]. Zemyan et al. [Zem12] provided several algorithms to compute *the Fredholm determinant* of Fredholm integral equations over $\mathbb{R}^2$. However, Bornemann et al. [Bor10] addressed the numerical methods of computing infinite Fredholm determinants (i.e. extension from matrices to compact operators).

2.2 Dimensionality reduction

Mahajan et al. [MSRB07] provided the very first paper to analyze the dimensionality of light transport versus local patch/block sizes. They demonstrated that light transport is locally low–dimensional concluded from eigenvalues of the covariance matrix. Inspired by this fact, several approaches contained in the category of the PRT methods, have been developed.

Precomputed radiance transfer (PRT).

It is referred to a class of algorithms supporting complex and expensive illumination effects by precomputing the transport matrix. However, these early methods assumed that a static scene is lit by infinitely far away light sources, i.e. environment maps.
Ramamoorthi et al. [RH01] illustrated that the unshadowed irradiance can be represented with low-order Spherical Harmonics (SH). As in the original paper of the PRT methods, Sloan et al. [SKS02] introduced a real–time approach for shading diffuse and glossy objects lit by environment maps. Focusing on low–frequency illumination, they used spherical harmonics to represent both incident radiance and transfer functions. They showed that a few element vectors (9–25) suffice to support lighting effects of diffuse objects. Next, Ng et al. [NRH03]

[1] This field includes theoretical and practical algorithms to calculate Fredholm determinant.

extended this precomputation idea to all–frequency environments by applying an orthonormal basis (Haar basis) over a cube-map domain.

Clustered principal component analysis (CPCA) for rendering.

PRT methods record the transfer matrix over many surface points which forms a high–dimensional signal over a set of points. Sloan et al. [SHHS03] presented a method to compress the storage by approximating this surface signal with an affine subspace, provided by principal component analysis (PCA). Liu et al. [LSSS04] applied the BRDF factorization to CPCA method to support all–frequency shadowing effects of glossy materials that yields a great reduction in the number of rows of transfer matrices. These approaches propose a way for dimensionality reduction over recorded transfer functions, but Nowrouzezahrai et al. [NSKF07] introduced a new approach, that is building a new basis directly from the samples of transfer functions. This basis consists of eigenvectors of the transfer function samples and these basis elements are applicable for all–frequency diffuse relighting issues.

Spectral mesh basis in PRT.

Wang et al. [WZH07] presented a new method for real–time rendering of multi–bounce indirect lighting. The key idea of their approach is using a new basis, spectral mesh basis, to approximate the surface signal, which is defined over the mesh. Namely, these basis elements are actually the eigenvectors of a matrix, mesh Laplacian, and provided by singular value decomposition (SVD) of the matrix. To reduce the SVD cost, the mesh is divided into submeshes and the basis is exploited for each submesh.

Modular radiance transfer (MRT).

Extending PRT algorithms to local lighting [KAMJ05] increases the amount of recorded data. Based on direct–to–indirect transfer ([WZH07] and [HPB06]), Loos et al. [LAM+11] introduced a novel PRT method, in which the scene dependency is eliminated and the basis is more appropriate for direct and indirect lighting tasks. In details, the solution of the transport operator is approximated using SVD decomposition for the direct–to–indirect map over simple shapes, called shape dictionaries. Then, the full transport solution is estimated by combining precomputed values of shape dictionaries at run–time.

2.3 Matrix reconstruction from sparse sampling

It has been observed that the transport matrix is sparse [NRH03]. Namely, it is possible to recover the structure of this matrix using a small number of rows and columns.

Many–light problems.

Matrix row–column sampling algorithms search for a solution to the problem of shading sampled surface points lit by a number of light sources. Hašan et al. [HPB07] offered a GPU–based method for many–lights problem. The authors developed this algorithm based on the key observation that the transport matrix is close to low rank. This fact is concluded from the decreasing order of singular values of a large matrix (containing all samples interactions with an individual light in columns and shaded samples by all lights in rows), and also based on the fact that the illumination of the scene can be represented linearly in terms of other light sources.

Nÿstrom kernel method.

Wang et al. [WDT$^+$09] applied Nÿstrom method to reconstruct the transport matrix from a small number of images from a static scene. This method, which works well with complex illuminations and occlusions, can be used for image–based relighting applications. This approach is originally inspired by Nÿstrom method [WS00] and used for reconstructing a low–rank matrix from a small number of rows and columns. Applying this method to light transport matrix enables full reconstruction of the matrix from the known rows and columns.

Light-Slice approach.

Ou et al. [OP11] introduced *the Light-Slice algorithm* to explore both local and global structure of the transport matrix. The light-slice algorithm solves the many–lights problem by sampling matrix slices [2]. This method works better than similar algorithms, such as *light-cuts* and *mrcs*, for large and complex environments.

Sparse sampling.

Huang et al. [HR10] showed that the cost of precomputation step can be reduced by adaptive and sparse sampling. In their method, a subset of vertices, called *dense vertices*, are selected for which more angular dimensions are sampled. Then, for the remaining vertices, called *sparse vertices*, a few angular samples are used.

[2]These slices are similar (clustered) image pixels, determined based on their geometry.

Chapter 3

The physics of light transport

The aim of rendering is to produce images with a correct distribution of light in all point locations in the environment. In practice, it means to simulating the propagation of light into the environment. Dedicated algorithms require a deep insight into the physical properties of light, materials, and the interactions between these two. In particular, we are able to describe phenomena such as scattering, reflection, refraction, and absorption, which are physical descriptions of light interactions with the environment.
A rendering algorithm forms an image given a full description of objects, light sources, material properties, the shape of objects, and the camera. Therefore, it is sufficient to characterize the related physical quantities. These quantities reformulate the behavior of light and the environment that leads to designing a model. In this chapter, we review key radiometric quantities that build the foundation of rendering algorithms in computer graphics.

3.1 A touch of optics

Light is electromagnetic radiation. Electromagnetic radiation can be represented as a wave, which then could be characterized by a wavelength. The wavelength of radiation varies from radio waves up to gamma–ray.
There are different sub-fields of optics such as ray optics, wave optics, and photon optics. Ray optics is the most useful area of optics in computer graphics. Ray optics (or geometrical optics) describes the macroscopic properties of light. This is helpful to get insights into the basic laws, but not sufficient by itself. Wave optics (or physical optics) studies the key concepts of optics for developing technologies. Finally, it is photon optics (or quantum optics) that describes the interaction of light with objects at atomic scales.

3.2 Radiometry

In this section, we address the most important quantities, very well-known in computer graphics. Radiometry deals with measuring electromagnetic energy. Considering the energy of light, it is possible to measure the amount of energy for an arbitrary wavelength.
In comparison to radiometry, photometry measures the visual sensation of the light spectrum viewed by the eye. The human eye is sensitive to a specific range of waves, which is between 330 nm and 770 nm. It is worth mentioning that different light sources with different wavelengths appear to have different colors. It was Pierre Bouguer who established photometry in 1760. He asked observers to compare different light sources. He concluded that the human eye is not capable of recognizing the amount of brightness while observing different light sources, however, it could detect whether two sources are equally bright [Mid19]. It is now known that different spectra have different brightness to the human eye, and that is because of the differences in the responses received by our photo–receptors from different wavelengths.
In computer graphics and image synthesis, radiometric quantities are used. These quantities can be computed from radiometric measurements.
Radiant power or flux. The radiant power, also called flux, indicates the amount of energy flowing through a surface per unit of time. This quantity is usually denoted by ϕ and is expressed in watts (joules/sec).
Radiance. Radiance is the most important quantity in image synthesis. Considering global illumination problems, this is the quantity characterizing the *appearance* of objects in the scene. It is defined as power per unit projected area per unit solid angle (watts/(steradian.m^2)), denoted by $L(\mathbf{x}, \omega)$. That is,

$$L(\mathbf{x}, \omega) = \frac{d^2\phi}{\cos\theta dS_{\mathbf{x}} d\omega}, \tag{3.1}$$

see Figure 3.3.
Note that radiance is a five–dimensional quantity varying with position, $\mathbf{x}$, and direction ω.

Remark 3.2.1 *Note that in equation* (3.1) *we used the same notation* ω *for the direction and the solid angle, which are not exactly the same. But, since the direction is actually contained in a differential solid angle around* ω*, without loss of generality, we keep using the same notation in this dissertation.*

Here, we briefly recall two properties of radiance:

1. Radiance is constant along a ray, under the assumption of no loss due to absorption or scattering. One may conclude this property from the conservation law of energy.

2. The response of a sensor contained in a display device is proportional to the radiance received by the sensor.

Irradiance. Irradiance is defined as the incident radiant power per unit surface area (watts/m^2)

$$E = \frac{d\phi}{dS}.$$

One may formulate

$$d\phi = [\int_{\Omega_{\mathbf{x}}} L(\mathbf{x}, \omega) \cos\theta d\omega] dS_{\mathbf{x}} \tag{3.2}$$

where $\cos\theta d\omega$ is often called *projected solid angle* [1], in which θ indicates the angle between the differential solid angle and the normal at surface point $\mathbf{x}$, Figure 3.3. Therefore, using equation (3.2)

$$E(\mathbf{x}) = \int_{\Omega_{\mathbf{x}}} L(\mathbf{x}, \omega) \cos\theta d\omega. \tag{3.3}$$

Radiosity. This quantity is similar to irradiance, $E(\mathbf{x})$, with the difference that radiosity refers to the amount of energy per unit area that leaves a surface. Namely,

$$B(\mathbf{x}) = \int_{\Omega_{\mathbf{x}}} L(\mathbf{x}, \omega) \cos\theta d\omega. \tag{3.4}$$

Radiosity is also known as *radiant exitance*. [2]

We limit our focus on the previous radiometric quantities.

3.2.1 Distribution functions

Reflection refers to the event in which light leaves a surface after some interaction with the surface.
A distribution function is defined from $S \times \Omega$ to $\mathbb{R}^+$ that returns the amount of reflected light at a given surface point in some direction. In Section 3.5, we will briefly review the functional properties of this space.

Remark 3.2.2 *In this dissertation, we consider both incident and outgoing light distributions in the upper hemisphere for notation simplification.*

3.2.2 The Bidirectional Reflection Distribution Function (BRDF)

The BRDF characterizes the interaction of light with materials. Consider the differential irradiance incident through $d\omega_i$. Then, the differential radiance reflected from a local outgoing direction ω_o is proportional to $dE(\mathbf{x}, \omega_i)$, Figure 3.1. Namely,

$$dL(\mathbf{x}, \omega_o) \propto dE(\mathbf{x}, \omega_i) \tag{3.5}$$

As a conclusion of equation (3.5), increasing incident light energy per unit area increases the reflected light. It is also worth mentioning that it is possible

[1] Interpreted as the projected differential area of $\Omega_{\mathbf{x}}$ onto the base of it.
[2] In photometry it is called *luminosity*.

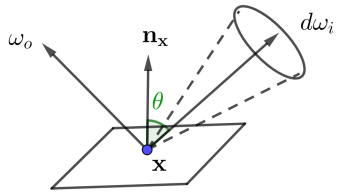


Figure 3.1: Bidirectional Reflection Distribution Function (BRDF).

to increase the incident radiance by increasing the solid angle subtended by a source.
The BRDF $\rho(\mathbf{x}, \omega_o, \omega_i)$ is defined as the constant of proportionality of equation (3.5):

$$\rho(\mathbf{x}, \omega_o, \omega_i) = \frac{L(\mathbf{x}, \omega_o)}{L(\mathbf{x}, \omega_i) \cos\theta d\omega_i}. \tag{3.6}$$

The BRDF is a four–dimensional function, one may denote it by $\rho(\theta_i, \phi_i; \theta_o, \phi_o)$, where θ_i (θ_o) and ϕ_i (ϕ_o) are polar angle and azimuthal angle respectively.
From physical aspect, the BRDF is actually a mathematical description of light interaction with different materials. This characterization is needed to compute the appearance of materials correctly.
In the general case, due to the subsurface scattering, light can leave a surface at a point that differs from the incident point. The function dealing with this phenomenon is called the *Bidirectional Surface Scattering Reflectance Distribution Function (BSSRDF)* [NRH+77]. In this dissertation, we do not consider the scattering phenomenon. We just focus on the BRDF [3].

Some properties of the BRDF.

1. Range. The BRDF is always positive.

2. Reciprocity. The value of the BRDF function does not change with reversing directions:
$$\rho(\mathbf{x}, \omega_o, \omega_i) = \rho(\mathbf{x}, \omega_i, \omega_o),$$
this property is also called *Helmholtz reciprocity.*

3. Anisotropy. The BRDF is anisotropic in general. Namely, the value of this function varies by rotating a given surface, without changing ω_o and ω_i. This phenomenon can easily be observed on brushed metals or some fabrics in which the distribution of normals at the microscale is not uniform.

[3] Note that our study does not include transparent surfaces. In that case, one may use the *Bidirectional Scattering Distribution Function (BSDF).*

4. Energy conservation. The law of conservation ensures that the total amount of reflected light in all directions is less than or equal to the total amount of light power incident on the surface.
   The necessary and sufficient condition to fulfill the conservation law is

$$\forall \omega_i \quad \int_{\Omega_{\mathbf{x}}} \rho(\mathbf{x}, \omega_o, \omega_i) \cos\theta d\omega_o \leq 1,$$

   see Dutre et al.'s book [DBB18].

To design models that characterize the BRDF accurately, it is important to ensure that energy conservation and Helmholtz reciprocity properties are satisfied. More specifically, bidirectional global algorithms compute images by tracing paths from both light sources and cameras. So, using both types of paths is feasible under the assumption of reversibility of paths, i.e. Helmholtz property.

3.2.3 BRDF examples

Depending on the material, the appearance of objects changes. One may characterize the appearance of a surface either as diffuse or glossy. In this section, we introduce some common BRDF examples.

Diffuse (Lambertian) reflection. For a diffuse surface, the BRDF value is constant in any outgoing direction. This is because of the uniform reflection over the entire hemisphere. More precisely,

$$\rho(\mathbf{x}, \omega_o, \omega_i) = \frac{\rho_d}{\pi}, \tag{3.7}$$

where ρ_d is the fraction of the total incident energy reflected from a surface , called *albedo* and belongs to $[0, 1]$. Factor π in the denominator compensates for the integral of the cosine over outgoing directions.

Specular reflection. An ideal specular surface reflects the incident light in a specific direction, which makes an equal angle with the incident direction about the normal in a plane containing both the incident direction and the normal. It is worth mentioning here that there is no such material, ideal specular, in the real world. However, some materials may behave very similar to it, like a combination of an ideal diffuse reflection with an ideal specular that becomes *the glossy reflection.*

Glossy reflection. In practice, one may consider a BRDF model as a combination of three components: ideal specular reflection, ideal diffuse (Lambertian) reflection, and glossy reflection.

As was pointed out before, the perfect diffuse and ideal specular reflection models are just the simplest mathematical descriptions of surface behavior. Therefore, a realistic BRDF contains a component that results in more complex light interactions. This component is variously called: glossy reflection, specular, narrow, and/or directional diffuse.

In computer graphics, this component, say glossy reflection, is studied by microfacet theory. In this theory, a surface is considered as a union of little reflective facets, reflecting light perfectly. This model consists of several terms that characterize the distribution of microfacets, accounting for self-shadowing and *Fresnel reflection* [4].
The analytical models of reflection include Phong's model (Phong et al. [Pho75]) and physically based models, such as Cook-Torrance (Cook et al. [CT82] and He et al. [HTSG91]).

3.3 The rendering equation

Reflectance equation. It is known that reflectance behaves linearly, meaning that incident lights from different directions independently contribute to the resulting outgoing radiance in a given direction. Therefore, the total reflected light from a surface point can be computed as an integral over all possible incident directions:

$$L(\mathbf{x}, \omega_o) = \int_{\Omega_\mathbf{x}} \rho(\mathbf{x}, \omega_o, \omega_i) L(\mathbf{x}, \omega_i) \cos\theta d\omega_i, \tag{3.8}$$

this equation is called *the reflectance equation.*

3.3.1 Direct illumination

Having the incident light distribution and the BRDF, it is possible to compute the reflected distribution using the reflectance equation.
The most straightforward case is considering no occlusion, and supposing that all light arriving at a surface comes from a distant light source. We may then put one more assumption: all light leaving the sources will arrive at the surface. Thus, we expect no shadow.
In this case, instead of considering a whole environment, surfaces are considered individually. This model is often called *local* or *direct illumination.*
Suppose that the source is a point light. It can be shown [CWH93] that the irradiance from a single point light source $\mathbf{x_s}$ that arrives at point $\mathbf{x}$ is

$$E(\mathbf{x}) = \frac{\Phi \cos\theta}{4\pi r^2_{\mathbf{x}\mathbf{x_s}}}.$$

Now, let ω_s be the direction that connects $\mathbf{x}$ to $\mathbf{x_s}$, that is $\omega_s := \mathbf{x_s} - \mathbf{x}$. Then the radiance reaching $\mathbf{x}$ is

$$L(\mathbf{x}, \omega_i) \frac{\Phi}{4\pi r^2_{\mathbf{x}\mathbf{x_s}}} \delta(\cos\theta_i - \cos\theta_s)\delta(\cos\Phi_i - \cos\Phi_s). \tag{3.9}$$

[4] Fresnel equations are used to calculate the reflectivity and transmissivity of light when it reaches the interface between two surfaces. This reflection is called Fresnel reflection.

Therefore, substituting equation (3.9) in (3.8) gives

$$L(\mathbf{x}, \omega_o) = \frac{\Phi}{4\pi} \cos\theta_s \rho(\mathbf{x}, \omega_o, \omega_s). \tag{3.10}$$

It is also possible to model direct illumination with linear and area light sources. To read more on the subject, check [NN85] and [Ama84].

3.3.2 Indirect illumination

Indirect/Global illumination is slightly more difficult than direct illumination since in this case the entire environment is involved. To add shadows, a ray tracer is required. In a ray tracer, it is possible to test whether a point on a surface is visible from another point or not. This information is essential to form shadows. In an indirect illumination model, the radiance of a surface depends on the incident light that comes from all possible surfaces. That is the reason why the indirect illumination model is able to capture inter-reflections of the environment.
Using the property that radiance is constant along a ray, it is possible to express the incident radiance at surface point $\mathbf{x}$ from another point $\mathbf{y}$,

$$L(\mathbf{x}, \omega_i) = L(\mathbf{y}, \omega_o) V(\mathbf{x}, \mathbf{y}), \tag{3.11}$$

where $V(\mathbf{x}, \mathbf{y})$ is called *the visibility function*, returning 1 when there is no

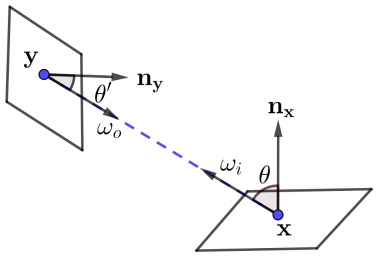


Figure 3.2: Expressing the incident radiance at point $\mathbf{x}$ in terms of radiance at point $\mathbf{y}$.

object between $\mathbf{x}$ and $\mathbf{y}$ otherwise 0, see Figure 3.2.
Considering the dependency characterized by equation 3.11, one may reformulate the equation in terms of an integral over surface points. This task can be done by a change of variable from $d\omega_i$ to $dS_\mathbf{y}$, that originates from the connection of the solid angle sustained by point $\mathbf{y}$ to its projected area

$$d\omega_i = \frac{\cos\theta'}{r^2_{\mathbf{xy}}} dS_\mathbf{y}. \tag{3.12}$$

Finally, substituting equation (3.12) into the reflectance equation leads to

$$L(\mathbf{x}, \omega_o) = \int_S \rho(\mathbf{x}, \omega_o, \omega_i) L(\mathbf{y}, \omega_o) \frac{\cos\theta'}{r^2_{\mathbf{xy}}} V(\mathbf{x}, \mathbf{y}) \cos\theta dS_{\mathbf{y}}. \tag{3.13}$$

Remark 3.3.1 *In the equation above, we use the same notation ω_o associated to both points $\mathbf{x}$ and $\mathbf{y}$, indicating the local outgoing directions in the local coordinate system centered at each point. However, these two directions are not necessarily the same.*

The equation above is known as *the rendering equation*. To complete the rendering equation to obey the energy conservation law, it is required to add one more term indicating the emitted radiance at $\mathbf{x}$, that is

$$L(\mathbf{x}, \omega_o) = \int_S \rho(\mathbf{x}, \omega_o, \omega_i) L(\mathbf{y}, \omega_o) V(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) dS_{\mathbf{y}} + L_e(\mathbf{x}, \omega_o), \tag{3.14}$$

in which

$$G(\mathbf{x}, \mathbf{y}) = \frac{\cos\theta'}{r^2_{\mathbf{xy}}} \cos\theta$$

, and $G(\mathbf{x}, \mathbf{y})$ is called *the geometric term* between points $\mathbf{x}$ and $\mathbf{y}$. It was first Kajiya [Kaj86] who used equation (3.14) as rendering equation however his derivation was slightly different from this equation.

3.3.3 The radiosity equation

The power emitted from a diffuse emitter [5] can be computed as

$$\begin{aligned} \Phi &= \int_S \int_{\Omega_{\mathbf{x}}} L(\mathbf{x}, \omega_o) \cos\theta d\omega_o dS_{\mathbf{x}} \\ &= L \int_S \int_{\Omega_{\mathbf{x}}} \cos\theta d\omega_o dS_{\mathbf{x}} \\ &= L\pi A, \end{aligned} \tag{3.15}$$

where A is the total area of this emitter. Since radiosity is defined as the exitant power per unit surface area, so

$$B(\mathbf{x}) = \pi L(\mathbf{x}). \tag{3.16}$$

Therefore, *the radiosity equation* can be derived from the rendering equation. The BRDF does not depend on the incident and exitant directions, so it can be taken out of the integral,

$$L(\mathbf{x}) = \frac{\rho_d(\mathbf{x})}{\pi} \int_S G(\mathbf{x}, \mathbf{y}) V(\mathbf{x}, \mathbf{y}) L(\mathbf{y}) dS_{\mathbf{y}} + L_e(\mathbf{x}). \tag{3.17}$$

Multiplying equation (3.17) by π yields,

$$B(\mathbf{x}) = \rho_d(\mathbf{x}) \int_S G(\mathbf{x}, \mathbf{y}) V(\mathbf{x}, \mathbf{y}) \frac{B(\mathbf{y})}{\pi} dS_{\mathbf{y}} + B_e(\mathbf{x}). \tag{3.18}$$

[5] A diffuse emitter emits equal light in all directions and points.

Discretization of radiosity equation.

The radiosity methods belong to the class of Galerkin finite element methods. They approximate the solution of the rendering equation by a function that belongs to a finite–dimensional vector space. The simplest of these methods uses the space of piecewise constant functions over a tessellation of the scene.
In the classic radiosity equation, the goal is to compute the average radiosity, let's say B_i on each patch [6] s_i. The self-emitted radiosity B_i^e and ρ_i, the reflectivity of patch s_i, are the inputs of the algorithm. The algorithm returns B_i associated to each patch.
Mathematically speaking, a system of equations formed by inputs is constructed, in which the solution is the pixel values.
The average radiosity emitted from patch s_i is

$$B_i = \frac{1}{A_i}\int_{s_i}\int_{\Omega_{\mathbf{x}}} L(\mathbf{x})\cos\theta d\omega_o dS_{\mathbf{x}}. \tag{3.19}$$

Since the environment is diffuse, $L(\mathbf{x})$ can be taken out of the inner integral,

$$\begin{aligned} B_i &= \frac{1}{A_i}\int_{s_i} L(\mathbf{x})(\int_{\Omega_{\mathbf{x}}}\cos\theta d\omega_o)dS_{\mathbf{x}} \\ &= \frac{1}{A_i}\int_{s_i} L(\mathbf{x})\pi dS_{\mathbf{x}} \\ &= \frac{1}{A_i}\int_{s_i} B(\mathbf{x})dS_{\mathbf{x}}. \end{aligned} \tag{3.20}$$

The radiosity equation is an integral equation. To solve such equations, the problem is usually reduced to an approximation problem. That is to say, the radiosity across an element (patch) is approximated by a set of *basis functions*, also called *shape functions* [ZTNZ77].
In the following, we will derive the so–called classical *radiosity system of equations* starting with the radiosity integral equation.
Integrating both sides of equation (3.18) yields

$$\frac{1}{A_i}\int_{s_i} B(\mathbf{x})dS_{\mathbf{x}} = \frac{1}{A_i}\int_{s_i}\int_S \rho_d(\mathbf{x})G(\mathbf{x},\mathbf{y})V(\mathbf{x},\mathbf{y})\frac{B(\mathbf{y})}{\pi}dS_{\mathbf{y}}dS_{\mathbf{x}} + \frac{1}{A_i}\int_{s_i} B_e(\mathbf{x})dS_{\mathbf{x}}, \tag{3.21}$$

which is equivalent to

$$\frac{1}{A_i}\int_{s_i} B(\mathbf{x})dS_{\mathbf{x}} = \frac{1}{A_i}\int_{s_i}\sum_j\int_{s_j} \rho_d(\mathbf{x})G(\mathbf{x},\mathbf{y})V(\mathbf{x},\mathbf{y})\frac{B(\mathbf{y})}{\pi}dS_{\mathbf{y}}dS_{\mathbf{x}} + \frac{1}{A_i}\int_{s_i} B_e(\mathbf{x})dS_{\mathbf{x}}. \tag{3.22}$$

Now, under the assumption that the radiosity across each patch is constant, which is very unlikely to happen in the real world, equation (3.22) becomes

$$B_i' = \sum_j B_j'\frac{1}{A_i}\int_{s_i}\int_{s_j}\rho_d(\mathbf{x})\frac{G(\mathbf{x},\mathbf{y})V(\mathbf{x},\mathbf{y})}{\pi}dS_{\mathbf{y}}dS_{\mathbf{x}} + B_i^e, \tag{3.23}$$

[6] These patches are often triangles or convex quadrilaterals.

where B'_i is a constant value of radiosity across patch s_i. Now, if we suppose that $\rho_d(\mathbf{x})$ is also constant over each patch, denoted by ρ_i for patch i, then equation (3.23) becomes

$$B'_i = \rho_i \sum_j F_{ij} B'_j + B^e_i. \tag{3.24}$$

In equation (3.24), F_{ij} coefficients are called *form factors*:

$$F_{ij} = \frac{1}{A_i} \int_{s_i} \int_{s_j} \frac{G(\mathbf{x}, \mathbf{y}) V(\mathbf{x}, \mathbf{y})}{\pi} dS_{\mathbf{y}} dS_{\mathbf{x}}. \tag{3.25}$$

To derive an equation closer to the real case, using $B_i + \epsilon_B(\mathbf{x})$ in equation (3.22) turns equation (3.23) to the following

$$B_i = \rho_i \sum_j F_{ij} B_j + B^e_i + M, \tag{3.26}$$

where M denotes the error term of the approximation.
To read more on the classic radiosity method, refer to Gloral et al. [GTGB84], Cohen et al. [CG85] and Nishita et al. [NN85].
Radiosity methods have developed more and more ever since. New classes of radiosity algorithms are based on sampling strategies. For example, one branch of such algorithm is called *stochastic relaxation*. These methods solve linear systems of equations using iterative algorithms.
The solution of a linear system may be achieved using Monte Carlo algorithms, see Forsythe et al. [FL50]. This new category of methods is called *discrete random walk radiosity*.
It is worth mentioning that there exists another class of methods using Monte Carlo approaches, in which the integral of the rendering equation is solved directly. This procedure does not rely on the assumption of piecewise constant reflectivity and works with every kind of reflectance model.
One may summarize the classic radiosity method in the following steps:

1. Tessellating the scene geometry, i.e. discretizing the scene into patches.
2. Computing form factors using equation (3.25).
3. Solving the radiosity system of equations, equation (3.24).
4. Turning the resulting solution into display colors by tone mapping.

Despite the fact that radiosity is an object–based method, there are major problems such as discretization and form factor computation. In the discretization step, the size of the patches is important. It should be small enough to have an acceptable estimate of the illumination of the surfaces. On the other hand, the number of patches should be taken into consideration. The number of patches should not exceed the storage capacity. For this problem, hierarchical refinement and clustering methods are proposed. The other challenge is computation

of form factors. Calculation of form factors is an expensive task since the cost grows quadratically with the number of patches.
Considering the error rising in equation (3.26), one may consider the radiosity method as an unreliable approach. In Chapter 4, we show the connection of this error bound with the compactness property of light transport. Namely, when the operator is compact, this error can be made arbitrarily small independent of the solution. However, we will see that this operator is not compact in general.

3.4 Monte Carlo Methods

Monte Carlo methods are techniques using sampling strategies to estimate integrals. These methods were first developed to simulate neutron transport by researchers like Stanislaw Ulam (1946), John von Neumann (1947), and Nicholas Metropolis (1949), while developing nuclear weapons. However, random sampling had been used earlier in an experiment conducted by Comte de Buffon (1777), to compute a specific probability. Later, it was Laplace who expressed that this repetitive experiment could be used to estimate the value of π.
The Monte Carlo approach is used to compute the value of an integration, which is provided by the expected value of a random variable.
One major advantage of Monte Carlo methods is the simplicity that can be applied to a large variety of stochastic problems, such as transport problems. However, one may also mention a big disadvantage of Monte Carlo techniques as their convergence rate $\frac{1}{\sqrt{N}}$ with N number of samples. This problem demands a search for optimization approaches, or variance reduction techniques. Despite all efforts until today, Monte Carlo techniques are still slow. However, for some specific problems, such as computing high–dimensional integrals or integrals with non smooth integrand, these techniques are the only practicable methods. For more details on Monte Carlo methods, refer to Kalos and Whitlock [KW09], Hammersley and Handscomb [HH64], Spanier and Gelbard [SG08]. For quasi–Monte Carlo methods, see Niederreiter [Nie92].

3.4.1 Monte Carlo integration

Let $f : \Omega \longrightarrow \mathbb{R}$ be a real–valued function defined on a measure space. Consider the following integral,

$$I = \int_{\Omega} f(x) d\mu, \tag{3.27}$$

where μ is the corresponding measure.
The goal is to estimate I. To reach this goal using the Monte Carlo method, it is required to draw samples in Ω according to a probability distribution function $p(x)$. Then, the following expression is an estimator of I using N samples

$$< I >:= \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}. \tag{3.28}$$

Using linearity property of expectation, it is easy to show that

$$E(< I >) = I.$$

The variance of this estimator is

$$\sigma^2 = \frac{1}{N} \int_{\Omega} (\frac{f(x)}{p(x)} - I)^2 p(x) d\mu. \tag{3.29}$$

Equation (3.29) indicates a dependency on N. That is, variance decreases as N increases. It actually indicates that the error of the estimation, which is proportional to variance, is decreasing by using four times more samples.

3.4.2 Bias

The expected value of the estimation error, denoted by $B[< I >]$, is defined as the difference between the expectation of $< I >$ and the actual value, i.e.

$$B[< I >] := E[< I >] - I.$$

This quantity is called *Bias* [7]. If $B[< I >] = 0$, then the estimator is called *unbiased*, otherwise *biased*.
This notation is very important in classifying Monte Carlo approaches. Using an unbiased estimator, we are more or less able to control the error. Despite the essential accuracy offered by an unbiased algorithm, these methods are expensive in practice.

3.4.3 Monte Carlo steps

A Monte Carlo algorithm includes the following steps:

1. Drawing samples from a probability distribution.
2. Evaluating the function at sample points.
3. Computing the estimator using resulting values of the previous step and equation (3.28).

3.5 Operator formulation

Ray space.

The four–dimensional *ray space* $\ell = S \times \Omega$ is the Cartesian product of the set of surface points and the unit outgoing directions, defined in the upper hemisphere centered at each surface point.

[7] Bias does not appear as noise in the image. It is often detectable in the form of blurring or incorrect surface shading.

Space of light distributions.

Definition 3.5.1 (Hilbert space) *A vector space which is a complete metric space with respect to the metric induced by the inner product in the vector space.*

Let $\mathbb{H}$ be the space of linear real-valued light distribution functions defined from ℓ to $\mathbb{R}^+$. One may define an inner product on $\mathbb{H}$ as follows

$$< L_1, L_2 >= \int_S \int_{\Omega_{\mathbf{x}}} L_1(\mathbf{x}, \omega) L_2(\mathbf{x}, \omega) \cos\theta dS_{\mathbf{x}} d\omega, \tag{3.30}$$

where θ indicates the angle between the normal at point $\mathbf{x}$ and the outgoing direction ω, see Figure 3.3. Veach [Vea98] showed that the induced norm by this inner product is complete. We denote this norm by $\|\cdot\|_{\mathbb{H}}$.

Measure on $\mathbb{H}$.

Definition 3.5.2 (Measure space) *A Measure space is a triple $(S, \mathcal{S}, \mu)$, where S is a set, $\mathcal{S}$ is a σ–algebra [8] of the subsets of S, and $\mu : \mathcal{S} \to [0, \infty]$ is a measure.*

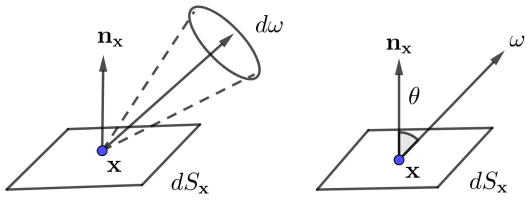


Figure 3.3: The projected area measure is defined by $dS^{\perp}_{\mathbf{x}} = \cos\theta\, dS_{\mathbf{x}}$.

Veach [Vea98] defined a measure on the ray space, called *throughput measure* and denoted by μ. For any given ray $(\mathbf{x}, \omega) \in \ell$

$$d\mu(\mathbf{x}, \omega) = dS^{\perp}_{\mathbf{x}} d\omega,$$

where $dS^{\perp}_{\mathbf{x}}$ is the projected area measure, see Figure 3.3.

The local reflectance operator.

Definition 3.5.3 *We define the local reflectance operator $K_{\mathbf{x}} : L(\Omega_{\mathbf{x}}) \longrightarrow L(\Omega_{\mathbf{x}})$ as the following integral operator*

$$(K_{\mathbf{x}} L_i)(\omega_o) = \int_{\Omega_{\mathbf{x}}} \rho(\mathbf{x}, \omega_o, \omega_i) L_i(\mathbf{x}, \omega_i) \cos\theta d\omega_i, \tag{3.31}$$

[8] A σ–algebra on a set S is a collection of S's subsets which is closed under countable unions, countable intersections and complement.

acting on an incident distribution, arriving at point $\mathbf{x}$*, and resulting another distribution,* $K_{\mathbf{x}}L_i = L(\omega_o)$ *at point* $\mathbf{x}$*.*

Note that this operator is defined locally, which means we are interested to interpret the interaction of light at fixed point $\mathbf{x}$.

The global reflectance operator.

Definition 3.5.4 *Similar to Definition 3.5.3, this operator transforms an incident distribution to another one. However, this operator, denoted by* K*, is not restricted to a fixed point and defined from* $\mathbb{H}$ *to itself. Therefore, these two operators are connected with the following equation*

$$K_{\mathbf{x}}L(\omega) = KL(\mathbf{x}, \omega).$$

The reparametrization operator.

Before defining the reparametrization operator, it is required to define *the ray–casting function.* Similar to Veach's definition [Vea98], we define $p : \ell \longrightarrow S$ such that

$$p(\mathbf{x}, \omega_i) = \mathbf{x} + t_{int}\,\omega_i,$$

where

$$t_{int} = \min\{t \mid \mathbf{x} + t\,\omega_i \in S, t \in (0, \infty)\}.$$

In practice, this function indicates the closest visible point to $\mathbf{x}$ in a global direction corresponding to local direction ω_i.

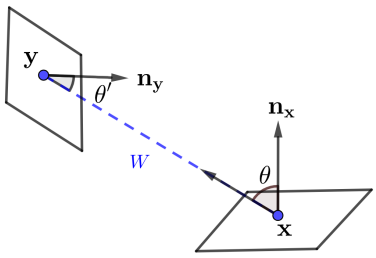


Figure 3.4: Reparametrization operator: representing the incoming radiance at point $\mathbf{x}$ in terms of the outgoing radiance from the nearest point.

Definition 3.5.5 *Arvo [Arv95b] defined an operator* $\mathbf{G}$ *to represent an incoming radiance at a point as transported radiance reflected from another (the nearest point), i.e.*

$$(\mathbf{G}L)(\mathbf{x},\omega_i) = L(\mathbf{y} = p(\mathbf{x},\omega_i),\omega_o) = L_i(\mathbf{x},\omega_i), \tag{3.32}$$

where ω_i *and* ω_o *are the results of localizing* W *at point* $\mathbf{x}$ *and* $-W$ *at point* $\mathbf{y}$ *in which* $W = \mathbf{y} - \mathbf{x}$*, see Figure 3.4.*

Note that this definition implies that the output distribution after action of $\mathbf{G}$ will be the same distribution, but at a different ray. That is why we call it *the reparametrization operator.*

The light transport operator.

Definition 3.5.6 *The light transport operator* $T : \mathbb{H} \to \mathbb{H}$ *is defined as composition of* K *and* $\mathbf{G}$ *i.e.* $T = K\mathbf{G}$*, that is*

$$\forall(\mathbf{x},\omega_o) \in \ell : (TL)(\mathbf{x},\omega_o) = \int_{\Omega_\mathbf{x}} \rho(\mathbf{x},\omega_o,\omega_i) L_i(\mathbf{x},\omega_i) \cos\theta d\omega_i. \tag{3.33}$$

Equation (3.33) can be reformulated in terms of point surfaces instead of solid angle i.e.

$$(TL)(\mathbf{x},\omega_o) = \int_S \kappa(\mathbf{x},\mathbf{y}) L(\mathbf{y},\omega_o) dS_\mathbf{y} \tag{3.34}$$

with

$$\kappa(\mathbf{x},\mathbf{y}) = \rho(\mathbf{x},\omega_o,\omega_i) \frac{\cos\theta\cos\theta'}{r^2_{\mathbf{xy}}} V(\mathbf{x},\mathbf{y}), \tag{3.35}$$

where $\frac{\cos\theta'}{r^2_{\mathbf{xy}}}$ is the Jacobian of change of variables from $d\omega_i$ to $dS_\mathbf{y}$, and $V(\cdot,\cdot)$ is the visibility function.

The radiant exitance transport operator.

Definition 3.5.7 *Let* $\mathcal{B}$ *denotes the radiant exitance field which is defined by equation* (3.4)*. For this specific case, the purely diffuse material, the integral of equation* (3.35) *simplifies to* $T_b : \mathcal{B} \longrightarrow \mathcal{B}$

$$(T_b B)(\mathbf{x}) = \int_S \kappa(\mathbf{x},\mathbf{y}) B(\mathbf{y}) dS_\mathbf{y} \tag{3.36}$$

with

$$\kappa(\mathbf{x},\mathbf{y}) = \rho_d(\mathbf{x}) \frac{\cos\theta\cos\theta'}{\pi r^2_{\mathbf{xy}}} V(\mathbf{x},\mathbf{y}). \tag{3.37}$$

We call this operator the radiant exitance transport operator.

Part I

On the functional study of the light transport operator

Chapter 4

Functional properties of the light transport operator

In this chapter, first, we review the basic functional properties of the light transport operator in Sections 4.1, 4.2, and 4.3. For the completeness of the context, some of the known proofs and extra information on the light transport operator in a very specific situation are delivered in the appendix of the chapter. In addition to these basic properties, we also discuss the *compactness* property of the light transport operator, Section 4.4. We finish the chapter by discussing the impact of this property on bounding the approximation error of algorithms, developed based on the discretized transport operator (matrices), and the difficulty of computing the eigenvalues of this operator in the absence of this property.

4.1 Boundedness

Definition 4.1.1 *Suppose* $(X, \|.\|_X)$ *and* $(Y, \|.\|_Y)$ *are normed spaces and* $T : (X, \|.\|_X) \to (Y, \|.\|_Y)$ *is linear.* T *is said to be bounded if*

$$\exists M \quad \forall x \in X \quad \|T(x)\|_Y \leq M.$$

Notation. We denote the space of bounded linear operators defined from a Hilbert space $\mathbb{H}$ to itself by $\mathcal{B}(\mathbb{H})$.

Definition 4.1.2 (Operator norm) *Let* $(X, \|.\|_X), (Y, \|.\|_Y)$ *be two normed spaces, and* $T : (X, \|.\|_X) \longrightarrow (Y, \|.\|_Y)$ *be a linear map. Then, the operator norm is defined*

$$\|T\| = \sup_{\|L\|_X = 1} \|TL\|_Y. \tag{4.1}$$

The light transport operator is bounded. Veach [Vea98] proved that $\|K\| < 1$. It can be shown that the reparametrization operator is also bounded,

i.e. $\|\mathbf{G}\| \leq 1$ (see Section 4.A.1). Therefore, The light transport operator is also bounded

$$\|T\| = \|K\|\|\mathbf{G}\| < 1. \tag{4.2}$$

4.2 Continuity

Definition 4.2.1 (Continuous operator) *A linear map $T : (X, \|.\|_X) \to (Y, \|.\|_Y)$ from one Hilbert space to another is continuous if for any $x \in X$ and given $\epsilon > 0$, there exists $\delta > 0$ s.t.*

$$\|x - x'\|_X < \delta \qquad \|Tx - Tx'\|_Y < \epsilon.$$

Using the boundedness property, the following proposition [Rud91] implies that the light transport operator is also continuous.

Proposition 4.2.1 *Suppose $(X, \|.\|_X)$ and $(Y, \|.\|_Y)$ are normed spaces. For a linear map $T : (X, \|.\|_X) \to (Y, \|.\|_Y)$, the following statements are equivalent*

1. *T is bounded;*
2. *T is continuous at 0;*
3. *T is continuous.*

The light transport operator is continuous. Since the light transport operator is linear and bounded, the previous proposition implies its continuity.

4.3 Normality

Before bringing the definition of a normal operator, it is required to recall the definition of the adjoint operator.

Definition 4.3.1 (Adjoint operator) *Let $T \in \mathcal{B}(\mathbb{H})$. There exists a unique operator, $T^* \in \mathcal{B}(\mathbb{H})$ such that*

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{H} \quad < Tx, y >_{\mathbb{H}} = < x, T^* y >_{\mathbb{H}} .$$

Definition 4.3.2 (Self–adjoint operator) *The operator $T \in \mathcal{B}(\mathbb{H})$ is self–adjoint if $T = T^*$.*

Adjoint of G and K operators. Veach [Vea98] proved that operator $\mathbf{G}$ is self–adjoint, and K is also self–adjoint under the assumption of symmetric BSDF. In the definition of K and $K_{\mathbf{x}}$, we use the BRDF instead of the BSDF, therefore, based on Remark 3.2.2 and the Helmoltz reciprocity property of the BRDF, we conclude the same for both of these operators, Section 4.A.3.

Adjoint of the light transport operator (importance transport). The light transport operator is reformulated as a composition of two other operators, i.e. $T = K\mathbf{G}$. By the property, related to the adjoint of the composition of operators in Hilbert spaces [Rud91], we have $T^* = G^* K^* = \mathbf{G}K$, which is concluded from the self–adjointness property of K and $\mathbf{G}$.

Definition 4.3.3 (Normal operator) *The operator $T \in \mathcal{B}(\mathbb{H})$ is normal if*

$$TT^* = T^*T.$$

The light transport operator is not normal. By a counterexample, we show that the light transport operator is not normal.

Proof 4.3.1 *Consider the following reflection model (perfect specular reflection) for a scene that only contains two surfaces with different reflectance properties (Figure 4.1):*

$$\rho(\mathbf{x}, \omega_o, \omega_i) = \frac{\delta(\omega_i - \omega_o) F_r(\omega_o)}{\cos\theta},$$

where F_r indicates the fraction of light reflected. For an arbitrary L, we show

$$TT^*L \neq T^*TL.$$

We know $TT^* = K^2$ and $T^*T = \mathbf{G}K^2\mathbf{G}$. Define $L'' := K^2L$, so

$$\begin{aligned} L''(\mathbf{x}, \omega_o) &= \int_{\Omega_\mathbf{x}} \rho(\mathbf{x}, \omega_o, \omega_i) KL(\mathbf{x}, \omega_i) \cos\theta d\omega_i \\ &= \int_{\Omega_\mathbf{x}} \rho(\mathbf{x}, \omega_o, \omega_i) \Big(\int_{\Omega_\mathbf{x}} \rho(\mathbf{x}, \omega_i, \omega'') L(\mathbf{x}, \omega'') \cos\theta'' d\omega'' \Big) \cos\theta d\omega_i \\ &= F_r^2(\omega_o) L(\mathbf{x}, \omega_o). \end{aligned} \tag{4.3}$$

On the other hand,

$$\begin{aligned} T^*TL(\mathbf{x}, \omega_o) &= \mathbf{G}K^2\mathbf{G}L(\mathbf{x}, \omega_o) = \mathbf{G}K^2L(\mathbf{y}, \omega_o^\mathbf{y}) \\ &= \mathbf{G}(F_r^2(\omega_o^\mathbf{y}) L(\mathbf{y}, \omega_o^\mathbf{y})) \\ &= F_r^2(\omega_o^\mathbf{y}) \mathbf{G}L(\mathbf{y}, \omega_o^\mathbf{y}) \\ &= F_r^2(\omega_o^\mathbf{y}) L(\mathbf{x}, \omega_o). \end{aligned} \tag{4.4}$$

Since F_r has different values on the surfaces, the values of equations (4.3) *and* (4.4) *are not the same.*

4.4 Compactness

Definition 4.4.1 (Compact operator) *Let $(X, \|.\|_X), (Y, \|.\|_Y)$ be two normed spaces. A linear map $T : (X, \|.\|_X) \to (Y, \|.\|_Y)$ is called compact if for each bounded sequence $(u_n) \in X^\mathbb{N}$, (Tu_n) has a convergent subsequence.*

The following theorem expresses an important property of compact operators. We refer to this theorem frequently in this manuscript.

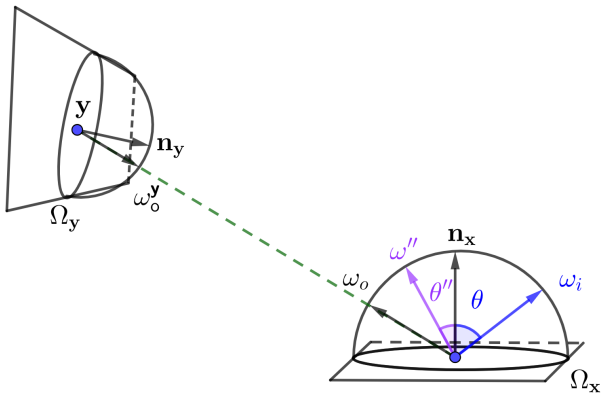


Figure 4.1: Notations used in Proof 4.3.1.

Theorem 4.4.1 (Fredholm alternative [CR19]) *Let $T : (X, \|.\|_X) \longrightarrow (X, \|.\|_X)$ be a compact map. Then, the non–zero elements of the spectrum of T are isolated with finite algebraic multiplicity, and the only possible accumulation point of the spectrum is* 0.

Theorem 4.4.2 *The light transport operator is not compact.*

Proof 4.4.1 *The integral operator defined in equation 3.33 is partial, because it is only performed over directions instead of space and directions. Therefore, this operator is not compact [KZ91].*

Theorem 4.4.2 concludes that the operator is not compact in general. However, by restricting the material to purely diffuse, we can still address the condition under which the radiant exitance transport operator is compact, Theorem 4.4.4. The following theorem provides the sufficient condition of a kernel integral operator over $\mathcal{L}_2$ functions to be compact.

Theorem 4.4.3 (Hilbert–Schmidt operator) *Let X and Y be two measure spaces with μ and σ as their corresponding measures respectively. Consider $\kappa(\mathbf{x}, \mathbf{y})$ be a measurable function on $X \times Y$ with the following property*

$$\int_{X\times Y} |\kappa(\mathbf{x}, \mathbf{y})|^2 d\mu(\mathbf{x}) d\sigma(\mathbf{y}) < \infty$$

then

$$(TL)(x) = \int_Y \kappa(\mathbf{x}, \mathbf{y})\, L(\mathbf{y}) d\sigma(\mathbf{y}),$$

is a compact map from $\mathcal{L}_2(Y, d\sigma)$ to $\mathcal{L}_2(X, d\mu)$ [CR19]. Such an operator is called Hilbert–Schmidt operator.

Theorem 4.4.4 *If the kernel of T_b is bounded over S^2, then the operator is compact.*

Proof 4.4.2 *The bounded property of the kernel implies that it is also square–integrable:*

$$\int_{S^2} |\kappa(\mathbf{x},\mathbf{y})|^2 dS_{\mathbf{x}} dS_{\mathbf{y}} \leq M^2 \int_{S^2} dS_{\mathbf{x}} dS_{\mathbf{y}} = M^2 A^2,$$

where A indicates the area of S. Theorem 4.4.3 concludes the compactness.

In addition to Theorem 4.4.4, providing a sufficient condition for the compactness of radiant exitance operator, we also illustrate that an unbounded kernel implies non–compactness of the radiant exitance operator.

Theorem 4.4.5 *If κ is not bounded, then T_b is not compact.*

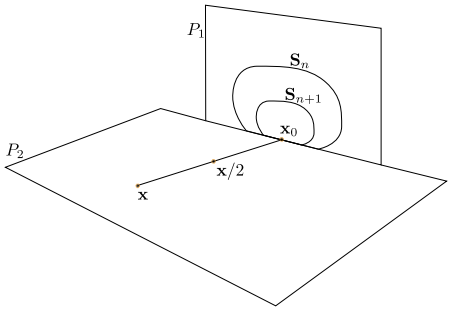


Figure 4.2: T_b is not compact on a scene with abutting edges.

Proof 4.4.3 *An arbitrary geometry is a composition of joint surfaces, sharing a common edge, and disjoint surfaces. The kernel of T_b becomes unbounded for arbitrary close points with different normals i.e. near a common edge. Therefore, a geometry with at least one edge is considered as a geometry with an unbounded kernel. In what follows, without loss of generality, we proceed with the proof for a simple scene with two orthogonally abutting half–planes.*
Let $\mathbf{x_0}$ be the common origin of both planes, lied on the common edge, and S_0 be a region of the unit area on plane P_1 (See Figure 4.2). We construct a sequence of regions in this plane by scaling S_0 by a factor of 2^n, and let s_n indicate the area of S_n. Next, we build a sequence of radiant exitance functions, B_n, as constant functions over S_n regions, i.e.

$$B_n(\mathbf{y}) = 2^n \quad \forall \mathbf{y} \in S_n$$

Each B_n has a unit $\mathcal{L}_2$ norm by construction

$$\|B_n\|_2^2 = \int_{S_n} B_n^2(\mathbf{y}) dS_{\mathbf{y}} = 4^n s_n = 1.$$

We will show that sequence (T_bB_n) *does not have any converging subsequence. Choose point* $\mathbf{x}$ *in* P_2*. Then*

$$T_bB_n(\mathbf{x}) = \rho\, 2^n F_n(\mathbf{x}),$$

where $F_n(\mathbf{x})$ *and* ρ *denote the form factor from* $\mathbf{x}$ *to* S_n *and the albedo of plane* P_1 *respectively. Since* $F_n(\mathbf{x}) = F_{n-1}(2\mathbf{x})$*, we have*

$$T_bB_n(\mathbf{x}) = \rho\, 2^n F_n(\mathbf{x}) = \rho\, 2^n F_{n-1}(2\mathbf{x}) = 2T_bB_{n-1}(2\mathbf{x}) = 2^n T_bB_0(2^n\mathbf{x}).$$

We define d_p *as the sequence of distances between* T_bB_n *and* T_bB_{n+p} *i.e.* $d_p = \|T_bB_n - T_bB_{n+p}\|_2$*. By definition:*

$$\begin{aligned} d_p^2 &= \int_S |T_bB_n(\mathbf{x}) - T_bB_{n+p}(\mathbf{x})|^2 dS_{\mathbf{x}} \\ &= \int_S |2^n T_bB_0(2^n\mathbf{x}) - 2^n T_bB_p(2^n\mathbf{x})|^2 dS_{\mathbf{x}}. \end{aligned} \tag{4.5}$$

Let $\mathbf{y} = 2^n\mathbf{x}$*. Expressing the integral of equation* (4.5) *in terms of* $\mathbf{y}$ *gives*

$$d_p^2 = \int_S |T_bB_0(\mathbf{y}) - T_bB_p(\mathbf{y})| dS_{\mathbf{y}}, \tag{4.6}$$

where $dS_{\mathbf{y}} = 4^n dS_{\mathbf{x}}$ *was used to derive the equation.*
To compute the limit of (d_p) *sequence, it is required to evaluate the integrand of equation* (4.6) *when* p *tends to infinity. We know that*

$$T_bB_p(\mathbf{y}) = 2^p\ T_bB_0(2^p\mathbf{y}) = \rho\ 2^p\ F_0(2^p\mathbf{y}).$$

Using a large value of p *for* $2^p\mathbf{y}$*, makes this point far away from* $\mathbf{x_0}$*. Therefore, the form factor* $F_0(2^p\mathbf{y})$ *is equal to the multiplication of* s_0 *and the point–to–point form factor from* $2^p\mathbf{y}$ *to* $\mathbf{x_0}$*. Orthogonality of* P_1 *and* P_2 *yields the cosine between the direction* $\mathbf{x_0} - 2^p\mathbf{y}$ *and the normal at* $\mathbf{y}$ *tends to* 0 *and the cosine between* $2^p\mathbf{y} - \mathbf{x_0}$ *and the normal at* $\mathbf{x_0}$ *tends to* 1*, therefore*

$$\lim_{p\to\infty} d_p = \|\mathbf{T}_bB_0\|_2. \tag{4.7}$$

Let $(T_bB_{q(n)})$ *be a convergent subsequence of* (T_bB_n)*, in which* $q : \mathbb{N} \longrightarrow \mathbb{N}$ *is a strictly increasing map. Convergence implies that this sequence is also Cauchy, i.e. for an arbitrary* ϵ*, there exists* δ_ϵ *s.t.*

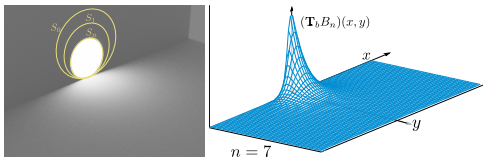
$$\forall n_1, n_2 > \delta_\epsilon \qquad \|T_bB_{q(n_1)} - T_bB_{q(n_2)}\|_2 < \epsilon \tag{4.8}$$

Let $n > \delta_\epsilon$ *then*

$$\|T_bB_{q(n)} - T_bB_{q(n+p)}\|_2 = d_{q(n+p)-q(n)} < \epsilon.$$

However, equation (4.7) *implies that the the limit of* d_p *is a non negative value, on the other hand because* q *is strictly increasing, therefore* $d_{q(n+p)-q(n)}$ *can not be arbitrary small, which contradicts equation* (4.8)*.*

We provide an experiment to illustrate the result of Theorem 4.4.5 practically. In this experiment, the sequence of S_n are disk emitters. The differences of the transported radiance to P_2 plane are given in a tabulation, see Figure 4.3.



	p = 1	p = 2	p = 3	p = 4	p = 5	p = 6	p = 7	p = 8
n = 2	0.003146	0.006452	0.009923	0.013522	0.017174	0.020789	0.024277	0.027568
n = 3	0.003371	0.006966	0.010740	0.014606	0.018462	0.022209	0.025769	0.029087
n = 4	0.003658	0.007549	0.011580	0.015638	0.019618	0.023430	0.027013	0.030325
n = 5	0.003950	0.008093	0.012310	0.016488	0.020533	0.024370	0.027950	0.031244
n = 6	0.004199	0.008527	0.012866	0.017112	0.021185	0.025026	0.028595	0.031871
n = 7	0.004385	0.008837	0.013247	0.017528	0.021612	0.025449	0.029007	0.032268
n = 8	0.004510	0.009037	0.013487	0.017785	0.021871	0.025703	0.029253	0.032504

Figure 4.3: Illustration of Theorem 4.4.5: the radiant exitance operator T_b is not compact. Left: rendered scene with diffuse emitter S_n. Center: the radiant exitance computed on the receiver plane for $n = 7$. Right: table contains the pairwise $\mathcal{L}_2$ distances between $T_b B_n$ and $T_b B_{n+p}$.

4.5 Discussion

Bounding the approximation error and compactness. A compact operator can be interpreted as the limit of a sequence of square matrices, say (T_n), Theorem 4.A.1, which is obviously independent of the choice of the initial vector. Mathematically speaking, this kind of convergence is characterized in terms of the operator norm in the space of the bounded operators with the following definition.

Definition 4.5.1 (Operator–norm convergence) *Let (T_n) be a sequence of bounded operators, defined from $(X, \|.\|_X)$ to itself. This sequence converges to T in the operator norm, if*

$$\lim_{n\to\infty} \|T_n - T\| = 0. \tag{4.9}$$

The operator norm convergence is stronger than the pointwise convergence defined below.

Definition 4.5.2 (Pointwise convergence)

$$\forall L \in X \quad \lim_{n\to\infty} \|(T_n - T)L\|_X = 0. \tag{4.10}$$

Pointwise convergence involved in approximation methods that use discretization, such as radiosity (Goral et al. [GTGB84]) and matrix row–column sampling ([HPB07]), explains the reason why such algorithms have struggled with bounding the approximation error given an arbitrary distribution. Definition 4.5.2 indicates that (T_n) converges to T, but the approximation error $\|(T_n - T)L\|_X$ depends on L.

Designing methods in the absence of compactness. Eigen–solvers are developed based on matrix representation, which implicitly implies the compactness property. To be able to use these algorithms for extracting the eigenvalues of an operator acting over an infinite–dimensional space, the compactness property is a sufficient condition to guarantee the decreasing order of eigenvalues with finite algebraic multiplicity (Theorem 4.4.1). The absence of this property for the light transport operator (even for Lambertian scenes) makes the eigenvalue extraction task more and more challenging.

4.A Appendix

4.A.1 G is bounded.

Let the scene be closed. So, $\mathbf{G}$ is defined everywhere. By the inner product defined in $\mathbb{H}$, Definition 3.30, we have

$$\begin{aligned}\|\mathbf{G}L\|^2_{\mathbb{H}} &= \int_S \int_{\Omega_{\mathbf{x}}} [(\mathbf{G}L)(\mathbf{x},\omega_i)]^2 d\omega_i dS^{\perp}_{\mathbf{x}} \\ &= \int_S \int_{\Omega_{\mathbf{y}}} L_i^2(\mathbf{y},\omega_o) d\omega_o dS^{\perp}_{\mathbf{y}} \\ &= \|L_i\|^2_{\mathbb{H}}.\end{aligned} \tag{4.11}$$

Therefore,

$$\|\mathbf{G}\| = \sup_{\|L\|_{\mathbb{H}}=1} \|\mathbf{G}L\|_{\mathbb{H}} = 1. \tag{4.12}$$

4.A.2 G is self–adjoint.

$$\begin{aligned}< \mathbf{G}L_i, L'_i >_{\mathbb{H}} &= \int_S \int_{\Omega_{\mathbf{x}}} (\mathbf{G}L)(\mathbf{x},\omega_i) L'_i(\mathbf{x},\omega_i) d\omega_i dS^{\perp}_{\mathbf{x}} \\ &= \int_S \int_{\Omega_{\mathbf{y}}} L_i(\mathbf{y},\omega_o) \mathbf{G}L'_i(\mathbf{y},\omega_o) d\omega_o dS^{\perp}_{\mathbf{y}} \\ &=< L_i, \mathbf{G}L'_i >_{\mathbb{H}},\end{aligned} \tag{4.13}$$

where $L'_i(\mathbf{x},\omega_i) = \mathbf{G}L'_i(\mathbf{y},\omega_o)$ was used to re–express the integral of the first line of equation (4.13) in terms of point $\mathbf{y}$, ω_i and ω_o are the same as in Definition 3.5.5.

4.A.3 $K_{\mathbf{x}}$ and K are self–adjoint.

For the given two incident distributions L_i, L'_i we have

$$\begin{aligned}< L'_i, K_{\mathbf{x}} L_i >_{\mathbb{H}} &= \int_{\ell} L'_i(\mathbf{x},\omega_o) K_{\mathbf{x}} L_i(\omega_o) d\omega_o dS^{\perp}_{\mathbf{x}} \\ &= \int_{\ell} L'_i(\mathbf{x},\omega_o) \int_{\Omega_{\mathbf{x}}} \rho(\mathbf{x},\omega_o,\omega_i) L_i(\mathbf{x},\omega_i) \cos\theta d\omega_i d\omega_o dS^{\perp}_{\mathbf{x}} \\ &= \int_S \int_{\Omega_{\mathbf{x}}} \int_{\Omega_{\mathbf{x}}} L'_i(\mathbf{x},\omega_o) \rho(\mathbf{x},\omega_o,\omega_i) L_i(\mathbf{x},\omega_i) \cos\theta d\omega_i d\omega_o dS^{\perp}_{\mathbf{x}} \\ &= \int_S \int_{\Omega_{\mathbf{x}}} L_i(\mathbf{x},\omega_i) \cos\theta \int_{\Omega_{\mathbf{x}}} L'_i(\mathbf{x},\omega_o) \rho(\mathbf{x},\omega_o,\omega_i) d\omega_o d\omega_i dS^{\perp}_{\mathbf{x}} \\ &=< K_{\mathbf{x}} L'_i, L_i >\end{aligned} \tag{4.14}$$

To obtain equation 4.14, the Fubini's theorem and the Helmoltz property were used. By definition, $K_{\mathbf{x}} L(\omega) = KL(\mathbf{x},\omega)$. Therefore, K is also self–adjoint.

4.A.4 $K_{\mathbf{x}}$ is compact.

The following theorem provides a necessary and sufficient condition for compactness of a linear operator [CR19]. Applying this theorem we conclude compactness of $K_{\mathbf{x}}$.

Theorem 4.A.1 *Let (L_k) be a basis for a Hilbert space $\mathbb{H}$, and T be a linear map from $\mathbb{H}$ to itself. Define*

$$\epsilon_n = \sup_{L \in span(L_k), \|L\|_{\mathbb{H}}=1} \|TL\|_{\mathbb{H}} \, , \; T_n = \sum_{k=0}^{n} < ., L_k > TL_k, \tag{4.15}$$

then

- $\epsilon_n = \|T - T_n\|$,
- *T is compact if and only if* $\lim_{n\to\infty} \epsilon_n = 0$.

Theorem 4.A.2 *The local reflectance operator is compact.*

Proof 4.A.1 *Recall the spherical harmonics, $(y_p^m)_{-p\leq m\leq p \, , 0\leq p\leq \infty}$, constructing an orthonormal basis for the space of functions defined over sphere. Therefore, for L be a radiance distribution at fixed point $\mathbf{x}$*

$$L(\omega) = \sum_{p=0}^{\infty} \sum_{m=-p}^{m=p} y_p^m(\omega) c_p^m \;\; , \;\; \omega \in \Omega_{\mathbf{x}}, \tag{4.16}$$

where $(c_p^m)_{m,p}$ are the coefficients of projecting L onto each basis element, $(y_p^m)_{m,p}$. One may reformulate equation (4.16) in terms of the spherical basis parametrized with one index set $i \in \{0, \cdots, n^2\}$, where $i = p(p+1) + m$. Next, approximating the infinite series of equation (4.16) with n^2 coefficients, i.e.

$$L_n(\omega) = \sum_{i=0}^{n^2-1} y_i(\omega) \, c_i. \tag{4.17}$$

So the previous equation is the n–th order approximation of L. Let

$$K_n := \sum_{i=0}^{n^2-1} < ., y_i > K_{\mathbf{x}} y_i$$

, since $(y_i)_{i=0}^{n^2}$ is an orthonormal basis, we have

$$\|K_{\mathbf{x}} - K_n\| \to 0,$$

in terms of operator norm. Therefore, applying the previous proposition implies the compactness of K.

4.A.5 The eigenvalues of the radiant exitance operator

Considering Lambertian scenes, Baranoski et al. [BBR97] proved that the eigenvalues of $I - T$, which they called it the radiosity matrix, are real and positive. Similar to this proof, we provide the proof in terms of continuous (operator) representation of T.

Theorem 4.A.3 *The eigenvalues of the radiant exitance operator are real.*

Proof 4.A.2 *We first prove that in Lambertian scenes, K is a positive operator. Let L_i be a directional distribution of light at a point:*

$$< KL_i, L_i >= \int_\ell \cos\theta' L_i(\mathbf{x}, \omega_o) \left(\int_{\Omega_\mathbf{x}} \cos\theta \rho(\mathbf{x}, \omega_o, \omega_i) L_i(\mathbf{x}, \omega_i) d\omega_i \right) d\omega_o dS_\mathbf{x}$$

When the material is Lambertian, we have $\rho(\mathbf{x}, \omega_o, \omega_i) = \frac{1}{\pi}\rho_d(\mathbf{x}) \geq 0$. Therefore, in this case

$$< KL_i, L_i >= \frac{1}{\pi} \int_S \rho_d(\mathbf{x}) \left(\int_{\Omega_\mathbf{x}} \cos\theta L_i(\mathbf{x}, \omega) d\omega \right)^2 dS_\mathbf{x} > 0.$$

Let (λ, Λ) be an eigenpair of T. It follows that $(\overline{\lambda}, \overline{\Lambda})$ is an eigenpair of $\mathbf{G}K$:

$$\mathbf{G}K\overline{\Lambda} = \overline{\lambda\Lambda}$$

Since K is a positive operator, there exists an operator C such that $K = C^*C$ [1]*. Thus we have*

$$\mathbf{G}C^*C\overline{\Lambda} = \overline{\lambda\Lambda}.$$

Acting C on the both sides of the previous equation leads to

$$(C\mathbf{G}C^*)C\overline{\Lambda} = \overline{\lambda}C\overline{\Lambda}$$

Since $\mathbf{G}$ is Hermitian, $C\mathbf{G}C^*$ is Hermitian too, and $\overline{\lambda}$ is one of its eigenvalues. It is therefore necessarily a real number, and so is λ.

4.A.6 Trace class property of the radiant exitance operator

Continuity of $\kappa(\mathbf{x}, \mathbf{y})$ at $\mathbf{x} = \mathbf{y}$.

Theorem 4.A.4 *The kernel function κ is bounded on any curved/flat surface (more specifically the surface needs to be of class G_2). It is however continuous in only one very specific situation: flat surfaces and portions of spheres. In the general case no value can be attributed to $\kappa(\mathbf{x}, \mathbf{x})$.*

[1] See Treves' book [Tre16] for a proof.

Proof 4.A.3 *In order to find the value in the limit, we consider the case of a point* $\mathbf{x}$ *on a surface that of class* G_2 *at this point and compute a series expansion of* $\kappa(\mathbf{x}, \mathbf{y})$ *at a nearby point* $\mathbf{y}$*. Without loss of generality, we use 2D coordinates to represent both points using an order 2 differentiable parametrization of the surface in the neighborhood of* $\mathbf{x}$*, along the tangent plane at* $\mathbf{x}$ *to the surface of the scene. We call* h *the distance between* $\mathbf{x}$ *and the projection of* $\mathbf{y}$ *on this tangent plane. Figure 4.4 shows a cross-section of the surface containing both points (their normals are in the same plane, since we consider the limit as* $\mathbf{x}$ *and* $\mathbf{y}$ *are arbitrarily close to each other).*
With these notations, the coordinates at $\mathbf{x}$ *are* $(0, 0)$*, and the coordinates at* $\mathbf{y}$ *are* $(h, \alpha_1 h + \frac{\alpha_2}{2} h^2 + \frac{\alpha_3}{6} h^h + ...)$*. Because the normal is vertical at* $\mathbf{x}$*, we have* $\alpha_1 = 0$*. The value* $\alpha_2 = 0$ *corresponds to a locally flat surface (or at least locally flat surfaces fall in this case). Coefficient* α_2 *is actually equal to the surface linear curvature at* $\mathbf{x}$*. When* $\alpha_2 \neq 0$*, the surface is locally equivalent to a spherical cap, because it has locally constant curvature. We have therefore* $\theta = \theta'$ *and*

$$\cos\theta \cos\theta' = \frac{(\frac{1}{2}\alpha_2 h^2 + \mathcal{O}(h^3))^2}{h^2 + (\frac{1}{2}\alpha_2 h^2 + \mathcal{O}(h^3))^2}.$$

So

$$\begin{aligned}
\kappa(\mathbf{x}, \mathbf{y}) &= \rho(\mathbf{x}) \frac{\cos\theta \cos\theta'}{\|\mathbf{x} - \mathbf{y}\|^2} \\
&= \rho(\mathbf{x}) \frac{(\cos\theta)^2}{\|\mathbf{x} - \mathbf{y}\|^2} \\
&= \rho(\mathbf{x}) \frac{(\frac{1}{2}\alpha_2 h^2 + \mathcal{O}(h^3))^2}{\left(h^2 + (\frac{1}{2}\alpha_2 h^2 + \mathcal{O}(h^3))^2\right)^2}.
\end{aligned}$$

Assuming that $\alpha_2 \neq 0$*, the limit of the kernel can therefore be studied when* h *tends to* 0*. We have*

$$\begin{aligned}
\lim_{h \to 0} \kappa(\mathbf{x}, \mathbf{y}) &= \rho(\mathbf{x}) \frac{(\frac{1}{2}\alpha_2 h^2 + \mathcal{O}(h^3))^2}{\left(h^2 + (\frac{1}{2}\alpha_2 h^2 + \mathcal{O}(h^3))^2\right)^2} \\
&= \rho(\mathbf{x}) \frac{1}{4} \alpha_2^2.
\end{aligned}$$

Now if the surface around $\mathbf{x}$ *is differentiable, the curvature* α_2 *of a cross section will depend on the direction* $\mathbf{v}$ *of that cross-section, involving the shape operator* $\nabla \mathbf{n}(\mathbf{x})$ *of the surface at* $\mathbf{x}$*. Writing* $\mathbf{h} = [h\cos\beta, h\sin\beta]$*, see Figure 4.4, we get*

$$\lim_{h \to 0} \kappa(\mathbf{x}, \mathbf{x} + \mathbf{h}) = \frac{\rho(\mathbf{x})}{4} \frac{\mathbf{h}^\intercal \nabla \mathbf{n}(\mathbf{x}) \mathbf{h}}{\|\mathbf{h}\|^2} = \frac{\rho(\mathbf{x})}{4} (k_1 \cos^2\beta + k_2 \sin^2\beta)^2,$$

where k_1 and k_2 are two principal curvatures.
This expression obviously depends on the direction θ at which $\mathbf{y}$ approaches $\mathbf{x}$, which proves that even if κ is bounded, it has no limit for $\mathbf{x} = \mathbf{y}$ when the two principal curvatures are not equal.

In summary, we have proved that:

- On differentiable surfaces the kernel of the operator is bounded, but generally not continuous;
- It is however continuous whenever the two principal curvatures of the surface are equal.

Specific case of the interior of a sphere.

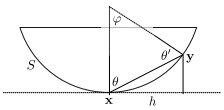


Figure 4.4: Continuity of the kernel at $\mathbf{x} = \mathbf{y}$.

When $k_1(\mathbf{x}) = k_2(\mathbf{x}) = k(\mathbf{x})$, κ becomes a continuous function at $\mathbf{y} = \mathbf{x}$ which makes T_b a trace-class operator in this case. In particular the trace of T_b can be computed by applying the theorem for trace class operators with continuous kernels:

$$\begin{aligned} \mathrm{Tr}(T_b) &= \int_S \kappa(\mathbf{x}, \mathbf{x}) dS_{\mathbf{x}} \\ &= \int_S \frac{\rho(\mathbf{x})}{4} (k(\mathbf{x}))^2 dS_{\mathbf{x}}. \end{aligned}$$

Inside a sphere of radius r, the two principal curvatures are constant and equal to $1/r$. Applying the above formula we get

$$\mathrm{Tr}(T_b) = \int_S \frac{1}{4r^2} \rho(\mathbf{x}) dS_{\mathbf{x}} = \overline{\rho_d}, \tag{4.18}$$

where $\overline{\rho_d}$ is the average albedo inside the sphere.

Chapter 5

A resolvent study of light transport

In this chapter, the main goal is to study the resolvent of the light transport operator and explain how it relates to the eigenspaces of this operator. This connection is expressed by an integral of the resolvent and the projections into the eigenspaces of the isolated eigenvalues.
In Section 5.1, we start with the resolvent of the matrix, then we update the required definitions according to the case of operators over infinite–dimensional spaces. Based on this background, we conclude the derivation regarding the resolvent of the light transport operator still holds for its isolated eigenvalues. In Section 5.2, we represent the resolvent of the light transport operator as a series in terms of the powers of the operator. Speaking of power series, we also discuss the convergence areas in terms of spectral radius. Finally, Section 5.4 elaborates the difficulties and obstacles of applying the resolvent of the light transport operator to extract eigenpairs.

5.1 Tools of the trade

In this section, we review the preliminary definitions and theorems required to proceed with this chapter. For operators acting on finite–dimensional vector spaces (matrices), the books by Brown et al. [BC09] and Kato et al. [Kat13], and for the resolvent of an operator acting on infinite–dimensional vector spaces the book by Cheverry et al. [CR19] are used.

Definition 5.1.1 (Open disk) *The open disk $D(z, r)$ is referred to the region of the complex plane which is defined as follows*

$$D(z_0, r) = \{z \in \mathbb{C} \mid |z - z_0| < r\}.$$

Definition 5.1.2 (Singular point) *Let f be a complex function defined over $D \subset \mathbb{C}$. Point $z_0 \in D$ is called singular, if f can not be expressed analytically at z_0.*

Definition 5.1.3 (Isolated singular point) *A singularity z_0 is called isolated if there exists at least a disk $D(z_0, \epsilon)$ such that f is analytic on $D(z_0, \epsilon) \setminus \{z_0\}$.*

Theorem 5.1.1 (Cauchy's integral formula) *Let f be an analytic complex function defined over a subset of the complex plane, denoted by U. Let γ be the positively oriented (oriented counterclockwise) boundary of U, then for any $a \in U$*

$$f(a) = \frac{1}{2\pi i} \oint_\gamma \frac{f(z)}{z-a} dz. \tag{5.1}$$

The proof of the theorem uses the Cauchy integral theorem. Since f is infinitely differentiable,

$$f^n(a) = \frac{n!}{2\pi i} \oint \frac{f(z)}{(z-a)^{n+1}} dz, \tag{5.2}$$

this formula is often referred as *Cauchy's differentiation formula.*

Theorem 5.1.2 (Laurent series) *Let f be a function which is analytic over an annular domain $r < |z - z_0| < R$, and Γ be a positively oriented closed curve around z_0. Then, $f(z)$ has the following representation*

$$f(z) = \sum_{n=0}^{\infty} a_n (z - z_0)^n + \sum_{n=1}^{\infty} \frac{b_n}{(z - z_0)^n}, \tag{5.3}$$

where

$$a_n = \frac{1}{2\pi i} \oint_\Gamma \frac{f(z)\, dz}{(z - z_0)^{n+1}}, \tag{5.4}$$

and

$$b_n = \frac{1}{2\pi i} \oint_\Gamma \frac{f(z)\, dz}{(z - z_0)^{-n+1}}. \tag{5.5}$$

Definition 5.1.4 (Residue) *Under the same assumptions of Theorem 5.1.2, where z_0 is an isolated singular point of f, the residue of function f at $z = z_0$ is referred to b_1 coefficient in equation (5.3), i.e.*

$$\oint_\Gamma f(z)\, dz = 2\pi i b_1$$

, we denote this coefficient with $Res_{z=z_0} f(z)$.

Theorem 5.1.3 (Cauchy's residue theorem) *Let Γ be a positively oriented closed curve the interior of which contains a finite number of singular points $\{z_1, \cdots, z_k\}$ of f, then*

$$\oint_\Gamma f(z)\, dz = 2\pi i \sum_{i=1}^{i=k} Res_{z=z_i} f(z).$$

Theorem 5.1.4 (Cauchy–Hadamard theorem) *Consider the analytic expression of f, $f(z) = \sum_{n=0}^{\infty} a_n(z-a)^n$, then the radius of convergence, r, of this series is defined as*

$$\frac{1}{\limsup_{n\to\infty}(|a_n|^{\frac{1}{n}})}.$$

5.1.1 Analytic continuation for a complex function

The aim of this section is to recall *the Analytic Continuation* technique. Let $f(z) = \sum_{n=0}^{\infty} a_n(z-z_0)^n$ be the Taylor series of $f(z)$ around z_0. This series is absolutely convergent in a disk $D(z_0, r_0)$. In addition to have the analytic expression of $f(z)$ valid on this disk, it is also possible to extend this analytic form in a larger area. This is exactly the task of the analytic continuation process.
In the following of this section, we recall this process using an example then, apply this tool for a specific function, the resolvent function.

Example 5.1.1 *Consider function $f(z) = \frac{1}{z}$. This function has the following expansion series*

$$f(z) = \sum_{n=0}^{\infty}(-1)^n(z-1)^n,$$

which is absolutely convergent in the disc $D(1,1)$. The aim is to express $f(z)$ around other point, say a, in $D(-1,1)$. That is

$$f(z) = \sum_{n=0}^{\infty} a_n(z-a)^n.$$

Since this is supposed to be the Taylor series of $f(z)$ around $z = a$, we know

$$a_n = \frac{f^n(a)}{n!}.$$

Applying Cauchy's differentiation formula for a circle, centered at a with a radius less than $1 - |a-1|$, we obtain

$$a_n = (-1)^n a^{-n-1}$$

[1]. *So $f(z)$ is represented as follows*

$$f(z) = \sum_{n=0}^{\infty} a_n(z-a)^n = \sum_{n=0}^{\infty}(-1)^n a^{-n-1}(z-1)^n = a^{-1}\sum_{n=0}^{\infty} a_n(1-\frac{z}{a})^n, \quad (5.6)$$

using Cauchy–Hadamard theorem, the radius of convergence of the series (5.6)

[1] Since this example is a very known case to explain analytic continuation, to avoid repetition, the details of computing a_n coefficients are skipped, for more information on the subject check Ahlfors et al.'s book [Ahl79].

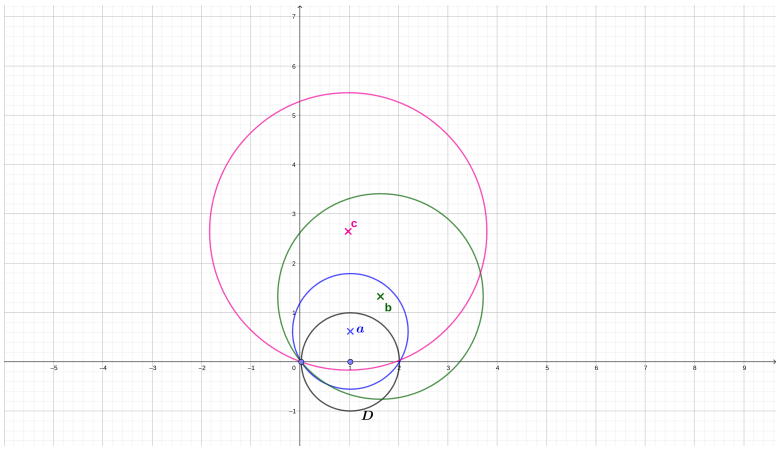


Figure 5.1: The analytic continuation allows to extend a series expression of a function beyond its initial convergence disk when carefully avoiding poles of the function.

is $|a|$, see Figure 5.1.
Therefore $f(z)$ might be represented in a larger area than D, that is $D'(a, |a|)$. Choosing the points carefully, avoiding poles of the function, this procedure can be repeated, which leads to extending the analytic form of $f(z)$ on the complex plane, see Figure 5.1.

5.1.2 Resolvent of an operator over a finite–dimensional vector space

This section is dedicated to explain the concept of the resolvent of an operator defined on a finite vector space, X. The main goal is to obtain a decomposition of the resolvent in terms of infinite series, Laurent series. This series is expressed in terms of an eigenvalue of the operator, since the singularities of the resolvent are exactly the eigenvalues of the operator.
At the end of this section, we bring a proposition stating a line integral of the resolvent in terms of the sum over projections of the resolvent into the related eigenspace.

Definition 5.1.5 (Resolvent of an operator) *Let $T : X \longrightarrow X$ be a linear map. The operator–valued function $R_T(z)$*

$$R_T(z) := (T - z)^{-1} \tag{5.7}$$

is called the resolvent of T. This operator is defined when z is not an eigenvalue.

Proposition 5.1.1 (Resolvent formula) *Let z_1, z_2 to not be eigenvalues of T. Then*

$$(z_1 - z_2)R_T(z_1)R_T(z_2) = R_T(z_1) - R_T(z_2) \tag{5.8}$$

This equality holds for the more general case, such as T be a bounded operator(or unbounded but closed) on a Banach space. For the proof of the proposition refer to [CR19].
Applying equality (5.8), one may write

$$R_T(z) = [1 - (z - z_0)R_T(z_0)]^{-1}R_T(z_0) = \sum_{n=0}^{\infty}(z - z_0)^n R_T^{n+1}(z_0). \tag{5.9}$$

This infinite series is absolutely convergent inside $D(z_0, r)$ by Theorem 5.1.4, where

$$r = (\lim_{n\longrightarrow\infty} \|R_T^n(z_0)\|^{\frac{1}{n}})^{-1}. \tag{5.10}$$

Note that the value of r in equation above is also related to the spectral norm of R_T, see Definition 5.1.6. This connection is concluded from *the Gelfand spectral radius theorem* [Nev93]. This theorem expresses the spectral norm of the matrix in terms of the limit of $\|T^n\|^{\frac{1}{n}}$ as $n \to \infty$:

$$\|T\|_{spr} = \lim_{n\to\infty} \|T^n\|^{\frac{1}{n}}.$$

Therefore,

$$r = \|R_T(z_0)\|_{spr}^{-1}. \tag{5.11}$$

Definition 5.1.6 (Spectral radius of a matrix) *Let T be a $k \times k$ matrix with the set of eigenvalues $\{\lambda_1, \cdots, \lambda_k\}$. The spectral radius norm is defined*

$$\|\cdot\|_{spr} := \max_{1\le i\le k} |\lambda_i|.$$

Note that since $R_T(z)$ is analytic on $D(z_0, \|R_T(z_0)\|_{spr}^{-1})$ with Taylor series expansion on the right–hand–side of (5.9), so

$$\frac{d^n R_T(z)}{dz^n} = n! R_T^{n+1}(z_0) \;\; n = 1, 2, \cdots \tag{5.12}$$

Neumann series for the resolvent. To expand $R_T(z)$ in terms of T,

$$R_T(z) = -(z - T)^{-1} = -z^{-1}(I - z^{-1}T) = -\sum_{n=0}^{\infty} \frac{T^n}{z^{n+1}}, \tag{5.13}$$

this series is convergent if $|z| > \|T\|_{spr}$.
One may reformulate equation (5.3) by

$$f(z) = \sum_{-\infty}^{\infty}(z - z_0)^n A_n, \tag{5.14}$$

where

$$A_n = \frac{1}{2\pi i}\int_{\Gamma} z^{-n-1} f(z)dz.$$

Let $\{\lambda_1, \lambda_2, \cdots, \lambda_m\}$ be the set of eigenvalues of T. Note that the eigenvalues of T are singularities of $R_T(z)$. Therefore, it is possible to express the resolvent with the Laurent series around a specific eigenvalue. Without loss of generality, suppose $\lambda_i = 0$ for some i. So, one may formulate $R_T(z)$ as follows

$$R_T(z) = \sum_{-\infty}^{\infty} z^n A_n,$$

where

$$A_n = \frac{1}{2\pi i} \int_\Gamma z^{-n-1} R_T(z) dz, \tag{5.15}$$

where Γ is a positive–oriented circle that only contains $z = 0$.
Let Γ' contain Γ, it can be shown [Kat13]

$$A_n A_m = \frac{\alpha_n + \alpha_m - 1}{2\pi i} \int_\Gamma z^{-n-m-2} R_T(z) dz = (\alpha_n + \alpha_m - 1) A_{n+m+1}, \tag{5.16}$$

where

$$\alpha_n = 1 \;\; n \geq 0,$$

and

$$\alpha_n = 0 \;\; n < 0.$$

To obtain equation (5.16), one may start from

$$\begin{aligned} A_n A_m &= (\frac{1}{2\pi i})^2 \int_{\Gamma'} \int_\Gamma z^{-n-1} z'^{-m-1} R_T(z) R_T(z') dz\, dz' \\ &= (\frac{1}{2\pi i})^2 \int_{\Gamma'} \int_\Gamma z^{-n-1} z'^{-m-1} (z'-z)^{-1} [R_T(z') - R_T(z)] dz\, dz', \end{aligned} \tag{5.17}$$

where the RHS of the second line is obtained using resolvent formula (Equation (5.8)). Then, the following equations are also applied

$$\begin{aligned} \frac{1}{2\pi i} \int_\Gamma z^{-n-1} (z'-z)^{-1} dz &= \alpha_n z'^{-n-1} \\ \frac{1}{2\pi i} \int_{\Gamma'} z'^{-m-1} (z'-z)^{-1} dz &= (1-\alpha_m) z^{-m-1}. \end{aligned} \tag{5.18}$$

Equations listed in (5.18) are derived using Cauchy's integral formula.
Therefore, using equation (5.16)

$$A_{-1}^2 = -A_{-1},$$

which concludes that $-A_{-1}$ is a projection[2].
Let $D := -A_{-2}$, so

$$A_{-n} = -D^{n-1} \;\; , \;\; n \geq 2$$

[2] A linear map from a vector space to itself is called projection if $P^2 = P$.

and

$$A_n = S^{n+1} \ , \ n \geq 0$$

, where $S := A_0$.
Finally, the Laurent series of $R_T(z)$ around a given eigenvalue λ_i is

$$R_T(z) = -(z-\lambda_i)^{-1}P_i - \sum_{n=1}^{\infty}(z-\lambda_i)^{-n-1}D_i^n + \sum_{n=0}^{\infty}(z-\lambda_i)^n S_i^{n+1} \tag{5.19}$$

, where

$$P_i = \int_{\Gamma_i} R_T(z)dz$$

in which Γ_i is a circle containing just λ_i. Note that P_i indicates $-A_{-1}$, where the subscript i indicates the index of the eigenvalue, considered to from the Laurent series around it. Using the Laurent series of the resolvent, it is possible to form a decomposition of X using P_i as follows

$$X = M_i \oplus M_i', \tag{5.20}$$

where

$$M_i = P_i X \ , \ M_i' = (I - P_i)X.$$

Applying the Laurent expansion of the resolvent around λ_i and Cauchy's residue theorem, it is possible to project a vector, belonging to X, into the eigenspace associated to λ_i, we denote it by Λ_i. The following theorem expresses this fact.

Lemma 5.1.1 *Let Γ_i be a closed curve with positive direction only containing λ_i. Then*

$$\oint_{\Gamma_i} R_T(z)dz = -2\pi i P_i \tag{5.21}$$

Proof 5.1.1 *From Cauchy's residue theorem, we have*

$$\begin{aligned}\oint_{\Gamma} R_T(z)dz &= 2\pi i Res_{z=\lambda_i} R_T(z) \\ &= 2\pi i Res_{z=\lambda_i} \left(-(z-\lambda_i)^{-1}P_i - \sum_{n=1}^{\infty}(z-\lambda_i)^{-n-1}D_i^n + \sum_{n=0}^{\infty}(z-\lambda_i)^n S_i^{n+1}\right) \\ &= -2\pi i(P_i).\end{aligned} \tag{5.22}$$

Applying Cauchy's residue theorem, it is possible to derive an equation in terms of the resolvent and projections onto the eigenspaces. The following proposition illustrates this fact.

Proposition 5.1.2 *Let Γ be a closed curve with positive direction containing isolated eigenvalues $\lambda_1, \cdots, \lambda_k$ inside. It can be shown*

$$\int_{\Gamma} R_T(z)dz = -2\pi i \sum_{i=1}^{i=k} P_i. \tag{5.23}$$

Proof 5.1.2 *Since it is assumed that eigenvalues are isolated, therefore, it is possible to form closed curves those contain each eigenvalue separately, and every eigenvalue lies at the center of the associated circle. Using 5.1.1 for each eigenvalue separately, concludes equation* (5.23).

Remark 5.1.1 *It is worth mentioning that the derivation of equation* (5.23) *does not depend on an extra assumption on T except, having isolated eigenvalues. Considering stronger assumptions on a matrix, such as normality or being diagonalizable allows concluding equation* (5.23) *simpler, i.e. without using Laurent series.*

Example 5.1.2 *Let T be diagonalizable. Therefore*

$$T = \sum_{i=1}^{i=k} \lambda_i < ., \overline{\Lambda_i} > \Lambda_i. \tag{5.24}$$

Based on the fact that eigenvalues of $(T-z)^{-1}$ are related to the eigenvalues of T, i.e. $(\lambda_i - z)^{-1}$ are $R_T(z)$'s eigenvalues, and on the other hand, eigenvectors of T and $R_T(z)$ are the same, applying (5.24)*, one may build a decomposition for $R_T(z)$ as follows*

$$R_T(z) = \sum_{i=1}^{i=k} \frac{1}{\lambda_i - z} < ., \overline{\Lambda_i} > \Lambda_i. \tag{5.25}$$

Consider the same assumptions as Proposition 5.1.2. Residue of $R_T(z)$ at $z = \lambda_i$ is

$$Res_{z=\lambda_i} R_T(z) = \lim_{z \to \lambda_i} [(z - \lambda_i) R_T(z)] = - < ., \overline{\Lambda_i} > \Lambda_i. \tag{5.26}$$

On the other hand, we know

$$\int_{\Gamma_i} R_T(z)dz = 2\pi i \, Res_{z=\lambda_i} R_T(z) = -2\pi i < ., \overline{\Lambda_i} > \Lambda_i = -2\pi i P_i. \tag{5.27}$$

So, integrating the resolvent on Γ confirms equality in (5.23).

5.1.3 Resolvent of an operator over an infinite–dimensional vector space

Before starting a discussion about the resolvent of light transport, we need to clarify which definitions are generalized to the case when X is infinite–dimensional vector space, such as a Hilbert space. This attention is required, since light transport is defined from $\mathbb{H}$ to itself, and this space is infinite–dimensional.

Definition 5.1.7 (The resolvent set) *The resolvent set is the set of $z \in \mathbb{C}$ such that the operator $T - z : \mathbb{H} \longrightarrow \mathbb{H}$ is a bijective map*[3]*. We denote this set with $\rho(T)$* [4]*.*

Definition 5.1.8 (spectrum) *The spectrum of T is the set $sp(T) := \mathbb{C} \setminus \rho(T)$.*

It is worth mentioning that since $\mathbb{H}$ is infinite–dimensional then, the spectrum of T does not contain only eigenvalues. In other words, each point of the spectrum is not necessarily an eigenvalue. Therefore, it is required to generalize some definitions.

Definition 5.1.9 (Eigenvalue) *An eigenvalue λ is a number such that the kernel of $T-\lambda$ is not empty. The set of eigenvalues of T is called point spectrum, and denoted by $sp_p(T)$.*

Definition 5.1.10 (Spectral radius of a general operator) *The definition of the spectral radius of a general operator, we denote it by $r(T)$, is defined similar to the matrix case, however in this case, the maximum absolute value is taken over the entire spectrum, i.e.*

$$r(T) := \max_{\lambda \in sp(T)} |\lambda|.$$

Proposition 5.1.3 *Let T be bounded. Then $z \in \rho(T)$ if and only if $T - z$ is bijective. Therefore, $(T - z)^{-1}$ is bounded and well–defined for $z \in \rho(T)$. Moreover,*

$$sp(T) \subset D(0, \|T\|), \tag{5.28}$$

[CR19]. Based on Proposition 5.1.3, the resolvent of a bounded operator over $\mathbb{H}$ is defined for $z \in \rho(T)$. Similar to the finite–dimensional case, because the operator $(T - z)$ is invertible, it has the infinite series expression

$$R_T(z) = -\sum_{n=0}^{\infty} \frac{T^n}{z^{n+1}}, \tag{5.29}$$

for $|z| > \|T\|$.

Remark 5.1.2 *The resolvent formula holds for $T : \mathbb{H} \longrightarrow \mathbb{H}$, for $z_1, z_2 \in \rho(T)$,*

[CR19].
Applying the resolvent formula for light transport enables to represent its resolvent with Taylor expansion, as equation (5.9). Note that this expansion is valid for both norms $\|T\|$ and $r(T)$, however these two norms do not necessarily coincide, unlike the matrix case in which these two norms coincide when the matrix is normal [CR19]. In general, the spectral norm of a bounded operator is less than the operator norm using Gelfand's formula, which is also valid for an operator defined on $\mathbb{H}$.

Proposition 5.1.4 (Gelfand's Formula) *For a bounded operator R*

$$r(R) = \inf_{n \in \mathbb{N}^*} \|R^n\|^{\frac{1}{n}} \leq \|R\|. \tag{5.30}$$

[3] A map is called bijective if it is both injective and surjective.
[4] Note that $\rho(T)$ is not the same as $\rho_d(\mathbf{x})$ which is the albedo at surface point $\mathbf{x}$.

5.2 Resolvent of light transport

Since the light transport operator is bounded, then all the mathematical properties reviewed in Section 5.1.3 remain valid for it. Therefore,the resolvent of light transport also verifies equation (5.12), and the Taylor representation in (5.9) is valid on $D(z_0, r(R_T(z_0))^{-1})$. To be able to extend the representation of the resolvent on a wider area of the complex plane, one may think of analytic continuation. To make analytic continuation for the resolvent of light transport practically feasible, it is required to use an equation containing powers of T. Practically speaking, to build any algorithm related to the resolvent of the light transport, images of powers of T are only available materials. The key to bring powers of T into the play is the Neumann series of resolvent. However, to be able to use the Neumann series of the resolvent z must be out of $D(0, \|T\|)$. In Section 5.2.1 we will elaborate this idea.
This is also possible to build the Laurent series of the light transport operator around an isolated eigenvalue. In general the eigenvalues of this operator may not be isolated, however, Proposition 5.1.2 is still valid for set of isolated eigenvalues of the operator [Cha11].

5.2.1 Analytic continuation for the resolvent of light transport

When T is acting on a finite–dimensional space then, the only primary information needed to do analytic continuation for R_T, is the location of eigenvalues. However, when the operator defined on an infinite–dimensional space, it becomes even more complicated. Since, the singularities are the whole spectrum, not only eigenvalues.
To be able to use the Neumann series of the resolvent in its Taylor representation, z must be outside of $D(0, \|T\|)$. Therefore, to start the analytic continuation, let $z_0 \in \mathbb{C} \setminus D(0, \|T\|)$. Equation (5.9) provides the analytic expression of R_T around z_0, and plugging the Neumann series of R_T in this equation gives

$$\begin{aligned} R_T(z) &= \sum_{n=0}^{\infty} (z - z_0)^n R_T^{n+1}(z_0) = \sum_{n=0}^{\infty} (z - z_0)^n [- \sum_{m=0}^{\infty} \frac{T^m}{z_0^{m+1}}]^{n+1} \\ &= \sum_{n=0}^{\infty} (z - z_0)^n c_n, \end{aligned} \tag{5.31}$$

where

$$c_n = (-1)^{n+1} [\sum_{m=0}^{\infty} \frac{T^m}{z_0^{m+1}}]^{n+1} \tag{5.32}$$

The convergence radius of the series is

$$r_0 = \frac{1}{r(R_T(z_0))} = \frac{1}{\sup_{\lambda \in sp(T)} |\lambda - z_0|^{-1}} = \frac{1}{dist(z_0, sp(T))^{-1}} = dist(z_0, sp(T)) \tag{5.33}$$

where

$$dist(z_0, sp(T)) = \inf_{\lambda \in sp(T)} |z_0 - \lambda|.$$

Equation (5.33) illustrates the dependency of the convergence disk on the nearest point of the spectrum. Therefore, before choosing a new point for the next step of the analytic continuation, it is essential to know the nearest spectrum element. Unfortunately, in general, no primary information on the spectrum of light transport is available.
Dealing with operators over infinite–dimensional spaces, the compactness property usually helps to simplify problems. For this particular problem, under the assumption of compactness, computation of the convergence radius of the circle in any step of analytic continuation simplifies to searching the nearest eigenvalue. From our study, we know the state in which the exitance radiant operator is compact. So we finish this section for this special case, and under the assumption of knowing the location of the first (largest) eigenvalue, λ_1. Under compactness property the convergence radius of the first circle is

$$r_0 = |z_0 - \lambda_1|. \tag{5.34}$$

Then, for the next step of analytic continuation, we chose $a \in D(z_0, r_0)$. To compute the new coefficients of the new representation of the resolvent, we need to evaluate n–th order derivatives of the resolvent at $z = a$. From Taylor series representation around point $z = a$, we have

$$a_n = \frac{1}{n!}\frac{d^n R_T(a)}{dz^n} = R_T(a)^{n+1} = [-\sum_{m=0}^{\infty} \frac{T^m}{a^{m+1}}]^{n+1} = (-1)^{n+1}[\sum_{m=0}^{\infty} \frac{T^m}{a^{m+1}}]^{n+1}. \tag{5.35}$$

Therefore, $R_T(z) = \sum_{n=0}^{\infty} a_n (z-a)^n$ is the new representation of the resolvent in the new disk $D(a, r_1)$ where

$$r_1 = |a - \lambda_1|.$$

5.3 Application of the resolvent to extract eigenvalue

As it is pointed out in the previous section, equation (5.23) holds under the assumption of having isolated eigenvalues. Under this only assumption, the information it brings is the projection coefficients of $L \in \mathbb{H}$ onto the related eigenspaces, and it can be theoretically used to compute images of the eigenfunctions. However, considering some extra assumptions on T may bring us more information. Namely, if we suppose that the set of eigenfunctions of T is complete and the first eigenvalue is dominant, we are able to extract an estimate of the dominant eigenvalue. In theory, this is possible using both equations

(5.24) and (5.26) together. In other words

$$\forall L \in \mathbb{H} : \lambda_1 \approx -2\pi i \frac{TL}{\int_{\Gamma_1} R_T(z)Ldz}. \tag{5.36}$$

To have decreasing eigenvalues in magnitude, the compactness property of the operator is sufficient (See Theorem 4.4.1). So, we assume compactness as a primary assumption on the operator. And for completeness, one may address the following conditions:

- When T is self–adjoint [Rud73].
- when T is normal [NS82].

In practice, to use the resolvent of light transport in this way, it is required to check whether the listed assumptions are provided or not. From this study, we know that the light transport is not compact in general. Therefore, in general, there is no information on the set eigenvalues to even be sure that they are discrete or isolated. The operator is not neither self–adjoint nor normal. Therefore, in the absence of completeness of light transport's eigenfunctions, decomposing an arbitrary light distribution using eigenfunctions is hopeless. Except, if the distribution belongs to the space spanned by eigenfunctions.

5.4 Discussion

In spite of the theoretical beauty of the resolvent study of an operator, it is impossible to apply the resolvent of light transport for any eigenpair–related objective. This is basically because of the nature of the resolvent concept and light transport. The resolvent concept is highly dependent on the singularities of the operator. On the other hand, the light transport operator does not have any *nice* properties, leading to extract information on the spectrum of T, such as compactness or normality. The only information, extracted from our study on light transport, is that the eigenvalues of the radiant exitance operator are real.
When the operator is not compact, it is very challenging to obtain information about eigenvalues. Since the light transport operator is not compact in general, there is no mathematical proof to claim that eigenvalues are isolated, or there is an decreasing order on the set of eigenvalues [5].
About the analytic continuation of the resolvent, even under the assumption of compactness, it it essential to have information of the location of eigenvalues, which is not available for an arbitrary scene. Under the assumption of having this information too, the only way to perform the analytic continuation practically, is to use powers of T. On the other hand, there is a limitation of using Neumann series of T in the expression of the resolvent namely, $z \in \mathbb{C}\backslash D(0, \|T\|)$,

[5] However, eigenvalues of light transport matrices, provided by descritization, are always decreasing.

and computing the coefficients of each new representation of the resolvent suffers from numerical instability.
It is worth mentioning that, the Neumann series of the resolvent diverges inside $D(0, \|T\|)$, but the resolvent obeys the Laurent series expression around each eigenvalue. However, in this case there is no practical way to do the job. That is also an obstacle to investigate the idea described in Section 5.3.

Part II

Computing eigenvalues using Monte Carlo algorithms

Information on test scenes

We implemented and tested all methods described in the manuscript on various Lambertian test scenes. These scenes are extremely simple in order to compare the results of each method with the reference values.
Reference matrix calculation uses an eigensolver. Each scene is first tessellated. Then, surface–to–surface form–factor matrix is computed using Monte Carlo integration. Finally, the reference eigenvalues are computed using Arpack++.

Below is a small description for each scene.

- "TwoSquares": This is the most basic scene possible: only two squares facing each other. The scene only contains two polygons, the eigenvalues come by pairs of opposite signs. On this scene, the light transport kernel is continuous over its entire domain, see Table 5.1.
- "TwoSquaresNoSymmetry": Same scene than above, but with a geometric setup that prevents any symmetry. Here again eigenvalues come by pair of opposite signs. The kernel is here again continuous, see Table 5.1.
- "Contact": Elementary scene with a discontinuous light transport kernel, see Table 5.1.
- "Cubes0027": 27 cubes stacked regularly, but with a random orient. The light transport kernel is continuous, see Table 5.2.
- "Cubes0343": 343 cubes stacked regularly ($7\times7\times7$), with a random orient. The light transport kernel is continuous, see Table 5.2.
- "CornellBox": A classical Cornell Box that has been re–meshed in order to avoid contact surfaces. Contact surfaces indeed results in the kernel to be null on finite areas. Albedo is 0.295 (resp. 0.15) on the right face (resp. left face), and 0.4 everywhere else, see Table 5.2.

Scene	 "TwoSquares"	 "TwoSquaresNoSymmetry"	 "Contact"
κ	Continuous Bounded	Continuous Bounded	Not continuous Unbounded

Table 5.1: Test scenes: For each scene the albedo is 0.8.

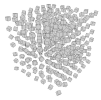
Scene	 "Cubes0027"	 "Cubes0343"	 "CornellBox"
κ	Not Continuous Bounded	Not Continuous Bounded	Not continuous Unbounded

Table 5.2: Test scenes: For each scene the albedo is 0.8, except the Cornell box scene, which has albedos of $0.15, 0.295, 0.4$ and 0.5.

Chapter 6

Stable computation of the largest eigenvalue

In this chapter, we estimate the dominant eigenvalue using a Monte Carlo formulation of the power iteration method. In this method, a sequence of vectors that converges to the eigenvector associated with the dominant eigenvalue is generated. To apply this method to the light transport operator, it is required to generate the images of the powers of light transport given an initial light distribution L. Since producing an image of long light paths, say for a large k, would be noisy with path tracing, we use Metropolis light transport (MLT) for $T^k L$, with an additional bounce of path tracing for $T^{k+1}L$. Using this strategy, both images of $T^k L$ and $T^{k+1}L$ will have the same scaling factor. Applying these two images, the dominant eigenvalue is obtained by computing the ratio of the two images.

6.1 Method's strategy

6.1.1 Power iteration for a matrix

The power iteration [1] method is a simple *eigenvalue algorithm*. In the field of numerical analysis, eigenvalue analysis is referred to as the method in which the aim is computing the eigenvalues of a given matrix. This method starts with a random vector, say b_0, and the goal is to achieve the eigenvector associated with the largest eigenvalue in magnitude. Namely, a sequence of vectors, (b_k), is generated, then under some sufficient conditions, the sequence converges to the eigenvector. The power iteration method is stated as follows:

1. Build a random vector to start the iteration process.

[1] This algorithm is also known as *Von Mises iteration*.

2. Generate vectors using the following formula

$$b_{k+1} = \frac{Ab_k}{\|Ab_k\|} \ , \ k \geq 1. \tag{6.1}$$

If the largest eigenvalue is strictly greater than others in magnitude, and b_0 has at least one non–zero component in the direction of the associated eigenvector, then (b_n) converges to that eigenvector [VLG96]. And the sequence $\frac{b_k^t A b_k}{b_k^t b_k}$ converges to the dominant eigenvalue.

6.1.2 Power iteration for light transport

The light transport operator is an operator over an infinite–dimensional space, therefore it does not act necessarily as a matrix. To apply the power iteration method for this operator, it is required to adapt the arguments regarding the convergence issue of this method. We consider the light transport operator when it is compact, therefore all the eigenvalues are isolated.
Define (a_n) as the sequence of the ratio of two consecutive powers, that is $a_n := \frac{\|T^{n+1}L\|_2}{\|T^nL\|_2}$. Since this sequence depends on L, its convergence also depends on the representation of this vector. If the set of eigenvectors is complete, then each light distribution can be expressed in terms of eigenfunctions, (Λ_i), provided by eigendecomposition:

$$TL = \sum_{i=1}^{\infty} \lambda_i < L, \overline{\Lambda_i} > \Lambda_i. \tag{6.2}$$

Using the equation above, we have

$$T^kL = \sum_{i=1}^{\infty} \lambda_i^k < L, \overline{\Lambda_i} > \Lambda_i. \tag{6.3}$$

Therefore,

$$\lim_{n\to\infty} a_n = |\lambda_1|. \tag{6.4}$$

Let $I(T^{n+1}L)$ and $I(T^nL)$ indicate the correspondent images of $T^{n+1}L$ and T^nL respectively. Since the image is the result of a linear operation, projection, the sequence $\frac{\|I(T^{n+1}L)\|_2}{\|I(T^nL)\|_2}$ also converges to λ_1:

$$\lim_{n\to\infty} \frac{\|I(T^{n+1}L)\|_2}{\|I(T^nL)\|_2} = |\lambda_1|. \tag{6.5}$$

6.2 Computing required images

To be able to obtain $|\lambda_1|$ from equation (6.5), it is required to have the images of T^kL and $T^{k+1}L$ for a large value of k. Since computing these images using

path tracing methods consumes a large amount of time, and on the other hand the rendered image for a large k is noisy, the Metropolis light transport (MLT) method sounds a better option. However, in comparison to path tracing, the energy of the rendered image, the norm of the image vector is exact up to an unknown factor. Namely, if the exact energy of an image be E, then each run of MLT produces an image with a new unknown scaled energy, say αE. For our purpose, paying attention to this fact is very crucial. Therefore, we make a modification on the MLT algorithm, and also add a new step of path tracing to render the required images with the same α.
In Section 6.2.1, we review the original MLT algorithm, and in Section 6.2.2, we describe the way to compute the required images with the same scaling factor for the given Lambertian scene.

6.2.1 A brief reminder of Metropolis light transport (MLT) method

It was first Veach [Vea98] who applied the Metropolis algorithm, developed in computational physics, for transport problems. For the case of light transport, the aim is to compute an integral, $\int_D f(\overline{x})d\mu_{\overline{x}}$ [2], which is the energy transported by a set of light paths $\overline{x} \in D$ to the image plane, i.e. the value of this integral determines the illumination on the screen. The algorithm produces a Markov chain of light paths that follows a probability density proportional to f. It does so by mutating light paths and accepting them according to an acceptance probability that is computed as the ratio of the paths contributions normalized by their respective sampling probabilities.
The advantage of the Metropolis method is that the rendered image is unbiased and it is usable for any scene with any material model and geometry. The Metropolis algorithm may be started with either a single light path or multiple ones. Generating an image with a single light path assumes that the ergodicity [3] of the set of mutation strategies used, which is not necessarily the case, depending on the implementation. Contrariwise, if multiple light paths are used, they need to be sampled using the same probability density, in order to keep the balance between the different related components (due to the mutations) of the total set of light paths. One solution to do this is to generate initial light paths proportional to f. Generating multiple light paths ensures building different families of light paths. Considering the case with multiple initial paths, the method is separately performed for each initial path in parallel, but the result of mutations on each initial path is recorded in a common image.
Let $\overline{X_0}$ be an initial path generated by a sampling method. Recall that MLT does not depend on $\overline{X_0}$, but it is important to generate the samples proportional to f. For example, using any importance sampling strategy provides us with this goal. Once $\overline{X_0}$ is built, the mutation step is repeated long enough to reach

[2] $\overline{x}$ indicates the light path, and $\mu_{\overline{x}}$ is the corresponding measure defined on the space of light paths.

[3] Considering two arbitrary light paths, ergodicity means that one light path can be reached by performing a finite set of mutations on the another one.

the equilibrium distribution [4]. Peskun [Pes73] illustrated that to obtain this distribution faster, the best strategy is to maximize $a(\overline{x} \to \overline{y})$, the acceptance probability of $\overline{y}$ given $\overline{x}$, and $a(\overline{y} \to \overline{x})$. It is sufficient to compute $a(\overline{x} \to \overline{y})$ as follows

$$a(\overline{x} \to \overline{y}) = \min\{1, \frac{f(\overline{y})P(\overline{y} \to \overline{x})}{f(\overline{x})P(\overline{x} \to \overline{y})}\}, \tag{6.6}$$

where $P(\cdot \to \cdot)$ is *the tentative transition function* [5], and can be chosen arbitrary.

6.2.2 Producing the required images with the same scaling factor

To compute the image of $T^k L$ using MLT first, it is required to generate at least one initial path, $\overline{X_0}$, with k number of bounces. This initial path starts from the given static light source, denoted by $\mathbf{x_s}$, and makes k number of intersections in the scene. Note that, the initial path is generated such that it is not connected to the camera at this stage, i.e. $\overline{X_0}$ is of the form

$$\overline{X_0} = (\mathbf{x_s}, \mathbf{x_1}, \cdots, \mathbf{x_k}).$$

The reason behind this modification is that if the algorithm only selects the paths that can be connected to the camera after k bounces, then the result will be biased. Since the paths that can not be connected to the camera after k intersections can still contribute to the second image after making one more intersection in the scene. Connecting the last vertex of light paths to the camera is done in the step of projecting the contribution of paths into the screen plane, or in the stage of recording pixel values.
Once the initial path is generated, then the mutation process starts operating. Speaking of the mutation, we perform *the point perturbation* strategy on the given path, which preserves the length of the path [Vea98]. Using this strategy, we keep the calculation of the acceptance probability simple. However, to ensure covering all necessary families of paths (ergodicity), more than one initial path is required. We explain the strategy for an initial path, and then generalize it for multiple seeds.
Generating images using a single seed path. We proceed the following three steps:

- **Step 1**. Producing a set of paths with length k, set D, by mutating one initial path or more.
- **Step 2**. Adding one more vertex to each path belonging to D using path tracing.
- **Step 3**. Recording pixel values corresponding to each related path of length k and $k+1$. After this step, we will have two separate images of

[4] When a Markov chain reaches an equilibrium distribution, or a stationary distribution then, it will maintain its behavior for future times.
[5] This function returns the probability density of $\overline{y} = \overline{X_j}$ given $\overline{x} = \overline{X_{j-1}}$.

$T^k L$ and $T^{k+1} L$, both proportional to the unbiased image, with a common unknown factor.

In the remainder of this section, we elaborate each of these steps in detail.
Step 1. To start the MLT algorithm, either with a single seed or multiple seeds, it is required to generate at least one initial path according to a probability density function proportional to f. In terms of practice, we sample $\overline{X_0}$ using an importance sampling technique.
First, we sample a direction at $\mathbf{x_s}$, according to the source emittance distribution, to trace a new point, $\mathbf{x_1}$. For the next $k-1$ points, at each current intersection point $\mathbf{x_t}$, we sample a direction according to the BRDF, to get a new surface point, $\mathbf{x_{t+1}}$.
Once the initial path is available, then we mutate the path, using point perturbation, many times. To review this mutation technique briefly, consider an index $p \in \{1, \cdots, k\}$. The result of the mutation on the current path $\overline{X_j}$ will be a new path which has the same vertices, except the point with index p, $\mathbf{x_p}$, we denote the index of this new point by p'. That is, if $\overline{X_{j+1}}$ indicates the new path, then

$$\overline{X_j} = (\mathbf{x_s}, \cdots, \mathbf{x_p}, \cdots, \mathbf{x_k}) \ , \ \overline{X_{j+1}} = (\mathbf{x_s}, \cdots, \mathbf{x_{p'}}, \cdots, \mathbf{x_k})$$

, under the assumption that $p \neq k$. If $p = k$, then

$$\overline{X_j} = (\mathbf{x_s}, \cdots, \mathbf{x_p}) \ , \ \overline{X_{j+1}} = (\mathbf{x_s}, \cdots, \mathbf{x_{p'}})$$

Note that, $\mathbf{x_{p'}}$ is chosen such that $V(\mathbf{x_{p'}}, \mathbf{x_{p+1}}) = 1$, where $V(\cdot, \cdot)$ is the visibility function. Therefore, one may summarize this approach with the following steps:

1. Choosing an index $p \in \{1, \cdots, k\}$.
2. Sampling a direction from $\mathbf{x_{p-1}}$ to trace $\mathbf{x_{p'}}$.
3. Checking the visibility condition on $\mathbf{x_{p'}}$ to decide whether accept or deny this point.
4. If $\mathbf{x_{p'}}$ is accepted then, subpath $\mathbf{x_{p'} x_{p+1}}$ contributes in the calculations with the following factor

$$\rho(\mathbf{x_{p-1}} \to \mathbf{x_{p'}} \to \mathbf{x_{p+1}}) \rho(\mathbf{x_{p'}} \to \mathbf{x_{p+1}} \to \mathbf{x_{p+2}}) G(\mathbf{x_{p'}}, \mathbf{x_{p+1}}). \tag{6.7}$$

Recall that, in our implementation, the paths are not connected to the camera in the first place. Therefore, we do not involve any coefficient related the last vertex of the path while computing the acceptance probability of the following cases: $p = k$ and $p+1 = k$. We will take care of this coefficient in the projection step.
So, to compute the acceptance probability, we separate the following three cases:

- $p = k$:

$$a(\overline{X_j} \longrightarrow \overline{X_{j+1}}) = 1 \tag{6.8}$$

- $p+1=k$:

$$a(\overline{X_j} \longrightarrow \overline{X_{j+1}}) = \frac{\rho(\mathbf{x_{k-2}} \rightarrow \mathbf{x_{p'}} \rightarrow \mathbf{x_k})G(\mathbf{x_{p'}}, \mathbf{x_k})}{\rho(\mathbf{x_{k-2}} \rightarrow \mathbf{x_p} \rightarrow \mathbf{x_k})G(\mathbf{x_p}, \mathbf{x_k})} \tag{6.9}$$

- $p+1<k$:

$$a(\overline{X_j} \longrightarrow \overline{X_{j+1}}) = \frac{\rho(\mathbf{x_{p-1}} \rightarrow \mathbf{x_{p'}} \rightarrow \mathbf{x_{p+1}})\rho(\mathbf{x_{p'}} \rightarrow \mathbf{x_{p'+1}} \rightarrow \mathbf{x_{p+2}})G(\mathbf{x_{p'}}, \mathbf{x_{p'+1}})}{\rho(\mathbf{x_{p-1}} \rightarrow \mathbf{x_p} \rightarrow \mathbf{x_{p+1}})\rho(\mathbf{x_p} \rightarrow \mathbf{x_{p+1}} \rightarrow \mathbf{x_{p+2}})G(\mathbf{x_p}, \mathbf{x_{p+1}})}. \tag{6.10}$$

To explain the terms, appearing in equation (6.9) and (6.10), we look back at the mutation strategy used. The acceptance probability is computed in terms of factors that form equation (6.7).

Step 2 and 3. In this part, we add the contribution of each related path to the corresponding image, considering some factors coming from first step of this method.

In the remainder of this section, we discuss factors that are required to be multiplied by the contribution of the paths of length k and $k+1$.

Image of T^kL. Let $\{\overline{X_0}, \cdots, \overline{X_d}\}$ be the sequence of paths generated by repeating mutation operation on paths, starting from $\overline{X_0}$ as an initial path.

Let $\overline{X_j}$ be the current light path, and $\overline{X_{j+1}}$ be the new path as a result of one mutation operation. To reduce the variance, the new path is always accepted. Therefore, the contribution of both old and new paths are scaled with factors $1-pr_{mu}$ and pr_{mu} respectively, where

$$pr_{mu} := \min\{1, |a(\overline{X_j} \rightarrow \overline{X_{j+1}})|\}. \tag{6.11}$$

Given $\overline{X}$, let $(u_{\overline{X}}, v_{\overline{X}})$ denote the corresponding pixel indices in the image plane, I, which are determined by the last vertex of $\overline{X}$. Therefore,

$$I(u_{\overline{X_{j+1}}}, v_{\overline{X_{j+1}}}) = pr_{mu}\, c_{\mathbf{x_k x_e}}\, C(\overline{X_{j+1}}) \tag{6.12}$$

, and

$$I(u_{\overline{X_j}}, v_{\overline{X_j}}) = (1-pr_{mu})\, c_{\mathbf{x_k x_e}}\, C(\overline{X_j}). \tag{6.13}$$

In both equations (6.12) and (6.13), $C(\overline{X}) = \frac{f(\overline{X})}{p(\overline{x}=\overline{X})}$ indicates a function that returns the contribution value of a path, which is sampled according to a probability density function $p(\overline{x})$. $c_{\mathbf{x_k x_e}}$ represents the factor, corresponding to the fact of connecting $\mathbf{x_k}$ to the camera, computed as follows

$$c_{\mathbf{x_k x_e}} = \rho(\mathbf{x_{k-1}} \rightarrow \mathbf{x_k} \rightarrow \mathbf{x_e})G(\mathbf{x_k}, \mathbf{x_e})V(\mathbf{x_k}, \mathbf{x_e}),$$

where $\mathbf{x_e}$ denotes the camera [6].

So what remains is to compute $C(\overline{X_j})$ and $C(\overline{X_{j+1}})$.

[6] Note that $\mathbf{x_k}$ in both equations (6.12) and (6.13) represents the last vertex of $\overline{X_{j+1}}$ and $\overline{X_j}$ respectively. However, these vertices may not be the same point.

In our implementation, each vertex of $\overline{X_0}$ is sampled using importance sampling. Namely, vertex $\mathbf{x_{t+1}}$ is traced from $\mathbf{x_t}$ according to the BRDF at this point. Therefore, the contribution of $\overline{X_0}$ is

$$C(\overline{X_0}) = \frac{\Pi_{t=2}\rho(\mathbf{x_{t-1}} \to \mathbf{x_t} \to \mathbf{x_{t+1}})}{\Pi_{t=2}\rho(\mathbf{x_{t-1}} \to \mathbf{x_t} \to \mathbf{x_{t+1}})} = 1. \tag{6.14}$$

Equation (6.14) is true since at each surface point, light reflects with the probability equal to the BRDF value. Similarly, $C(\overline{X_j}) = C(\overline{X_{j+1}}) = 1$.

Image of $T^{k+1}L$. Constructing the second image requires paths of length $k+1$. Since in this method we use the same paths that already contributed to the first image, by sampling a new vertex $\mathbf{x_{k+1}}$. Similar to other vertices along the given path $\overline{X_j}$, we sample a direction at vertex $\mathbf{x_k}$ to trace $\mathbf{x_{k+1}}$. We denote this new path with $\overline{Y_j}$, Figure .

Similar to equations (6.12) and (6.13), the following values are added to the second image

$$I(u_{\overline{Y_{j+1}}}, v_{\overline{Y_{j+1}}}) = pr_{mu}\, c_{\mathbf{x_{k+1}x_e}}\, C(\overline{Y_{j+1}}), \tag{6.15}$$

, and

$$I(u_{\overline{Y_j}}, v_{\overline{Y_j}}) = (1 - pr_{mu})\, c_{\mathbf{x_{k+1}x_e}}\, C(\overline{Y_j}) \tag{6.16}$$

, where

$$c_{\mathbf{x_{k+1}x_e}} = \rho(\mathbf{x_k} \to \mathbf{x_{k+1}} \to \mathbf{x_e}) G(\mathbf{x_{k+1}}, \mathbf{x_e}) V(\mathbf{x_{k+1}}, \mathbf{x_e}).$$

To compute $C(\overline{Y_j})$ ($C(\overline{Y_{j+1}})$) from $C(\overline{X_j})$ ($C(\overline{X_{j+1}})$), it is supposed that light is reflected at $\mathbf{x_k}$. To consider this fact, it is required to scale these values with this probability. In the specific case of Lambertian scenes, this value is equal to the albedo, so

$$C(\overline{Y_j}) = C(\overline{X_j})\rho_d(\mathbf{x_k}) = \rho_d(\mathbf{x_k}), \tag{6.17}$$

similarly

$$C(\overline{Y_{j+1}}) = C(\overline{X_{j+1}})\rho_d(\mathbf{x_k}) = \rho_d(\mathbf{x_k}). \tag{6.18}$$

Generating images using multiple initial paths.

Image of T^kL. In this case, we repeat the three steps, expressed earlier for each initial path, for each initial path individually.

To ensure ergodicity, we need to sample all initial paths, $\{\overline{X}_0^{(i)}\}_i$, using a common probability distribution. This distribution needs to be proportional to the image on the screen. To do this, we sample points uniformly in the scene and then form paths in the way that any two consecutive vertices are not on the same surface i.e. subpath $\mathbf{x_t x_{t+1}}$ belongs to $\overline{X}_0^{(i)}$ if

$$V(\mathbf{x_t}, \mathbf{x_{t+1}}) = 1 \quad , \quad < \mathbf{n_{x_t}}, \mathbf{n_{x_{t+1}}} >< 0.$$

Then, each initial path is mutated in parallel and contributes to the same image using equations (6.13), (6.12), (6.16) and (6.15).

The only difference is the value of $C(\overline{X}_0^{(i)})$. Since each initial path is built explicitly, so $C(\overline{X}_0^{(i)})$ is the multiplication of point–to–point form factors, i.e.

$$C(\overline{X}_0^{(i)}) = \Pi_{t=1}^{t=k-1}\rho(\mathbf{x_{t-1}} \rightarrow \mathbf{x_t} \rightarrow \mathbf{x_{t+1}})G(\mathbf{x_t}, \mathbf{x_{t+1}}). \tag{6.19}$$

Since each $\overline{X}_{j+1}^{(i)}$ is obtained by mutating $\overline{X}_j^{(i)}$, and variable pr_{mu} has already accounted for the modification applied on $\overline{X}_j^{(i)}$, so

$$C(\overline{X}_j^{(i)}) = C(\overline{X}_{j+1}^{(i)}) = C(\overline{X}_0^{(i)}). \tag{6.20}$$

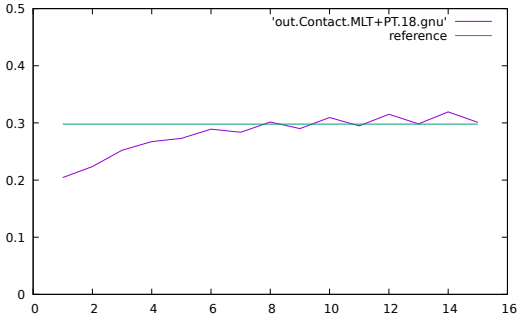
The values of $C(\overline{Y}_j^{(i)})$ ($C(\overline{Y}_{j+1}^{(i)})$) are computed using its connection with $C(\overline{X}_j^{(i)})$ ($C(\overline{X}_{j+1}^{(i)})$):

$$C(\overline{Y}_j^{(i)}) = C(\overline{X}_j^{(i)})\rho_d(\mathbf{x_k}). \tag{6.21}$$

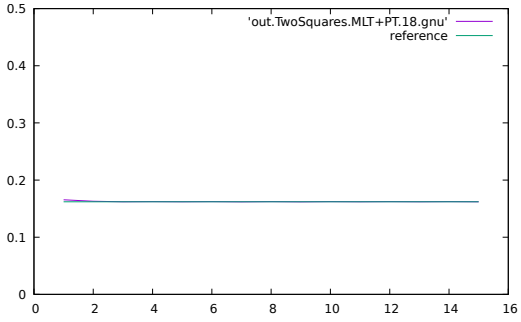
6.3 Results

To obtain the dominant eigenvalue, we experimented the power iteration method on our test scenes. For each scene with a static light source, the images of pairs of powers of T have been computed. Then, the graph of the ratio of images T^kL and $T^{k+1}L$, for $k = 2, \cdots, 15$, versus k are plotted. The following table contains the parameters used to render the pairs of images.
Since the final results of the experiment are mainly graphs, we only bring a pair of images of a few scenes, Figures 6.2, 6.3, and 6.4.

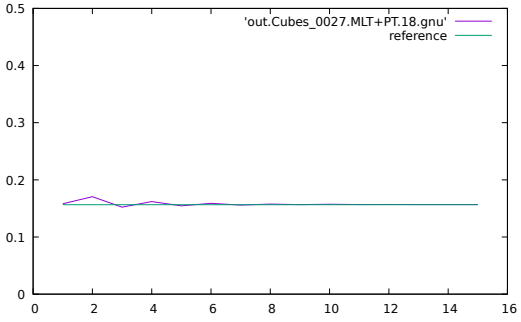
Results on Different Scenes Using Single Seed			
Scene	Number of Initial Paths	Number of Mutations	Figure
"Contact"	1	10^4	6.6
Results on Different Scenes Using Multiple Seeds			
Scene	Number of Initial Paths	Number of Mutations	Figure
"Contact"	10^5	10^5	6.1a
"CornellBox.ifs"	10^5	10^5	6.1f
"TwoSquaresNoSymmetry"	10^5	10^5	6.1e
"TwoSquares"	10^5	10^5	6.1b
"Cubes027"	10^5	10^5	6.1c
"Cubes0343"	10^5	10^5	6.1d



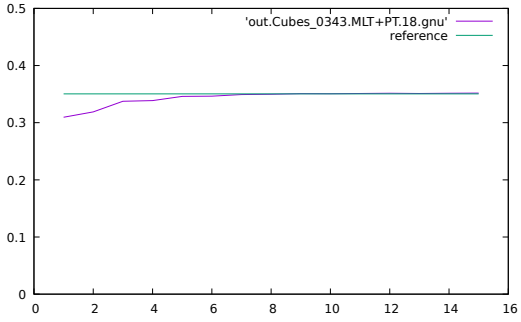
(a) $\lambda_1 = -0.29799, \lambda_2 = 0.29794$



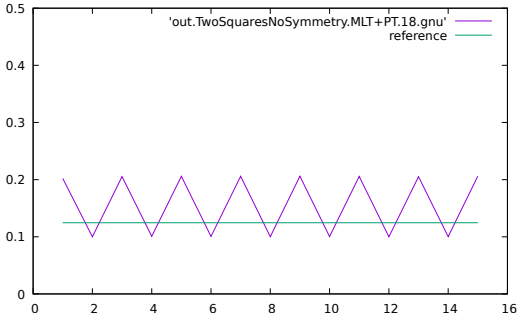
(b) $\lambda_1 = 0.16203, \lambda_2 = -0.16203$



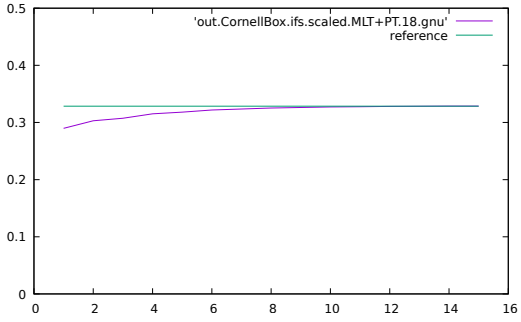
(c) $\lambda_1 = 0.15655, \lambda_2 = 0.09947$



(d) $\lambda_1 = 0.35055, \lambda_2 = 0.35055$



(e) $\lambda_1 = 0.12448, \lambda_2 = -0.12448$



(f) $\lambda_1 = 0.32856, \lambda_2 = 0.26386$

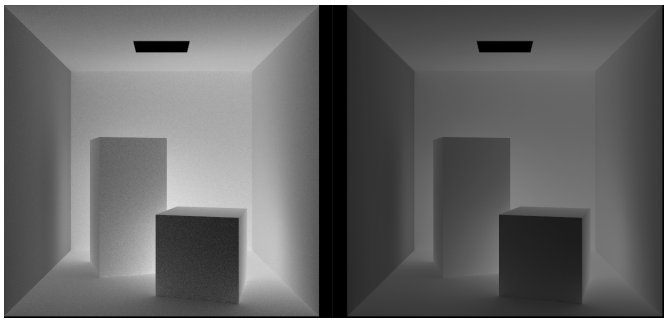


Figure 6.2: Noise–free images of $T^{16}L$ and $T^{17}L$ of the "CornellBox.ifs" scene, computed using multiple initial paths.

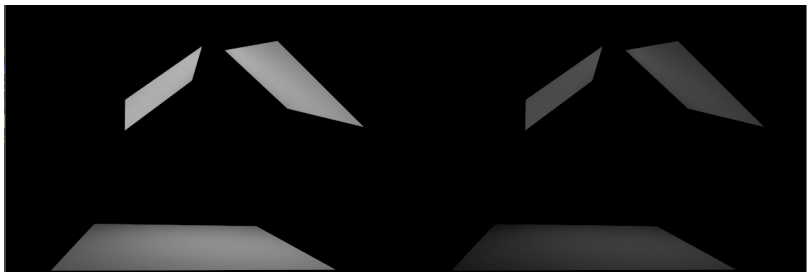


Figure 6.3: Noise–free images of $T^{9}L$ and $T^{10}L$ of the "ThreeSquaresNoSymmetry" scene, computed using multiple initial paths.

6.4 Discussion

In this section, we discuss two subjects related to the theory and implementation parts respectively.

6.4.1 Power iteration method

To be able to analyze the power iteration algorithm for the light transport operator, the compactness property is required to limit the spectrum set to the set of eigenvalues. We also make an extra assumption that the set of eigenvectors span $\mathbb{H}$.
So, every light distribution L can be written in terms of eigenvectors i.e.

$$L = \sum_{i=1}^{\infty} c_i \Lambda_i.$$

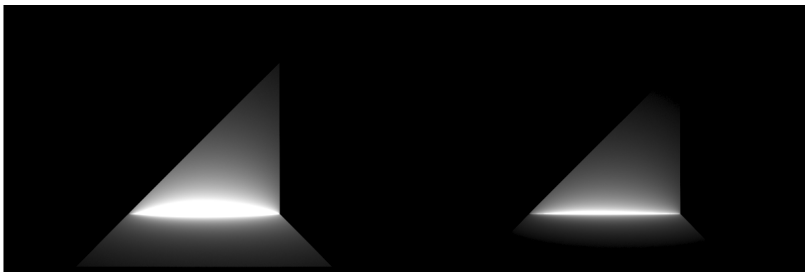


Figure 6.4: Noise–free images of T^9L and $T^{10}L$ of the "Contact" scene, computed using an initial path.

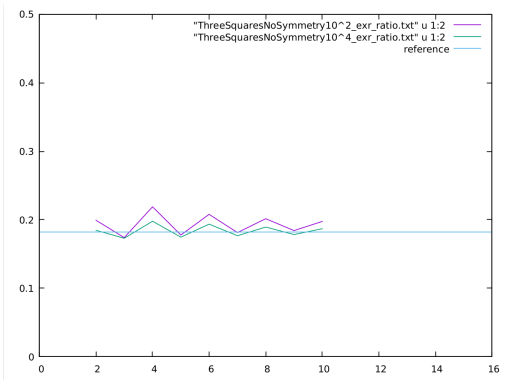


Figure 6.5: Comparison of the method's performance on "ThreeSquaresNoSymmetry" by increasing the number of initial paths from 100 to 10000.

$$\begin{aligned} b_{k+1} :&= Tb_k = T^kL = T^k(\sum_{i=1}^{\infty} c_i\Lambda_i) = \sum_{i=1}^{\infty} c_i\lambda_i^k\Lambda_i \\ &= \lambda_i^k[c_1\Lambda_1 + \sum_{i\neq 1}^{\infty} c_i(\frac{\lambda_i}{\lambda_1})\Lambda_i]. \end{aligned} \tag{6.22}$$

Equation (6.22) indicates that that sequence (b_k) converges to Λ_1 if $|\frac{\lambda_i}{\lambda_1}| < 1$. Therefore, if $\lambda_1 = -\lambda_2$, then (b_k) does not converge to Λ_1. Depending on index k, (b_k) *flip–flops* in the direction of two vectors $\Lambda_1+\Lambda_2$ and $\Lambda_1-\Lambda_2$. This reason may explain the method's behavior performed on the "TwoSquaresNoSymmetry" scene. However, this event does not happen for "TwoSquares". This scene is symmetric and if the light source lies in the middle of two surfaces, it may cause orthogonality of L to one of the eigenvectors Λ_1 or Λ_2. Therefore, the corresponding coefficient (c_0 or c_2) is zero, which avoids the method's failure.

Remark 6.4.1 *Note that the compactness and the completeness properties are sufficient conditions to explain how the method works. Therefore, the method may still work for the scenes without these properties, such as the "Cornell-*

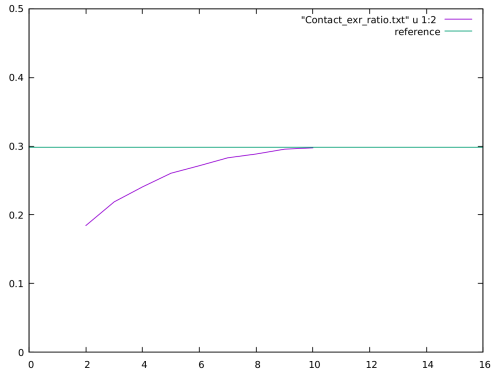


Figure 6.6: Performing the method on the "Contact" scene with a single initial path.

Box.ifs" scene.

6.4.2 Sampling strategies for long light paths

The problem of choosing a sampling strategy for long light paths is the following:

1. Sampling a path explicitly, sampling vertices uniformly then connecting them, is costly because of visibility calculation, in particular, in scenes where the couples of points have a high probability of not seeing each other.

2. In the case following importance sampling, it is hard to get a full light path of length k in open scenes, because the geometry is easily be missed (e.g. the "TwoSquares" scene has this problem).

It is generally better to try both strategies, assuming that their respective probability densities are correctly set. This the reason that we have used both strategies, using the importance sampling technique for a single initial path, and building multiple initial paths explicitly.
As it is expected, using more that one initial light path ensures ergodicity. Considering the "ThreeSquaresNoSymmetry" scene, the curves of the eigenvalue estimation in Figure 6.5 illustrates more accurate estimation versus increasing the number of initial light paths. However, for a very specific scene among our test scenes, "Contact" scene, the values computed by equation (6.19) become very small, so it will increase the numerical instability, Figure 6.1a. Since this scene is very simple, using even one initial path still estimates the dominant eigenvalue, Figure 6.6.

Chapter 7

Calculation of eigenvalues via regularized determinant

It is known that the Fredholm theory is a useful tool in several fields such as operator theory, mathematical physics and quantum field theory. Therefore, the numerical methods to evaluate the so–called Fredholm determinant is very critical. But choosing a numerical method to fulfill the task is highly dependent on the properties of the kernel, such as continuity, singularity, and the domain of the kernel function. For an operator defined on a Hilbert space, based on the assumption of Hilbert–Schmidt property of the operator, the eigenvalues of the operator are the reciprocals of a determinant, called *the regularized determinant*, which is not exactly the Fredholm determinant but relates to it. Similar to the Fredholm determinant, this determinant is also expressed in terms of some specific coefficients, denoted by c_n and p_n in this chapter. To evaluate p_n coefficients for the light transport operator, we derive an interesting equation in terms of path integrals. Therefore, it is possible to compute the determinant using the Monte Carlo algorithms. Despite of exploring all the connections between these coefficients and eigenvalues of the light transport operator, our implementation suffers from a numerical obstacle.
In Section 7.1, we recall the original definition of the Fredholm determinant, and then we address briefly other similar constructions of the determinant for the specific case of the Hilbert–Schmidt operator. In Section 7.2, we explain how to compute the regularized determinant of light transport. In Section 7.3, c_n and p_n coefficients with the eigenvalues of the light transport operator evaluated for our test scenes are provided. We finish the chapter by a discussion on the numerical results, Section 7.4.

7.1 Fredholm determinant and its generalizations

It was Fredholm himself who introduced the determinant defined by (7.2) for the first time, and then the subject started to grow and it has been generalized

to the operators over infinite–dimensional spaces. In addition to the great interest of the theoretical point of view, the numerical evaluation of the Fredholm determinant has a great significance. Regardless of the theoretical development, there is no general framework for numerical methods to compute the Fredholm determinant since implementations depend on the kernel's property. For example, *Nystrom–type* method seems fit to the case when the kernel is smooth, or some other ways are suitable when the operator is self–adjoint, for example Gaussian unitary ensemble (GUE) problem, see [Gau61] and [Meh04].
When the operator is trace class, there is a projection–based method. This method requires an orthogonal basis which projects the Hilbert space to some finite subspaces, check Bornemann et al.'s paper [Bor10].
In Section 7.1.1, we start with the Fredholm integral equation of the second kind under the assumption that the unknown solution function is an integrable function on (a, b). In Section 7.1.2, we list equivalent infinite–dimensional determinants that are applicable when the operator is trace class. In Section 7.1.3, we review the regularized determinant, which is used in the place of Fredholm operator when the operator is Hilbert–Schmidt.

7.1.1 The original definition

Consider the following equation

$$\phi(x) = f(x) + \mu \int_a^b \kappa(x, y)\phi(y)dy \ , \ x \in (a, b) \tag{7.1}$$

, which is called a *Fredholm equation* of the second kind when the kernel is bounded and continuous. Fredholm showed that equation (7.1) is uniquely solvable if and only if $d(\mu)$ with the following expression

$$d(\mu) = \det(I+\mu T) = \sum_{n=0}^{\infty} \frac{\mu^n}{n!} \int_{(a,b)^n} \det \begin{bmatrix} \kappa(t_1, t_1) & \kappa(t_1, t_2) & \cdots & \kappa(t_1, t_n) \\ \kappa(t_2, t_1) & \kappa(t_2, t_2) & \cdots & \kappa(t_2, t_n) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(t_n, t_1) & \kappa(t_n, t_2) & \cdots & \kappa(t_n, t_n) \end{bmatrix} dt_1 \cdots dt_n \tag{7.2}$$

be a non–zero. $d(\mu)$ is known as *Fredholm determinant*. Searching for the solution of equation (7.1), $\phi(x)$, this unknown function is assumed to be Riemann integrable. One may also make extra assumptions on $f(x)$ and κ such as continuity on the interval (a, b) and on the square $(a, b) \times (a, b)$ respectively.
Define c_n coefficients as

$$c_n = \int_{(a,b)^n} \det \begin{bmatrix} \kappa(t_1, t_1) & \kappa(t_1, t_2) & \cdots & \kappa(t_1, t_n) \\ \kappa(t_2, t_1) & \kappa(t_2, t_2) & \cdots & \kappa(t_2, t_n) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(t_n, t_1) & \kappa(t_n, t_2) & \cdots & \kappa(t_n, t_n) \end{bmatrix} dt_1 \cdots dt_n, \tag{7.3}$$

which reformulates equation (7.2) into

$$d(\mu) = \sum_{n=0}^{\infty} \frac{\mu^n}{n!} c_n.$$

Let $B_0(x, y) := \kappa(x, y)$, and $c_0 = 1$. For $n \geq 1$ define $B_n(x, y)$ as a recursive linear combination of iterated kernel

$$B_n(x, y) = c_n \kappa(x, y) - n \int_a^b \kappa(x, y) B_{n-1}(y, z) dy, \tag{7.4}$$

be a family of functions defined in terms of c_n. With this setting, it can be shown [Zem12]

$$c_n = \int_a^b B_{n-1}(x, x) dx. \tag{7.5}$$

7.1.2 Fredholm determinant of a trace class operator

Let $\mathbb{H}$ be a Hilbert space and $T : \mathbb{H} \longrightarrow \mathbb{H}$ be a trace class operator. One may find several equivalent reformulations of

$$d(\mu) = \det(I + \mu T),$$

for $\mu \in \mathbb{C}(\neq 0)$. We address some of these constructions as follows:

- Gohberg et al. defined $d(\mu)$ as an infinite product [GK78], namely

$$d(\mu) = \prod_{n=1}^{N(T)} (I + \mu \lambda_n), \tag{7.6}$$

  where $(\lambda_n)_{n=1}^{N(T)}$, counting multiplicity, is either finite or infinite set of eigenvalues. The infinite product on the right side of equation (7.6) is locally uniformly convergent. He called this determinant as *characteristic determinant*, and showed that it is bounded [GK78] i.e.

$$|d(\mu)| \leq \prod_{j=1}^{\infty} (1 + |\mu| s_j(T)) \leq \exp^{|\mu| \|T\|_{J_1}}.$$

- Consider (T_n) as a sequence of finite–rank operators that converges to T, in the trace norm, $\|\cdot\|_{J_1}$ [1]. Then, the sequence of $(\det(I + \mu T_n|_{ran(T_n)}))$ converges locally uniform to $d(\mu)$ [GGK13]. Note that the definition of $d(\mu)$ as this limit is independent of the choice of (T_n).

[1] This norm is defined as the sum of singular values of the operator.

- Dunford et Schwartz $d(\mu)$ in terms of the trace of powers of T, $tr(T^n)$,

$$\det(I+\mu T) = \exp(\sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} \mu^n tr(T^n)), \tag{7.7}$$

  which converges in $\{\mu \in \ | \ |\mu| < \frac{1}{|\lambda_1|}\}$. This definition comes from the theorem called *Plemelj-Smithies formulas* [GGK96]. Note that using analytic continuation, $d(\mu)$ can extended for all $\mu \in \mathbb{C}$.

7.1.3 The regularized determinant for a Hilbert–Schmidt operator

Let $\mathbb{H}$ be a Hilbert space and $T : \mathbb{H} \longrightarrow \mathbb{H}$ be a Hilbert–Schmidt operator. In comparison to the trace class property, $\sum_{n=1}^{N(T)} \lambda_n^2$ is finite but not necessarily $\sum_{n=1}^{N(T)} \lambda_n$. Therefore, the definition of $d(\mu)$ by equation (7.6) is not meaningful, since it may not converge. In replace, one may define $d_2(\mu)$ as follows

$$d_2(\mu) := \det_2(I+\mu T) := \prod_{n=1}^{N(T)} (1+\mu\lambda_n) \exp^{-\mu\lambda_n} .$$

This definition introduced by Hilbert (1904) and Carleman (1921), and it is also known as *Hilbert–Carleman determinant*. Originally, $\det_2(I+T)$ is defined as $\det[(I+T)\exp(-T)]$ [GGK96]. For this case, equation (7.7) turns to

$$\det_2(I+\mu T) = \exp(\sum_{n=2}^{\infty} \frac{(-1)^{n+1}}{n} \mu^n tr(T^n)),$$

[2].

For a specific case, when $\mathbb{H}$ is $\mathcal{L}_2(a,b)$, there exists an expression for $\det_2(I+\mu T)$ similar to the one for $d(\mu)$ in equation (7.2). However in this case $\kappa(x,x)$ is set to zero while computing the determinant, that is

$$\sum_{n=0}^{\infty} \frac{\mu^n}{n!} \int_{(a,b)^n} \det \begin{bmatrix} 0 & \kappa(t_1,t_2) & \cdots & \kappa(t_1,t_n) \\ \kappa(t_2,t_1) & 0 & \cdots & \kappa(t_2,t_n) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(t_n,t_1) & \kappa(t_n,t_2) & \cdots & 0 \end{bmatrix} dt_1 \cdots dt_n. \tag{7.8}$$

Hilbert proved that the expression above converges to $\det_2(I+\mu T)$ [Hil12].

7.2 Calculation of regularized determinant of light transport

To be able to reach eigenvalues of light transport using equations developed in Fredholm theory, it is essential to choose an appropriate equation based on the

[2]Note that this series starts with index 2, since the trace of T^n for $n \geq 2$ is finite i.e. all powers of T^n for $n \geq 2$ are trace class operators.

property of the operator. Considering the light transport operator, acting on a Hilbert space, we can only use equation (7.8) in a very specific situation, when the operator is Hilbert–Schmidt. This property holds for the Lambertian light transport operator when we remove the edges from the scene.

7.2.1 Computing c_n coefficients of the regularized determinant

To evaluate $d_2(\mu)$ using equation (7.8) numerically, it is required to evaluate the interior integral, which is c_n coefficients, defined by equation (7.3). To calculate these coefficients, we only consider the Lambertian materials. Recall the kernel expression of the light transport operator, equation (3.35). Since the material model is Lambertian, then

$$\forall \mathbf{x_i}, \mathbf{x_j} \in S : \kappa(\mathbf{x_i}, \mathbf{x_j}) = \frac{\rho_d(\mathbf{x_i}) \cos\theta \cos\theta'}{\pi r^2_{\mathbf{x_i x_j}}} V(\mathbf{x}_i, \mathbf{x}_j). \tag{7.9}$$

To compute c_n coefficient, given n:

1. n number of surface points are sampled in the scene,
2. using equation (7.9), the matrix containing values of the kernel evaluated at a pair of surface points, selected in the previous step, is constructed,
3. the determinant of the matrix, which is constructed in the previous step, is computed.

7.2.2 The integral of cyclic light paths

In this section we define a quantity, p_n, which brings a very interesting interpretation for general scenes.
Under the assumption of Hilbert–Schmidt property for the light transport operator, we derive a new equation, indicating the relationship between c_n and p_n coefficients.
We define $\Pi_n(\mathbf{x}) : S \longrightarrow S$ as a path integral given by the following formula

$$\forall \mathbf{x} \in S : \Pi_n(\mathbf{x}) := \int_{S^{n-1}} \kappa(\mathbf{x}, \mathbf{x_2}) \cdots \kappa(\mathbf{x_{n-1}}, \mathbf{x_n}) \kappa(\mathbf{x_n}, \mathbf{x}) d\mathbf{x_2} \cdots d\mathbf{x_n} \tag{7.10}$$

in which $(\mathbf{x_i})_{i=2}^n \subset S$. In Lambertian scenes, this quantity can be interpreted as the integral of the amount of energy returning to a surface point $\mathbf{x}$, after propagation along all paths starting from vertex $\mathbf{x}$ and bouncing $n-1$ times in the scene. Then define

$$p_n := \int_S \Pi_n(\mathbf{x}) d\mathbf{x} \tag{7.11}$$

, which evaluates the density of all cyclic light paths [3] with length n in the scene.

The relationship between c_n and p_n coefficients. Starting from equation (7.4), we have

$$B_n(\mathbf{x},\mathbf{x}) = -n\int_S \kappa(\mathbf{x},\mathbf{s})B_{n-1}(\mathbf{s},\mathbf{x})d\mathbf{s} \tag{7.12}$$

, in which we put $\kappa(\mathbf{x},\mathbf{x}) = 0$. One may use equation (7.4) to expand $B_{n-1}(\mathbf{s},\mathbf{x})$, so

$$\begin{aligned} B_n(\mathbf{x},\mathbf{x}) &= -nc_{n-1}\int_S \kappa(\mathbf{x},\mathbf{s})\kappa(\mathbf{s},\mathbf{x})d\mathbf{s} + n(n+1)\int_S\int_S \kappa(\mathbf{x},\mathbf{s})\kappa(\mathbf{s},\mathbf{t})B_{n-2}(\mathbf{t},\mathbf{x})d\mathbf{t}d\mathbf{s} \\ &= -nc_{n-1}\Pi_2(\mathbf{x}) + n(n+1)\int_S\int_S \kappa(\mathbf{x},\mathbf{s})\kappa(\mathbf{s},\mathbf{t})B_{n-2}(\mathbf{t},\mathbf{x})d\mathbf{t}d\mathbf{s}, \end{aligned} \tag{7.13}$$

if we continue applying equation (7.4), and using $B_0(\mathbf{x},\mathbf{x}) = \kappa(\mathbf{x},\mathbf{x})$, then

$$B_n(\mathbf{x},\mathbf{x}) = \sum_{k=1}^{n}(-1)^k\,\Pi_{k+1}(\mathbf{x})\,\frac{n!}{(n-k)!}\,c_{n-k}. \tag{7.14}$$

Now plugging the previous expression of $B_n(\mathbf{x},\mathbf{x})$ in equation (7.5) yields

$$\begin{aligned} c_{n+1} &= \int_{\mathbf{x}\in S}(\sum_{k=1}^{n}(-1)^k\,\Pi_{k+1}(\mathbf{x})\,\frac{n!}{(n-k)!}\,c_{n-k})d\mathbf{x} \\ &= \sum_{k=1}^{n}(-1)^k p_{k+1}\,c_{n-k}\,\frac{n!}{(n-k)!}. \end{aligned} \tag{7.15}$$

Therefore, equation (7.15) is another way to compute c_n coefficients from the integrals of cyclic light paths.

7.3 Results

In this section, we provide the values of c_n, computed using equation (7.3) in which the diagonal of the matrix is set to zero, and p_n coefficients, using equation (7.11). The eigenvalues of the light transport operator are also calculated solving for the roots of equation (7.2).

c_n coefficients. A number of surface points are sampled in the scene. To construct the matrix in equation (7.8), it is required to evaluate the kernel at pairs of sampled points. To do this task, we construct the matrix coefficients in two ways:

[3] By cyclic paths, we mean the light paths that have the same endpoints. For example, $(\mathbf{x},\mathbf{x_2},\cdots,\mathbf{x_n},\mathbf{x})$ is a cyclic light path. Since the sequence of points starting from $\mathbf{x}$ returns to $\mathbf{x}$, as the final vertex.

- **Method 1.** Using equation (7.9) directly.
- **Method 2.** To decrease the numerical instability caused by the infinitely large values of the kernel evaluations for pairs of points that are located arbitrarily close, the kernel values, which are point–to–point form factors, can be approximated by an analytic formula, which is the point–to–disk form factors [Nar88]. Using this approximation, the kernel values are bounded.

Table 7.1: Computing c_n coefficients and eigenvalues of the light transport operator of the "Contact" scene using both methods.

	$\mathbf{c_n}$		**Eigenvalues**		**Reference Eigenvalues**
	Method 1	**Method 2**	**Method 1**	**Method 2**	
$n=0$	1.000000000000	1.000000000000	1.2636945069	2.1786475943	0.2979420180
$n=1$	0.000000000000	0.000000000000	1.2636945069	2.1786475943	0.2979950679
$n=2$	−3.325305689225	−9.654203629623	0.2983084146	0.2190373153	0.2873354311
$n=3$	0.000000000000	−0.000000000000	0.2983084146	0.2190373153	0.2873464849
$n=4$	2.717504977530	9.228603131626	0.2983084146	0.1421630612	0.2793885915
$n=5$	0.000000000000	−0.000000000000	0.2983084146	0.1421630589	0.2793980238
$n=6$	−9.536002701504	−6.758301880979	0.0759941015	0.1421630601	0.2727184421
$n=7$	−0.000000000001	−0.000000000000	0.0759941015	0.1421630597	0.2727202517
$n=8$	2.944601263605	3.750401401166	0.0000000000	0.0000000000	0.2663676942

Table 7.2: Computing c_n coefficients and eigenvalues of the light transport operator of the "TwoSquares" scene using both methods.

	$\mathbf{c_n}$		**Eigenvalues**		**Reference Eigenvalues**
	Method 1	**Method 2**	**Method 1**	**Method 2**	
$n=0$	1.000000000000	1.000000000000	0.1621222881	0.1621751199	0.1620320938
$n=1$	0.000000000000	0.000000000000	0.1621222881	0.1621751199	0.1620320938
$n=2$	−0.057301200393	−0.057339884526	0.0334121462	0.0335610978	0.0333833732
$n=3$	−0.000000000000	−0.000000000000	0.0334121460	0.0331730098	0.0333816687
$n=4$	0.001530773040	0.001532971797	0.0334121461	0.0335610912	0.0333816687
$n=5$	0.000000000000	−0.000000000000	0.0334121461	0.0331730163	0.0333833732
$n=6$	−0.000029844986	−0.000029730419	0.0092171512	0.0088483792	0.0089627591
$n=7$	−0.000000000000	−0.000000000000	0.0092171512	0.0088483792	0.0089627591
$n=8$	0.000000203247	0.000000202278	0.0053963737	0.0057280355	0.0055606790

p_n coefficients. To compute p_n coefficients using the Monte Carlo algorithm, set of N points are sampled uniformly in the scene for the starting points of the cyclic paths. By doing the change of variable from point surfaces to the projected solid angles in equation (7.10), and growing the path according to the cosine–weighted BRDF distribution, the estimation of the p_n values simplifies to

$$\begin{aligned} \widehat{p_n} &= \frac{A}{N}\sum_{i=1}^{N}\frac{\Pi_n(\mathbf{x_1^i})}{P(\Pi_n(\mathbf{x_1^i}))} \\ &= \frac{A}{N}\sum_{i=1}^{N}\kappa(\mathbf{x_n},\mathbf{x_1^i}) \end{aligned} \tag{7.16}$$

Table 7.3: Computing c_n coefficients and eigenvalues of the light transport operator of the "Cubes0027" scene using both methods.

	c_n		Eigenvalues		Reference Eigenvalues
	Method 1	**Method 2**	**Method 1**	**Method 2**	
$n = 0$	1.000000000000	1.000000000000	0.3511334245	0.2775410936	0.1565540933
$n = 1$	0.000000000000	0.000000000000	0.3511334244	0.2459133874	0.0994781132
$n = 2$	−0.276343122240	−0.278730786957	0.3475862150	0.2545465924	0.0953148799
$n = 3$	0.007329412940	0.007540001169	0.3475862150	0.2545465904	0.0926948370
$n = 4$	0.228419718834	0.222793371663	0.2820318993	0.2334714561	0.0915003017
$n = 5$	−0.017332102881	−0.018117951668	0.2820318993	0.2334714541	0.0895453232
$n = 6$	−0.263808779790	−0.272502148583	0.2826110841	0.1578578081	0.0879342769
$n = 7$	0.048396818600	0.052066549738	0.2826110846	0.1578578084	0.0737271983
$n = 8$	3.825454796271	0.334991712935	0.0611937985	0.1878533838	0.0722159153

Table 7.4: Computing c_n coefficients and eigenvalues of the light transport operator of the "Cubes0343" scene using both methods.

	c_n		Eigenvalues		Reference Eigenvalues
	Method 1	**Method 2**	**Method 1**	**Method 2**	
$n = 0$	1.000000000000	1.000000000000	1.0363432047	1.2415905679	0.3505501843
$n = 1$	0.000000000000	0.000000000000	1.0363431855	1.0166310190	0.2476734734
$n = 2$	−5.205916661381	−5.182023579929	0.9019082987	1.0166310054	0.2469152473
$n = 3$	0.295064978695	0.306798943340	0.9019082987	1.0912010202	0.2465260685
$n = 4$	80.258544188411	78.925546806727	0.8764974121	1.0912010215	0.1972150667
$n = 5$	−15.308959730771	−12.272793542900	0.8764974080	1.0264440391	0.1959398298
$n = 6$	−1634.094540949442	−2216.335405989682	0.9897568270	1.0264440279	0.1943451506
$n = 7$	834.139205163719	1857.323790146492	0.9897568238	0.7134296635	0.1768256458
$n = 8$	26463.081001032875	45867.097969776034	0.0652208256	0.0540458117	0.1752493375

Table 7.5: Computing c_n coefficients and eigenvalues of the light transport operator of the "TwoSquaresNoSymmetry" scene using both methods.

	c_n		Eigenvalues		Reference Eigenvalues
	Method 1	**Method 2**	**Method 1**	**Method 2**	
$n = 0$	1.000000000000	1.000000000000	0.1244746359	0.1245973535	0.1244895625
$n = 1$	0.000000000000	0.000000000000	0.1244746359	0.1245973535	0.1244895625
$n = 2$	−0.033250376741	−0.033300067914	0.0271942234	0.0270582658	0.0271874040
$n = 3$	−0.000000000000	−0.000000000000	0.0271942234	0.0270582658	0.0271874040
$n = 4$	0.000428066864	0.000426715049	0.0183179597	0.0184898859	0.0184309427
$n = 5$	−0.000000000000	0.000000000000	0.0183179597	0.0184898859	0.0184309427
$n = 6$	−0.000003452045	−0.000003435042	0.0074953708	0.0049126138	0.0050972532
$n = 7$	−0.000000000000	−0.000000000000	0.0074953708	0.0049126137	0.0050972532
$n = 8$	0.000000008709	0.000000008619	0.0000000000	0.0046015566	0.0044293806

Table 7.6: Computing c_n coefficients and eigenvalues of the light transport operator of the "CornellBox.ifs.scaled" scene using both methods.

	$\mathbf{c_n}$		Eigenvalues		Reference Eigenvalues
	Method 1	**Method 2**	**Method 1**	**Method 2**	
$n=0$	1.000000000000	1.000000000000	4.8229403402	2.8585052364	0.3285623066
$n=1$	0.000000000000	0.000000000000	4.8220427448	2.8553865461	0.2638649926
$n=2$	−47.505257342098	−23.374798908970	0.7132813219	1.8569583077	0.2568015504
$n=3$	0.160890070692	0.146119099663	0.6866522622	1.8577259732	0.2249185119
$n=4$	277.928801862918	697.021844726002	0.4669344426	0.2966750835	0.2090403967
$n=5$	−18.004937513940	−10.328695760862	0.4669344495	0.2739748424	0.2051277453
$n=6$	−690.642124122663	−1601.926746953372	0.3019531367	0.3120472092	0.1906039872
$n=7$	17668.392941887538	257.308340793193	0.3019531367	0.3120472094	0.1820359231
$n=8$	−25010.917556261225	9859.270460478130	0.4156391272	0.3023588298	0.1793384107

, A indicates the scene area, and P denotes the probability density function to sample cyclic light paths.
Similar to the c_n coefficients, in Method 1, we use directly kernel values at pairs of points to compute p_n values, in Method 2, the estimation of these values are used in the calculation.
Note that due to the relationship between c_n and p_n coefficients by equation (7.15), it is also possible to compute c_n from p_n. For all our test scenes, the column titled as "$\mathbf{c_n}$ **from** $\mathbf{p_n}$" contains these values.

Table 7.7: Computing p_n, c_n coefficients, and eigenvalues of the light transport operator of the "Contact" scene using both methods.

	$\mathbf{c_n}$ **from** $\mathbf{p_n}$		$\mathbf{p_n}$		Eigenvalues	
	Method 1	**Method 2**	**Method 1**	**Method 2**	**Method 1**	**Method 2**
$n=0$	1.000000000000	1.000000000000	1.000000000000	1.000000000000	1.3907698005	1.3907125085
$n=1$	0.000000000000	0.000000000000	0.000000000000	0.000000000000	1.3907698020	1.3907125101
$n=2$	−8.577855369742	−8.577160274253	8.577855369742	8.577160274253	1.3334583402	1.3334032271
$n=3$	0.000000000000	0.000000000000	0.000000000000	0.000000000000	1.3334583402	1.3334032271
$n=4$	218.128309592543	218.092748371455	0.435083106684	0.435047789866	1.3334583788	1.3334032657
$n=5$	0.000000000000	0.000000000000	0.000000000000	0.000000000000	1.3334583399	1.3334032268
$n=6$	−9134.798360298048	−9132.551246765881	0.027980828495	0.027978640276	1.1509198836	1.1508717629
$n=7$	0.000000000000	0.000000000000	0.000000000000	0.000000000000	1.1509198847	1.1508717640
$n=8$	529163.620711958967	528989.027907695854	0.002032652401	0.002032504455	1.1509198828	1.1508717621

Table 7.8: Computing p_n, c_n coefficients, and eigenvalues of the light transport operator of the "TwoSquares" scene using both methods.

	$\mathbf{c_n}$ **from** $\mathbf{p_n}$		$\mathbf{p_n}$		Eigenvalues	
	Method 1	**Method 2**	**Method 1**	**Method 2**	**Method 1**	**Method 2**
$n=0$	1.000000000000	1.000000000000	1.000000000000	1.000000000000	0.1692882029	0.1527052551
$n=1$	0.000000000000	0.000000000000	0.000000000000	0.000000000000	0.1692882024	0.1527052440
$n=2$	−0.057270822778	−0.057263842231	0.057270822778	0.057263842231	0.1413293727	0.1527052570
$n=3$	0.000000000000	0.000000000000	0.000000000000	0.000000000000	0.1413293726	0.1527052552
$n=4$	0.001534556347	0.001525765990	0.001384214180	0.001385279482	0.1413293726	0.1057740510
$n=5$	0.000000000000	0.000000000000	0.000000000000	0.000000000000	0.1413293723	0.1378764782
$n=6$	−0.000030362664	−0.000039532082	0.000036228677	0.000036352179	0.1322830080	0.1378764770
$n=7$	0.000000000000	0.000000000000	0.000000000000	0.000000000000	0.1322830080	0.1378764789
$n=8$	−0.000051072754	−0.000012860359	0.000000961465	0.000000958461	0.1322830080	0.1378764782

Table 7.9: Computing p_n, c_n coefficients, and eigenvalues of the light transport operator of the "Cubes0027" scene using both methods.

	c_n from p_n		p_n		Eigenvalues	
	Method 1	**Method 2**	**Method 1**	**Method 2**	**Method 1**	**Method 2**
$n=0$	1.000000000000	1.000000000000	1.000000000000	1.000000000000	0.2064156593	0.2138074460
$n=1$	0.000000000000	0.000000000000	0.000000000000	0.000000000000	0.2064156114	0.2138073540
$n=2$	−0.279244300237	−0.278933873304	0.279244300237	0.278933873304	0.2023577579	0.1937585918
$n=3$	0.007500427708	0.007547716663	0.003750213854	0.003773858332	0.2023577187	0.1937584785
$n=4$	0.224246713064	0.223781268110	0.001614237430	0.001605174820	0.1961984398	0.1989876302
$n=5$	−0.018504984867	−0.018687671759	0.000101647167	0.000098561111	0.1961984385	0.1989875820
$n=6$	−0.288053968272	−0.287195516755	0.000021370385	0.000021071462	0.1944873320	0.1966731877
$n=7$	0.046583497529	0.047025961034	0.000002807100	0.000002452263	0.1944873320	0.1966731665
$n=8$	0.498389928135	0.495545750119	0.000000281482	0.000000447366	0.1765352924	0.1839885444

Table 7.10: Computing p_n, c_n coefficients, and eigenvalues of the light transport operator of the "Cubes0343" scene using both methods.

	c_n from p_n		p_n		Eigenvalues	
	Method 1	**Method 2**	**Method 1**	**Method 2**	**Method 1**	**Method 2**
$n=0$	1.000000000000	1.000000000000	1.000000000000	1.000000000000	1.1000928389	1.1631356201
$n=1$	0.000000000000	0.000000000000	0.000000000000	0.000000000000	1.0756078855	1.1631355857
$n=2$	−5.210606089970	−5.210790443404	5.210606089970	5.210790443404	1.0545023393	1.1447176044
$n=3$	0.302767329542	0.302594756490	0.151383664771	0.151297378245	1.0545023393	1.1447175722
$n=4$	81.067671352881	81.073486929993	0.063929353603	0.063920700870	1.0351373416	1.0188967352
$n=5$	−15.535566211644	−15.526468744297	0.010018612496	0.010046246214	1.0351373416	1.0188967346
$n=6$	−2091.539631230130	−2091.770349300102	0.003203685718	0.003215610492	0.9023724343	1.0052980202
$n=7$	833.393159904196	832.895413020379	0.000890292848	0.000895596109	0.9023724240	1.0052980005
$n=8$	75143.352967791128	75154.797648109001	0.000290081501	0.000293446770	0.9102283847	0.0753217116

Table 7.11: Computing p_n, c_n coefficients, and eigenvalues of the light transport operator of the "TwoSquares-NoSymmetry" scene using both methods.

	c_n from p_n		p_n		Eigenvalues	
	Method 1	**Method 2**	**Method 1**	**Method 2**	**Method 1**	**Method 2**
$n=0$	1.000000000000	1.000000000000	1.000000000000	1.000000000000	0.1198089324	0.1127072930
$n=1$	0.000000000000	0.000000000000	0.000000000000	0.000000000000	0.1198089321	0.1127072926
$n=2$	−0.033270522719	−0.033271666352	0.033270522719	0.033271666352	0.1198089324	0.1127072965
$n=3$	0.000000000000	0.000000000000	0.000000000000	0.000000000000	0.1198089323	0.1127072925
$n=4$	0.000429321568	0.000430682084	0.000481910246	0.000481721544	0.0978065694	0.1051197717
$n=5$	0.000000000000	0.000000000000	0.000000000000	0.000000000000	0.0978065692	0.1051197713
$n=6$	−0.000002882675	−0.000011735242	0.000007445569	0.000007514570	0.0978065693	0.1053575006
$n=7$	0.000000000000	0.000000000000	0.000000000000	0.000000000000	0.0978065694	0.1053574955
$n=8$	0.000036624733	−0.000018131278	0.000000108105	0.000000120506	0.0931735515	0.1053574955

Table 7.12: Computing p_n, c_n coefficients, and eigenvalues of the light transport operator of the "CornellBox.ifs.scaled" scene using both methods.

	c_n from p_n		p_n		Eigenvalues	
	Method 1	**Method 2**	**Method 1**	**Method 2**	**Method 1**	**Method 2**
$n=0$	1.000000000000	1.000000000000	1.000000000000	1.000000000000	2.1713602868	2.1660484073
$n=1$	0.000000000000	0.000000000000	0.000000000000	0.000000000000	2.1684989334	2.1631873046
$n=2$	−20.582239380717	−20.481654891761	20.582239380717	20.416548917761	2.0847585063	2.0796579898
$n=3$	0.178850361615	0.177936933362	0.089425180807	0.088968466681	2.0847585459	2.0796580294
$n=4$	1268.830874347934	1256.457537196904	0.342476571244	0.339504019859	2.0821364073	2.0770359614
$n=5$	−36.589350752259	−36.226214537492	0.009252450356	0.009092253285	2.0821363673	2.0770359214
$n=6$	−130154.572883096058	−128255.009634325063	0.007707471209	0.007451536397	1.8081862240	1.8037606255
$n=7$	7847.178572910076	7731.666955233658	0.000803148431	0.000787517193	1.8081861617	1.8037605634
$n=8$	18661115.733795121312	18298792.824101842940	0.000319090656	0.000295352848	1.8062892201	1.8018632427

7.4 Discussion

Considering the fact that the light transport operator is acting on infinite-dimensional space, the generalization of the Fredholm determinant to the operators over infinite–dimensional spaces is required. On the other hand, the Fredholm determinant is generalized for the operators with specific properties, such as compactness, or even stronger assumptions on the operator, such as being Hilbert–Schmidt or trace class. Therefore, the background theory to ensure the convergence of the calculations to the real values is limited to these specific classes of operators. We know that the light transport operator is not even compact in general (Theorem 4.4.5). So, we restrict the discussion to a very specific scene, the scene without edges, since the operator is Hilbert–Schmidt in this situation (Theorem 4.4.4), and the regularized determinant is allowed to be used for the operator. Among our test scenes, the scenes in which the operator is Hilbert–Schmidt are the "TwoSquares" scene, the "Cubes0027" scene, and the "Cubes0343" scene. Although the theory ensures extracting eigenvalues from this determinant, only the estimated eigenvalues of the "TwoSquares" scene are close to the reference values. The failure of the method for the two other scenes may originate from calculating lots of visibility calculations between the cubes conjugated with the fact that computing the roots of a polynomial is highly sensitive to the polynomial coefficients [Act90].

Chapter 8

Eigenvalue extraction based on a polynomial approach using path tracing

Under the assumption of the compactness property, we know that the eigenvalues of the light transport operator are isolated, and have a decreasing order(tend to zero). On the other hand, the spectrum of the operator is contained in the unit disk. Benefiting from these properties, it is possible to consider a finite number of these eigenvalues and construct a polynomial of a finite degree and compute its roots by numerical algorithms.
We develop a method to approximate the eigenvalues of the light transport operator by computing the roots of the (estimated) characteristic polynomial, depending on the initial distribution. To compensate for this dependency, we form histograms of the roots and obtain better estimates of the eigenvalues.
In Section 8.1, we explain the theoretical background of the method. In Section 8.2, we perform the method on a matrix with the given properties such as, size, rank, explicit form of the matrix, and eigenvalues. In Section 8.3, we perform the method for light transport both in screen–space and object–space. However, since the method is originally designed based on histograms, we focused on the performance of the object–space run. Therefore, only a static scene is used for the former and different test scenes for the latter. We finish the chapter by a discussion on the designed method, Section 8.4.

8.1 Method's strategy

Suppose that T be the light transport operator when it is compact. Therefore, the eigenvalues of this operator are isolated. Let $\{\lambda_1, \cdots, \lambda_m\}$ be the set of first m dominant eigenvalues of T and $T_m : \oplus\Lambda_i \longrightarrow \oplus\Lambda_i$ indicates a finite rank approximation of the light transport operator of rank m. Let $p_m(x) =$

$\sum_{k=0}^{m} \alpha_k x^k$ denote the characteristic polynomial of T_m. From Cayley–Hamilton theorem [1], we have $p_m(T) = 0$.
Let $(\mathbf{p_i})_{1 \leq i \leq n}$ be a set of points sampled in the scene, and J be a matrix in which the columns of J are filled by the evaluations of $T^k L$ at these points, i.e. $t_{ik} := (T^{k+1}L)(\mathbf{p_i})$:

$$J = \begin{bmatrix} v_1 & t_{11} & \cdots & t_{1m} \\ \vdots & \vdots & & \vdots \\ v_n & t_{n1} & \cdots & t_{nm} \end{bmatrix}, \tag{8.1}$$

in which $v = (TL)(\mathbf{p_i})_{1 \leq i \leq n}$. Practically, it is possible to estimate $\alpha = (1, \alpha_1, \cdots, \alpha_m)^t$ [2] by solving the following system

$$\begin{bmatrix} v_1 & t_{11} & \cdots & t_{1m} \\ \vdots & \vdots & & \vdots \\ v_n & t_{n1} & \cdots & t_{nm} \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ \alpha_m \end{bmatrix} = 0 \tag{8.2}$$

, which the solution can be approximated by the following optimization problem

$$\alpha^* := \underset{\alpha}{\operatorname{argmin}} \|J\alpha\|_2.$$

The optimization problem above is equivalent to the following problem

$$\alpha^* := \underset{\alpha}{\operatorname{argmin}} \|J'\alpha - v\|_2 \tag{8.3}$$

, where J' contains the same columns as J, except the first one.
One may solve the previous problem by differentiating $\phi(\alpha) := \|J'\alpha - v\|^2$ i.e.

$$\begin{aligned} \phi(\alpha) &= (J'\alpha - v)^t (J'\alpha - v) = \alpha^t J'^t J'\alpha - 2\alpha^t J'^t v + \|v\|^2 \\ \frac{d\phi}{d\alpha} &= 2J'^t J'\alpha - 2J'^t v, \end{aligned}$$

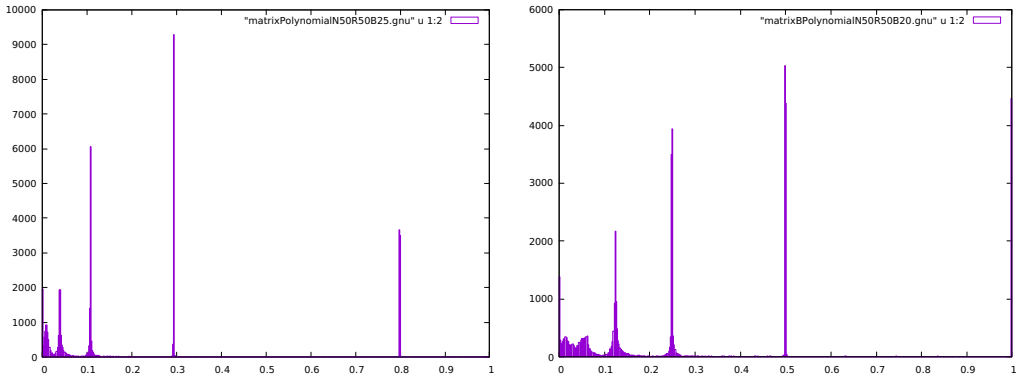
therefore $\alpha^* = (J'^t J')^{-1}(J'^t v)$ is an approximation of the coefficients of $p_m(x)$. So, depending on v, some of the eigenvalues of T_m can be obtained by extracting the roots of $p_m^*(x) = \sum_{k=0}^{k=m} \alpha_k^* x^k$.

8.2 Validation of the method on the matrix

To evaluate the method's performance for Lambertian scenes, we examine the method on two matrices A and B. We build these matrices such that their eigenvalues are real, have a magnitude less than 1, and follow a decreasing order.
Let A be a full–rank matrix of dimension 50 with the following eigenvalues

$$\lambda_i = 0.8e^{-i} \ , \ i = 0, 1, \cdots, m-1.$$

[1] This theorem states that evaluating the characteristic polynomial of a matrix in the matrix itself gives the zero matrix.
[2] t indicates the transpose.



(a) The histogram of the eigenvalues of a pre–constructed matrix with the reference values ($0.8e^{-i}$).

(b) The histogram of the eigenvalues of a pre–constructed matrix with the reference values ($\frac{1}{2^i}$).

Figure 8.1: Illustration of the method's performance on two pre–constructed matrices with decreasing eigenvalues.

It is clear that this sequence has the properties expressed earlier.
Extracting the roots of a polynomial is highly sensitive to its coefficients [Act90], therefore, we do not expect to approximate the eigenvalues by using only one individual v. But, one may repeat the process several times to plot a histogram of the computed roots. Therefore, the picks in the histogram are better approximations of the related eigenvalues.
We construct 10^4 samples of J', by computing $A^k v$ for $k = 0, \ldots, 25$, while randomly changing v. We denote these samples with $(J'_j)_{1 \leq j \leq M}$. For each J'_j, the α^* vector is computed then, the roots of polynomial $p_{25}(x)$ are calculated.

Remark 8.2.1 *There is a remarkable difference between this matrix simulation and light transport, rather than just the dimension. It concerns the representation of light transport. In our method, we do not use any matrix representation of the light transport operator to compute the columns of J'. Mathematically speaking, we keep using the operator (continuous) representation of light transport. This means that a path tracer computes each power of T for a given initial light distribution L using a Monte Carlo algorithm. In this way, we avoid discretizing scenes*[3]*. However, in this simulation, we produce $A^k v$ vectors by matrix–vector multiplication using the explicit form of the matrix directly.*

In Figure 8.1a, the histogram of the roots of $p_{25}(x)$ for matrix A is given. Reading eigenvalues with index $i \geq 5$ from the histogram in Figure 8.1a is not possible. Therefore, we look at the data from which the histogram is plotted. We observe that the estimation of eigenvalues becomes less accurate after $i = 5$, see Table 8.1. Note that the distances of consecutive eigenvalues of A become

[3]However, note that to provide the reference eigenvalues for our test scenes, we could not skip discretization.

Figure 8.2: The distance of consecutive eigenvalues of two pre–constructed matrices.

A	
d_1	0.5056
d_2	0.1860
d_3	0.0684
d_4	0.0251
d_5	0.0092
d_6	0.0034

(a) d_i values of matrix A.

B	
d_1	0.5
d_2	0.25
d_3	0.125
d_4	0.0625
d_5	0.03125
d_6	0.0156

(b) d_i values of matrix B.

smaller when index i grows. Moreover, the magnitude of each eigenvalue is also decreasing.
We also examine the method on another full–rank matrix, B. We set the eigenvalues of this matrix to be $(\frac{1}{2^i})$. We repeat the process on this matrix, where $m = 20$. However, the two peaks corresponding to λ_5 and λ_6 are less sharp in the histogram of Figure 8.1b, but the data of this experiment illustrates that the method still could catch these values, see Table 8.2.
Let $d_i = \lambda_i - \lambda_{i+1}$. In Figure 8.2, d_i values for matrices A and B are given. Looking at these values and the number of extracted eigenvalues by the method may imply the dependency of the process on the root's properties.

Table 8.1: Comparison of the estimated eigenvalues of matrix A from the histogram with the reference values.

A					
Estimated eigenvalue	0.8	0.294	0.108	0.038	0.012
Reference eigenvalue	0.8	0.294304	0.108268	0.0398297	0.0146525

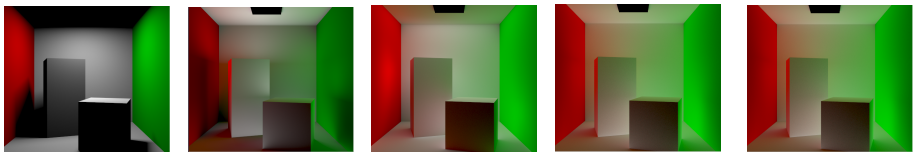
Table 8.2: Comparison of the estimated eigenvalues of matrix B from the histogram with the reference values.

B							
Estimated eigenvalue	0.998	0.498	0.248	0.124	0.06	0.03	0.014
Reference eigenvalue	1	0.5	0.25	0.125	0.0625	0.03125	0.015625

8.3 The method's performance on light transport

This method can be performed both on screen–space and object–space. For the former, a static light source and camera are used to produce a sequence of images using a path tracer. Each image of this sequence is computed by generating a large number of light paths, starting from the light source and making k number of intersections in the scene before reaching the camera. The disadvantage of using a static light source is that p_m will depend on the choice of L, so it will make the estimation of this polynomial biased by the choice of the initial distribution. To illustrate this fact, in Section 8.3.1, we perform the method on a single scene ("CornellBox" scene).
Considering the latter case, in the previous section, it was shown how the method estimates the suitable eigenvalues, the ones which are not too close, by constructing a histogram of the computed roots from the samples of J', $(J'_j)_{1\leq j\leq M}$, where M is a large positive value.
Implementing this method for the light transport operator in the object–space contains two important key parameters: the number of rows of J'_j, n, and M. Unlike the case of the matrix, in which the rank of the operator is finite, no information is available on the rank of the global light transport operator. On the other hand, applying the large values of n and M increases the cost. So, the question that arises here is What are the minimum possible values of these two parameters to achieve good approximations? Answering this question requires insight into the connection between the scene properties such as the material and the geometry, with the rank of the global light transport operator, moreover, the level of the importance between n and M in comparison to each other. Despite the interest in answering such a question, we do not discuss this issue here, and the question remains open.
In Section 8.3.2, the results of applying this method on the object–space is disposed. In this case, we do not compute the full images but only for a limited collection of points $(\mathbf{p_i})_{1\leq i\leq n}$. For this case, the point source, denoted by $\mathbf{x_s}$, is changed randomly to build new samples of J'_j, and the histogram of the roots of $p_m(x)$ is plotted for the test scenes.

8.3.1 Results with a single light source in image space



(a) Image TL (b) Image T^2L (c) Image T^3L (d) Image T^4L (e) Image T^5L

Figure 8.3: Images of T^kL of the "CornellBox" scene rendered by a path tracer.

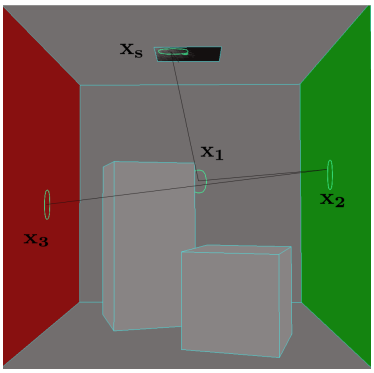


Figure 8.4: An example of a light path that contributes to the image of T^3L.

We experiment with the "CornellBox" scene in which the location of the light source is fixed in the ceiling of the scene, Figure 8.4. To render each image of T^kL, given k, lots of light paths of the form $\overline{x} = (\mathbf{x_s}, \mathbf{x_1}, \cdots, \mathbf{x_k}, \mathbf{x_e})$ contribute to the final image. To generate each path, first, $\mathbf{x_s}$ is sampled on the area light, then the path is grown by the Russian roulette technique. Adding a value to each corresponding pixel, is the task of projection step. That is, the reflected radiance from the last vertex of the *suitable* [4] path is multiplied by a factor, which we call it *sensor factor*. Therefore, this operation is linear, and feasible on the image–space.
The images of T^kL that fill matrix J' are given in Figure 8.3 [5]. The roots of the polynomial obtained from J' and the reference eigenvalues are given in the following table.

[4] By the word suitable we mean the path whose the last vertex of it is visible from camera position.

[5] It is worth mentioning that rendering full images even with a static light source using path tracing is very costly.

Table 8.3: Biased approximation of the eigenvalues of the "CornellBox" scene using the images of T^kL ($k = 1, \cdots, 5$).

Roots					
0.358585	0.224228	–	–	-0.185755	-0.24076
Reference Eigenvalues					
0.329183	0.265896	0.258180	0.229453	0.211690	-0.205733

8.3.2 Results with multiple light sources

For each row of the matrix J', one point, say $\mathbf{p_i}$, is sampled in the scene, then the powers of TL are computed at this point. Computing $(T^kL)(\mathbf{p_i})$ means to collect the energy, transported along the paths of length k, at this point. For a given k, since the last vertex of all paths must be $\mathbf{p_i}$, it is required to connect the point $\mathbf{x_k}$ to $\mathbf{p_i}$. In practice, it means to multiply the contribution of the subpath $(\mathbf{x_s}, \cdots, \mathbf{x_k})$ by the following factor

$$c_{\mathbf{x_k p_i}} = \rho(\mathbf{x_{k-1}} \to \mathbf{x_k} \to \mathbf{p_i}) G(\mathbf{x_k}, \mathbf{p_i}) V(\mathbf{x_k}, \mathbf{p_i}). \tag{8.4}$$

In our implementation, we grow the path applying Russian roulette, that is growing the path stops until it is absorbed with an absorption probability. Let this absorption probability be $1 - \rho(\mathbf{x})$. Therefore, light reflects with the probability which is exactly equal to $\rho(\mathbf{x})$.
Starting from $\mathbf{x_s}$, to trace $\mathbf{x_1}$, a direction is sampled according to the cosine–weighted BRDF distribution. For the next intersection points, at each current intersection point $\mathbf{x_t}$, we sample a direction according to the BRDF. Therefore, the contribution of the generated path with h number of bounces is

$$C(\mathbf{x_s}, \mathbf{x_1}, \cdots, \mathbf{x_h}) = \frac{\Pi_{t=1}^{h-1} \rho(\mathbf{x_{t-1}} \to \mathbf{x_t} \to \mathbf{x_{t+1}})}{\Pi_{t=1}^{h-1} \rho(\mathbf{x_{t-1}} \to \mathbf{x_t} \to \mathbf{x_{t+1}})}. \tag{8.5}$$

Since our method requires evaluating the contribution of the paths with k bounces in the scene, so the values, that are provided by equation (8.5), must be multiplied by the related factors. For each generated path by the Russian roulette technique, we construct the paths with different lengths 2 to k. Then, we evaluate the contribution of each path accordingly. Therefore, generating several light paths, and using the contribution measurements of all paths with different lengths, will lead to obtain J'_j matrix, needed to compute α^*.
So what remains is to compute the contribution of paths with different lengths correctly. Let $(\mathbf{x_s}, \cdots, \mathbf{x_h})$ be a path generated by the Russian roulette method. If the length of this path h is less than k, then it adds no energy to the estimate of $(T^kL)(\mathbf{p_i})$. But if $k \leq h$ then, the contribution of the path is non–zero. We separate the two cases $k < h$ and $k = h$ respectively:

- when $k < h$, means light is reflected at $\mathbf{x_k}$, but we need to cut the path at this point, and then connect it to $\mathbf{p_i}$, see Figure 8.5a. So

$$C(\mathbf{x_s}, \cdots, \mathbf{x_k}, \mathbf{p_i}) = \frac{c_{\mathbf{x_k p_i}}}{\rho(\mathbf{x_k})} \tag{8.6}$$

(a) Cutting the path at $\mathbf{x_k}$ and connecting it to $\mathbf{p_i}$.

(b) Connecting the point $\mathbf{x_k}$ to $\mathbf{p_i}$.

Figure 8.5: Building paths of length k from the generated paths by the Russian roulette technique.

- if $k = h$, means the path is absorbed at $\mathbf{x_k}$. It only requires to connect it to $\mathbf{p_i}$. Therefore,

$$C(\mathbf{x_s}, \cdots, \mathbf{x_k}, \mathbf{p_i}) = \frac{c_{\mathbf{x_k p_i}}}{1 - \rho(\mathbf{x_k})} \tag{8.7}$$

In equation (8.6), $C(\mathbf{x_s}, \cdots, \mathbf{x_k}, \mathbf{p_i})$ is divided by $\rho(\mathbf{x_k})$ to compensate for the reflection at $\mathbf{x_k}$, however, in equation (8.7), $C(\mathbf{x_s}, \cdots, \mathbf{x_k}, \mathbf{p_i})$ is divided by $1 - \rho(\mathbf{x_k})$ to compensate for the absorption.

The remainder of this section provides the pseudocode algorithm of the method. The method is implemented in the framework of a function, see Algorithm 2. There is a key function, we called it "pointSourcePathTrace", which computes the rows of the matrix J' in each iteration. The pseudocode of this function is also committed in Algorithm 1 separately. There are other functions applied in these two functions, such as "growPath", "computeAlphaValues" and "zroots". We briefly explain these functions.

- "growPath($\mathbf{x_s}$)". This function generates a path, starting from a given surface point $\mathbf{x_s}$, by the Russian roulette technique.
- "computeAlphaValues(J')". This function solves equation $(J'^t J')\alpha = J'^t v$ for α using conjugate gradient method.
- "zroots(α^*)". This function computes the roots of a polynomial of a finite degree with Laguerre's method [6].

In Table 8.4, the value of M, n, and N used for each scene, and the correspondent histogram are addressed.

[6] Laguerre's method is a fairly simple algorithm to compute the roots of a polynomial. Using this algorithm, it is almost guaranteed to converge to some root of the polynomial [Act90].

Algorithm 1 pointSourcePathTrace function

$\overline{x} \leftarrow \text{growPath}(\mathbf{x_s})$
$cumulRef \leftarrow 1$
for each vertex $\mathbf{x_t} \in \{\mathbf{x_1}, \cdots, \mathbf{x_h}\}$ **do**
  if $V(\mathbf{x_t}, \mathbf{p_i}) \neq 0$ **then**
    compute $\rho(\mathbf{x_t})$
    $W \leftarrow \mathbf{p_i} - \mathbf{x_t}$
    localize W at $\mathbf{x_t}$
    localize $-W$ at $\mathbf{p_i}$
    compute $\rho(\mathbf{x_{t-1}} \rightarrow \mathbf{x_t} \rightarrow \mathbf{p_i})$
    compute $\rho(\mathbf{x_t} \rightarrow \mathbf{p_i} \rightarrow -)$
    $J'_{i(t+1)} \leftarrow cumulRef$
    if $\|cumulRef\| \neq 0$ **then**
      $J'_{i(t+1)} \leftarrow \frac{J'_{i(t+1)}}{\|cumulRef\|}$
    end if
    if $k = h$ **then**
      $J'_{i(t+1)} \leftarrow \frac{J'_{i(t+1)}}{1-\rho(\mathbf{x_t})}$
    else
      $J'_{i(t+1)} \leftarrow \frac{J'_{i(t+1)}}{\rho(\mathbf{x_t})}$
    end if
    $J'_{i(t+1)} \leftarrow \frac{J'_{i(t+1)}}{2}$
    $J'_{i(t+1)} \leftarrow J'_{i(t+1)} * \rho(\mathbf{x_{t-1}} \rightarrow \mathbf{x_t} \rightarrow \mathbf{p_i}) * G(\mathbf{x_t}, \mathbf{p_i})$
  end if
  $cumulRef \leftarrow cumulRef * \rho(\mathbf{x_t} \rightarrow \mathbf{p_i} \rightarrow -)$
end for

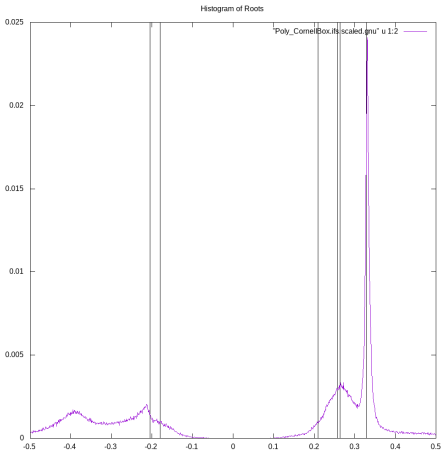
Algorithm 2 Polynomial method with path tracing

for each $\mathbf{x_s}$ **do**
  generate random measurement points $(\mathbf{p_i})_{1\leq i\leq n}$
  generate random directions for the measurement points
  $J' \leftarrow$ zero matrix
  for $g = 1, \cdots, N$ **do**
    $\overline{x} \leftarrow$ growPath($\mathbf{x_s}$)

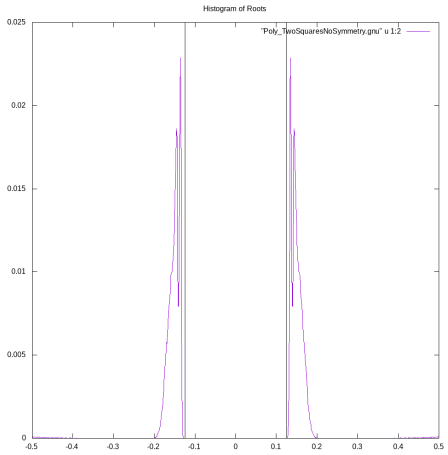
    for each $\mathbf{p_i}$ **do**
      sample a direction for $\mathbf{p_i}$
      $J'_{i\cdot} \leftarrow$ pointSourcePathTrace ($\mathbf{x_s}$, $\mathbf{p_i}$)
      add $J'_{i\cdot}$ to the i–th row of J'

      for $k = 2, \cdots, m$ **do**
        $J'_{ik} \leftarrow J'_{ik} * \pi$
      end for
    end for
  end for
  $J' \leftarrow J' * \frac{1}{N}$
  $\alpha^* \leftarrow$ ComputeAlphaValues(J')
  $\{r_1, \cdots, r_m\} \leftarrow$zroots($\alpha^*$)
  add the real roots to the histogram
end for

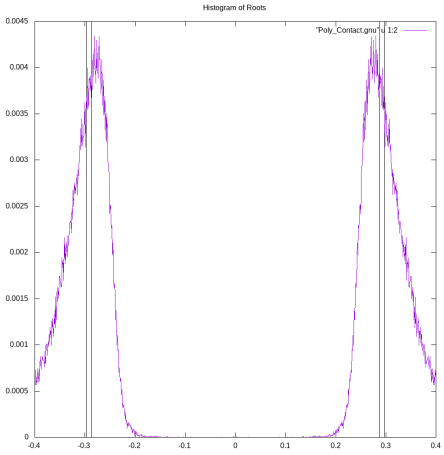
Figure 8.6: Performing the method on test scenes.



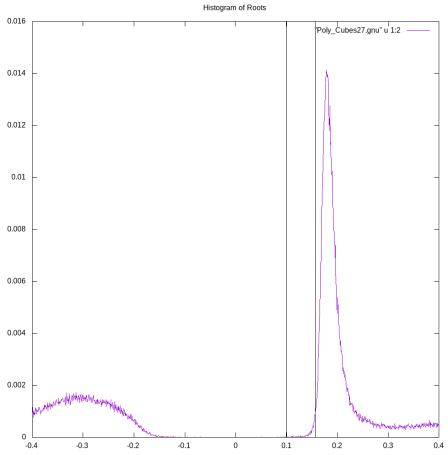
(a) "CornellBox.ifs.scaled" Scene



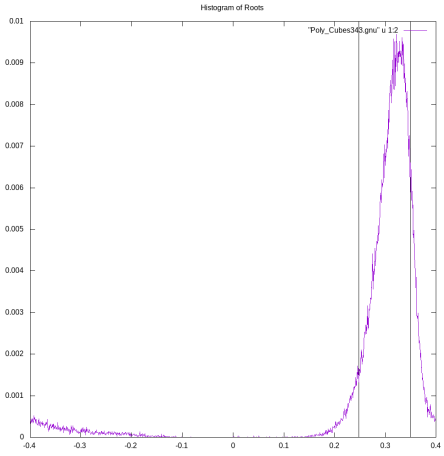
(b) "TwoSquaresNoSymmetry" Scene



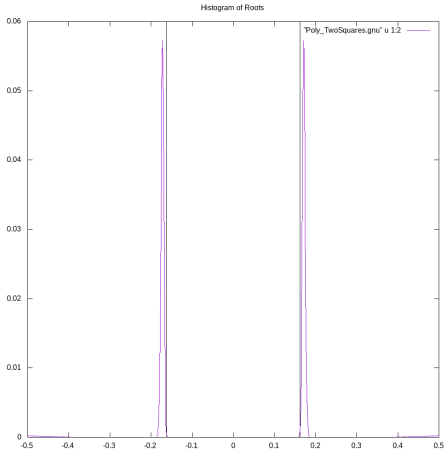
(c) "Contact" Scene



(d) "Cubes027" Scene



(e) "Cubes0343" Scene



(f) "TwoSquares" Scene

Table 8.4: The results of the method in object–space.

Results on Different Scenes			
Scene	"CornellBox.ifs.scaled"	"TwoSquares"	"TwoSquaresNoSymmetry"
M	10^5	10^5	10^5
n	10^3	200	10^3
N	10^5	10^5	10^5
Figure	8.6a	8.6f	8.6b
Results on Different Scenes			
Scene	"Contact"	"Cubes0027"	"Cubes0343"
M	10^5	10^5	10^5
n	10^3	500	10^3
N	10^5	10^5	10^5
Figure	8.6c	8.6d	8.6e

8.4 Discussion

There are several methods to compute the roots of polynomials, such as Newton's method, Laguerre's method, Lehmer method, Root-squaring methods and many others.
Regardless of such a simple task, there is no general method to extract all roots of a polynomial. This fact is because of the dependency of this procedure on many factors, especially root properties, like being real or complex, isolated or not, the algebraic multiplicity, distance of the roots with respect to each other, etc. Therefore, choosing an approach depends on the primary information (in the case of existence) on such factors and the purpose of the experiment. Computing the roots in our method is no exception among others. These factors influence the results produced by this method. The other disadvantage of extracting roots by machines is that, a very small change in the coefficients of a polynomial may change the expected roots notably. Wilkinson et al. [Wil59] showed that making a little change on a coefficient of a polynomial can even lead to get false complex roots in the place of reference real ones.
It is worth noting that, the problems do not only limit to the change of coefficients, but also on the roots itself. That is, most of methods do not obtain the roots at once, some roots are computed earlier. For example, if the polynomial contains complex roots, then the real roots are computed first. In such methods, if one root is not calculated accurately enough, then it may also interrupt extracting the rest of roots to search for. Wilkinson also revealed that the failure originates from the regular distance of the roots with respect to each other.
The worst-case scenario is when no primary information is available, which is generally the case. In this situation, it is required to have a system equipped with the decision making ability, to choose the method depending on the problem. Designing such a system is far more complicated than the original issue, which is computing the roots.

Considering the light transport operator, we know that the eigenvalues are real in Lambertian scenes (See Theorem 4.A.3). However, there is no information available on the multiplicity of the eigenvalue. Such information is precious, since extracting multiple roots cause troubles for numerical algorithms, such as Newton's method [7]. For a special geometry of the scene, in which the edges are removed, the light transport operator is compact. Therefore, the non–zero eigenvalues are isolated with finite algebraic multiplicity, but no detection of the multiplicity. Among our test scenes, the light transport acting on "TwoSquares-NoSymmetry", "TwoSquares", "Cubes0027" and "Cubes0343" is compact. In our implementation, the task of extracting roots is done by Laguerr's algorithm. This algorithm is able to obtain real and complex roots from complex coefficients. The advantage of this algorithm is that the method always converges to some root of the polynomial independent of the initial choice. This algorithm is also able to extract the roots with non–trivial multiplicity. However, this method is developed under some assumptions namely, the first root be isolated and the other roots to have *some* distance with the first root. The isolation assumption of all roots is not included in these assumptions, but very close space of roots increase the error of estimating roots using this algorithm. This originates from the stop condition of the iteration loop. Let $q(x)$ be a polynomial of degree m. Then this algorithm estimates a root, we denote it by $\hat{r}$, by choosing trials $\hat{r}_{k+1} = \hat{r}_k - a$. Variable a is defined as the following fraction

$$a = \frac{m}{D \pm \sqrt{(m-1).(mH - D^2)}},$$

where $D = \frac{q'_n}{q_n}$ and $H = [\frac{q'_n}{q_n}]^2 - \frac{q''_n}{q_n}$. The iteration loop stops when a is small. Therefore, even considering Lambertian scenes, when the operator is not compact, root properties stay covert, so using Laguerr's method becomes a blind action.
Considering the lack of the primary information on the eigenvalues of light transport, the limited theoretical background behind Laguerr's algorithm, and the dependency of our method on different factors, especially the position of all eigenvalues, it is not straightforward to conclude that which parameters exactly cause the behavior of histograms. Nevertheless, we finish the chapter by addressing a few observations from the histograms.

- Similar to the matrix case, discussed in Section 8.2, Laguerr's method is not accurate when the roots of the polynomial are closely spaced. This factor is also one of the items causes failure of detecting such eigenvalues of the "CornellBox.ifs.scaled" scene and the "Contact" scene.

- The accuracy of the method does not depend only on the magnitude of the first value. For example, the first eigenvalue of the "CornellBox.ifs.scaled" scene and the "Cubes0343" scene are similar however, the peaks of the

[7] Newton's method is an appropriate method to extract real isolated roots. However, this method is very likely to miss multiple roots.

"CornelBox.ifs.scaled" scene's histogram coincide the reference ones much better than the ones of the "Cubes0343" scene.

Chapter 9

Conclusion

In the simulation of light transport, it is essential to reduce the computation cost, which requires information on the dimension of the global light transport operator. The dimensionality might be defined as the dominant eigenvalues of the light transport operator. Even though local light transport has already been studied, the dimensionality of global light transport has not been studied extensively, up to our knowledge. In this project, we studied the light transport operator as an operator acting on an infinite-dimensional space, and proposed numerical methods, using path tracing algorithms, to search for eigenvalues of the operator in Lambertian scenes.

Theory. To design such algorithms, spectral information of the light transport operator is required. This necessity originates from the fact that the set of points in which operator $T - z$ does not have an inverse is not necessarily limited to the set of eigenvalues.
The literature on the spectral study of bounded operators offers the compactness property to simplify this issue. This property ensures that the operator behaves similarly to a matrix, and the spectrum of the operator becomes limited to a decreasing set of eigenvalues with an only accumulation point at 0.
In formulating the theories developed in this manuscript, this property showed up either as a necessary or a sufficient condition. Therefore, following Arvo's work [Arv95a], while studying the functional properties of this operator (Chapter 4), we studied the compactness property. We proved that for diffuse surfaces, abutting edges of the geometry prevent the operator to be compact.

Methods. Although lack of this property made the task of finding eigenvalues more challenging, we implemented our methods based on the continuous representation of the operator.

Using a Monte Carlo formulation of the power iteration method, in Chapter 6, we estimated the dominant eigenvalue using images of the powers of the operator.

In Chapter 5, we discussed the resolvent of the light transport operator by passing the theory from finite-dimensional vector spaces to infinite-dimensional Hilbert spaces. We showed how the resolvent theory over infinite–dimensional

spaces connects the integral of the resolvent to the eigenspaces of corresponding isolated eigenvalues. If the set of eigenfunctions of the light transport operator is complete, we can project an arbitrary distribution into the eigenspaces of the isolated eigenvalue by a line integral of the resolvent (at least in theory). Therefore, proving this property (completeness of the set of eigenfunctions) is beneficial. We also described the resolvent of light transport in terms of the power series of this operator. Speaking of power series, to extend the representation of the resolvent to wider regions of the complex plane, we also discussed the analytic continuation of the resolvent. However, performing analytic continuation on the resolvent of light transport demands spectral information from the operator. Namely, to be able to apply the powers of the operator, it is necessary to know the location of the largest spectrum element in magnitude, belonging to the unit disk.
In Lambertian scenes, when the operator is compact, searching for this element is reduced to the set of eigenvalues on the real line (since eigenvalues are real in this case). It could be possible to use estimates of the dominant eigenvalues, obtained in Chapter 6. But calculating coefficients that represent the resolvent in terms of power series suffer from numerical instability, due to sequential calculations of the coefficients from one region (circle) to another in the complex plane.

The generalization of the Fredholm determinant to a compact operator can be applied to estimate the eigenvalues of the operator by computing the poles of this determinant. However, the extension of this theory has developed for specific classes of compact operators, namely, Hilbert–Schmidt and trace class. We know that the light transport operator is not compact in general. However, in Lambertian scenes, it becomes Hilbert–Schmidt by removing the edges of the geometry. Therefore, the Fredholm theory provides a formula to compute the coefficients participating in the Fredholm determinant using kernel evaluations at pairs of points. Among our test scenes, there are three in which the operator is compact ("TwoSquares", "Cubes027", and "Cubes0343"), but the method only works for the "TwoSquares" scene. This failure may originate from intrinsic properties of the two other scenes which require multiple visibility calculations while computing kernel evaluations.
Among the operators related to the rendering equation, $K_{\mathbf{x}}$ is compact, therefore it is possible to solve the Fredholm equation of this integral to obtain its eigenvalues. On the other hand, the eigenpair of this operator is related to singular value decomposition (SVD) of the light transport operator [1]. Therefore, extracting eigenvalues of the local reflectance operator is helpful to represent the SVD expansion of T.

In chapter 8, we estimated the characteristic polynomial of a finite rank operator (which is an approximation of the light transport operator) using a matrix filled by the estimates of powers of the operator at measurement points. The background theory of this method is developed under the assumption of the

[1] Since T is not compact, the existence of the SVD for light transport is not guaranteed. But, we proved, under a limited set of configurations, that this expansion exists.

compactness property, which is a sufficient condition to have isolated eigenvalues with a decreasing order. However, the method still works for the "CornellBox" scene, in which the operator is not compact. Our intuition is that even though we do not know about the rank of global light transport and the operator is not compact, but this operator has a countable set of isolated and dominant eigenvalues allowing computing the eigenvalues of this scene.
The performance of this method might be improved using the Metropolis algorithm. However, in this way, one needs to scale each column with a factor. So, it is required to compute the factor for each consecutive column of the matrix.

In developing the supportive theory of the methods experimented on in our test scenes, the compactness property helped us to simplify the problems. However, without this sufficient condition, some of our methods still work. This observation suggests the existence of special spectral properties which are always fulfilled with the compactness property. These properties might be the countability and the isolation of eigenvalues of the light transport operator.

Bibliography

[Act90] Forman S Acton. *Numerical methods that work*. Maa, 1990.

[Ahl79] Lars Ahlfors. Complex analysis mcgraw-hill. *Inc., New York*, 1979.

[Ama84] John Amanatides. Ray tracing with cones. *ACM SIGGRAPH Computer Graphics*, 18(3):129–135, 1984.

[Arv95a] James Arvo. The role of functional analysis in global illumination. In *Eurographics Workshop on Rendering Techniques*, pages 115–126. Springer, 1995.

[Arv95b] James Richard Arvo. *Analytic methods for simulated light transport*. PhD thesis, Yale University, 1995.

[Ash01] Ian Ashdown. *Eigenvector radiosity*. PhD thesis, University of British Columbia, 2001.

[BBR97] Gladimir V. G. Baranoski, Randall Bramley, and Jon G. Rokne. *Eigen-Analysis for Radiosity Systems*. 1997.

[BC09] James Ward Brown and Ruel V Churchill. *Complex variables and applications*. McGraw-Hill,, 2009.

[BK70] J Bouknight and K Kelley. An algorithm for producing half-tone computer graphics presentations with shadows and movable light sources. In *Proceedings of the May 5-7, 1970, spring joint computer conference*, pages 1–10, 1970.

[Bor10] Folkmar Bornemann. On the numerical evaluation of fredholm determinants. *Mathematics of Computation*, 79(270):871–915, 2010.

[CG85] Michael F Cohen and Donald P Greenberg. The hemi-cube: A radiosity solution for complex environments. *ACM Siggraph Computer Graphics*, 19(3):31–40, 1985.

[Cha11] Françoise Chatelin. *Spectral approximation of linear operators*. SIAM, 2011.

[CR19] Christophe Cheverry and Nicolas Raymond. Handbook of spectral theory. *Lecture, September*, 2019.

[CT82] Robert L Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)*, 1(1):7–24, 1982.

[CWH93] Michael F Cohen, John R Wallace, and Pat Hanrahan. *Radiosity and realistic image synthesis*. Morgan Kaufmann, 1993.

[DAK01] I Dimov, V Alexandrov, and A Karaivanova. Parallel resolvent monte carlo algorithms for linear algebra problems. *Mathematics and Computers in Simulation*, 55(1-3):25–35, 2001.

[DBB18] Philip Dutre, Kavita Bala, and Philippe Bekaert. *Advanced global illumination*. AK Peters/CRC Press, 2018.

[FL50] George E Forsythe and Richard A Leibler. Matrix inversion by a monte carlo method. *Mathematics of Computation*, 4(31):127–129, 1950.

[Gau61] Michel Gaudin. Sur la loi limite de l'espacement des valeurs propres d'une matrice aléatoire. *Nuclear Physics*, 25:447–458, 1961.

[GGK96] Israel Gohberg, Seymour Goldberg, and Naum Krupnik. Traces and determinants of linear operators. *Integral Equations and Operator Theory*, 26(2):136–187, 1996.

[GGK13] Israel Gohberg, Seymour Goldberg, and Marius A Kaashoek. *Classes of linear operators*, volume 63. Birkhäuser, 2013.

[GK78] Israel Gohberg and Mark Grigor'evich Kreĭn. *Introduction to the theory of linear nonselfadjoint operators*, volume 18. American Mathematical Soc., 1978.

[Gou71] Henri Gouraud. *Computer display of curved surfaces*. The University of Utah, 1971.

[GTGB84] 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, 1984.

[GTZ97] Sergei A Goreinov, Eugene E Tyrtyshnikov, and Nickolai L Zamarashkin. A theory of pseudoskeleton approximations. *Linear algebra and its applications*, 261(1-3):1–21, 1997.

[HH64] JM Hammersley and DC Handscomb. Monte carlo methods, methuen & co. *Ltd., London*, 40, 1964.

[Hil12] David Hilbert. *Basic principles of a general theory of linear integral equations*. Number 3. BG Teubner, 1912.

[HPB06] Miloš Hašan, Fabio Pellacini, and Kavita Bala. Direct-to-indirect transfer for cinematic relighting. *ACM transactions on graphics (TOG)*, 25(3):1089–1097, 2006.

[HPB07] Miloš Hašan, Fabio Pellacini, and Kavita Bala. Matrix row-column sampling for the many-light problem. In *ACM SIGGRAPH 2007 papers*, pages 26–es. 2007.

[HR10] Fu-Chung Huang and Ravi Ramamoorthi. Sparsely precomputing the light transport matrix for real-time rendering. In *Computer Graphics Forum*, volume 29, pages 1335–1345. Wiley Online Library, 2010.

[HTSG91] Xiao D He, Kenneth E Torrance, Francois X Sillion, and Donald P Greenberg. A comprehensive physical model for light reflection. *ACM SIGGRAPH computer graphics*, 25(4):175–186, 1991.

[Kaj86] James T Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986.

[KAMJ05] Anders Wang Kristensen, Tomas Akenine-Möller, and Henrik Wann Jensen. Precomputed local radiance transfer for real-time lighting design. In *ACM SIGGRAPH 2005 Papers*, pages 1208–1215. 2005.

[Kat13] Tosio Kato. *Perturbation theory for linear operators*, volume 132. Springer Science & Business Media, 2013.

[KW09] Malvin H Kalos and Paula A Whitlock. *Monte carlo methods*. John Wiley & Sons, 2009.

[KZ91] AS Kalitvin and PP Zabrejko. On the theory of partial integral operators. *The Journal of Integral Equations and Applications*, 3(3):351–382, 1991.

[LAM+11] Bradford J Loos, Lakulish Antani, Kenny Mitchell, Derek Nowrouzezahrai, Wojciech Jarosz, and Peter-Pike Sloan. Modular radiance transfer. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–10, 2011.

[LSSS04] Xinguo Liu, Peter-Pike J Sloan, Heung-Yeung Shum, and John Snyder. All-frequency precomputed radiance transfer for glossy objects. *Rendering Techniques*, 2004, 2004.

[Meh04] Madan Lal Mehta. *Random matrices*. Elsevier, 2004.

[Mid19] William Edgar Knowles Middleton. *Pierre Bouguer's Optical Treatise on The Gradation of Light*. University of Toronto Press, 2019.

[MSRB07] Dhruv Mahajan, Ira Kemelmacher Shlizerman, Ravi Ramamoorthi, and Peter Belhumeur. A theory of locally low dimensional light transport. In *ACM SIGGRAPH 2007 papers*, pages 62–es. 2007.

[Nar88] MHN Naraghi. Radiation view factors from differential plane sources to disks-a general formulation. *Journal of thermophysics and heat transfer*, 2(3):271–274, 1988.

[Nev93] Olavi Nevanlinna. Spectrum, resolvent and power boundedness. In *Convergence of Iterations for Linear Equations*, pages 13–45. Springer, 1993.

[Nie92] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*. SIAM, 1992.

[NN85] Tomoyuki Nishita and Eihachiro Nakamae. Continuous tone representation of three-dimensional objects taking account of shadows and interreflection. *ACM SIGGRAPH Computer Graphics*, 19(3):23–30, 1985.

[NRH+77] FE Nicodemus, JC Richmond, JJ Hsia, IW Ginsberg, and T Limperis. Geometric considerations and nomenclaturefor reflectance. *US Dept. of Commerce, Washington, DC, NBS Monograph*, 160, 1977.

[NRH03] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. In *ACM SIGGRAPH 2003 Papers*, pages 376–381. 2003.

[NS82] Arch W Naylor and George R Sell. *Linear operator theory in engineering and science*. Springer Science & Business Media, 1982.

[NSKF07] Derek Nowrouzezahrai, Patricio Simari, Evangelos Kalogerakis, and Eugene Fiume. Eigentransport for efficient and accurate all-frequency relighting. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, pages 163–169, 2007.

[OP11] Jiawei Ou and Fabio Pellacini. Lightslice: matrix slice sampling for the many-lights problem. *ACM Trans. Graph.*, 30(6):179, 2011.

[Pes73] Peter H Peskun. Optimum monte-carlo sampling using markov chains. *Biometrika*, 60(3):607–612, 1973.

[Pho75] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[RH01] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500, 2001.

[Rud73] W Rudin. Functional analysis . macgraw-hill book company. *International Edition*, 1973.

[Rud91] Walter Rudin. Functional analysis, mcgrawhill. *Inc, New York*, 45:46, 1991.

[SG08] Jerome Spanier and Ely M Gelbard. *Monte Carlo principles and neutron transport problems*. Courier Corporation, 2008.

[SHHS03] Peter-Pike Sloan, Jesse Hall, John Hart, and John Snyder. Clustered principal components for precomputed radiance transfer. *ACM Transactions on Graphics (TOG)*, 22(3):382–391, 2003.

[SKS02] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 527–536, 2002.

[Tre16] François Treves. *Topological Vector Spaces, Distributions and Kernels: Pure and Applied Mathematics, Vol. 25*, volume 25. Elsevier, 2016.

[Vea98] Eric Veach. *Robust Monte Carlo methods for light transport simulation*. Stanford University, 1998.

[VLG96] Charles F Van Loan and G Golub. Matrix computations (johns hopkins studies in mathematical sciences). *Matrix Computations*, 1996.

[WDT+09] Jiaping Wang, Yue Dong, Xin Tong, Zhouchen Lin, and Baining Guo. Kernel nyström method for light transport. In *ACM SIGGRAPH 2009 papers*, pages 1–10. 2009.

[Wil59] James H Wilkinson. The evaluation of the zeros of ill-conditioned polynomials. part i. *Numerische Mathematik*, 1(1):150–166, 1959.

[WS00] Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. *Advances in neural information processing systems*, 13, 2000.

[WZH07] Rui Wang, Jiajun Zhu, and Greg Humphreys. Precomputed radiance transfer for real-time indirect lighting using a spectral mesh basis. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 13–21, 2007.

[Zem12] Stephen M Zemyan. *The classical theory of integral equations: a concise treatment*. Springer Science & Business Media, 2012.

[ZTNZ77] Olgierd Cecil Zienkiewicz, Robert Leroy Taylor, Perumal Nithiarasu, and JZ Zhu. *The finite element method*, volume 3. McGraw-hill London, 1977.