PHOTON SURFACES: RENDERING VOLUME ROBUSTLY WITH UNBIASED DENSITY ESTIMATORS

A Thesis
Submitted to the Faculty
in partial fulfillment of the requirements for the
degree of
Master of Science
in
Computer Science
by
Xi Deng
DARTMOUTH COLLEGE
Hanover, New Hampshire
2019-03-07

Examining Committee:

Wojciech K. Jarosz (Chair)

Peter Mann Winkler

Lorie Loeb

F. Jon Kull, Ph.D.
Dean of the Guarini School of Graduate and Advanced Studies

Abstract

Participating media, such as foggy atmosphere, fire, stained window glasses and smoke, are widely used in modern films, games and related fields for realistic synthesizing. Satisfying the demand of rendering participating media accurately and efficiently remains challenging today. Previous work either solved the problem by using unbiased estimators with low convergence rates, or by rendering efficiently at the expense of adding bias. Our work focuses on deriving robust photon density estimators for volume rendering that are both accurate and efficient.

We construct a new theory framework for the derivation of unbiased photon density estimators in participating media rendering. In this framework, we combine analytic integration of parts of the path integral in extended path space with Monte-Carlo estimators to approximate intensity of pixels. Through specific choices of analytic integration variables, we then derive a group of unbiased density estimators, named "photon surfaces." We concentrate mainly on the distance sampling domain ("generalized photon planes") and the point sampling domain on an area light source ("single scattering photon surfaces"), which makes it possible to render a light-to-medium path with unbiased density estimators.

The second contribution of our work relates to variance reduction. Due to the geometric nature of different photon surfaces, each estimator introduces singularities in some specific area. These singularities produce low frequency noise and slow down the convergence rate. To address this, we further develop several robust density estimators by applying multiple importance sampling (MIS) among our photon surfaces estimators to combine their advantages. As a result, MIS's estimators dramatically reduce the singularities and improve the convergence rate. Moreover, we also show how we combine a continuum of estimators with MIS, which helps to improve the robustness of single scattering photon surfaces estimators.

Purple mists rise from the Incense Burner Peak in the sun; The waterfall seems to hang above the stream, seen from far away. Straight down three thousand feet the white spraying torrent does run. Descending from Ninth Heaven, could this be the Milky Way?

— Bai Li (701-762), translated by Frank C Yu

Preface

I would like to thank all the people that helped me in either studying or daily life during my time at Dartmouth College. First of all, I would like to express my sincere gratitude to my thesis advisor Dr. Wojciech Jarosz, who inspired my interest in graphics research, for his continued support in my study during the past two years and his constructive suggestions on this work; and the director of Digital arts program at Dartmouth College Prof. Lorie Loeb for providing me the opportunity to presume my Master's degree in digital arts program, also for her warm encouragements throughout hard times. I would love to thank Dr. Peter Winkler for giving detailed guidance on thesis writing.

Next I want to express my genuine thankfulness to my colleagues in Dartmouth Visual Computing Lab, Benedikt Bitterli, who provided precise answer to my endless questions with patience and great professionality; Shaojie Jiao, who helped me in getting through many detail implementation problems; Dario Seyb, Kate Salesin and Zack Misso who aided me in revising my Ph.D. applications letters; My best friend Annie Dai who supports me with her free driving service regardless of time and my location. I would like to thank my friend Beibei Xu and Ouxiang Zhou for reading this thesis and giving detailed comments on writing.

Finally, I would love to thank my parents, Wenyi Tan and Kui Deng for their love and unconditionally backing through years; my grandparents, Wenxia Zhang and Jicai Deng for supports and understanding and my brother Bohai Hou for his encouragement at key times.

Contents

List of Figures

Chapter 1

Introduction

It can be observed from the visual effects of recent films and games that rendering techniques have experienced rapid growth during the past couple of decades. However, that growth of technique never satisfied the great aspirations of films and games makers; the demand in rendering realistic images pushes researchers to dive deeper into physically-based simulation. Among those realistic visual effects, rendering participating media is always a big topic, because we are actually living in participating media. Much appearance modeling, including skin, fabric, snow, clouds, fire, water, fog and smoke, is closely related to light transport within participating media.

Rendering participating media accurately and efficiently remains a challenging problem today due to the complexity of describing a volume and the limitations of current algorithms. The earliest model of it is given by Kajiya (1986a), which formulates the problem as solving a radioactive transfer equation. A popular Monte-Carlo algorithm named "path tracing" was introduced at that time and has been used to solve rendering problems for years because of its robustness, being an unbiased and solid theoretical foundation. A statistical Monte-Carlo rendering algorithm is unbiased as it is guaranteed to converge to exact answer and its only error is the variance. Nevertheless, path tracing can not simulate some complex light paths, while suffering from a low convergence rate. Some later unbiased rendering techniques, including next event estimation and bidirectional path tracing (Lafortune and Willems 1993; Veach and Guibas 1994; Lafortune and Willems 1996), were proposed to compensate this problem. These approaches listed above are widely used, even in the rendering of participating media, because of their generality and the ease of supporting both surfaces and media. However, those approaches suffer from efficiency problems as they usually rely on a huge amount of samples to converge.

Algorithms like photon mapping use density estimating methods along with Monte-Carlo methods and improve the convergence rate performance. Similar to bidirectional path tracing, those density estimators trace a subpath from camera and a subpath from light source. Different from bidirectional path tracing, photon mapping does not directly connect two subpaths by shadow rays as BPT does; it uses a blurring kernel to couple the subpaths. This allows the photon estimators to store photon samples first and reuse them later in render passes, but introduces additional errors by blurring along some dimensions. Researchers later discovered that the limitation on point sample is unnecessary, (Jarosz et al. 2008b) developed 1D camera "query beams" to replace query points. Then (Jarosz et al. 2011b) generalized this segment sample to photon subpath, and both on camera and light subpath. The 1D segment sample requires less blurring dimensions and therefore reduces related error at a certain extent. Although the photon beam is still a biased method, it provides insights for further research on developing unbiased density estimators by taking higher dimension samples. (Bitterli and Jarosz 2017) first introduced the unbiased density estimators "0D Photon Planes" as well as the "photon volume" which combines two or three sampling dimension and stores the photon sample as planes and volumes. Those photon planes and volumes do not ask for blurring along any dimension, and thus are unbiased with sample noise being the only error.

1.1 Motivation

Though the previous photon planes method provides unbiased estimators for simulating light transport in participating medium, it is not practical since photon planes suffer from singularities when viewed from glancing angles. It usually takes large amount of samples to smooth out those singularities. Moreover, a photon plane takes two propagation distance-sampling dimensions, so it can be used for the path that has at least two propagation distance in medium. However, this can not support the first bounce off surface and light source (single scattering event). Unfortunately, the single scattering event usually has significant impact on rendering images, thus it matters to extend the theory to support single scattering events.

Our initial goal is to generalize the photon plane theory to construct high dimensional photon samples so that we can render the single scattering event with unbiased density estimators. Then, we want to mitigate the singularities of those high dimensional photon samples by utilizing multiple importance sampling to combine their advantages.

1.2 Contribution

Solving the problems mentioned in section 1.1, There are two main contribution in this thesis.

We build up our theory based on photon mapping and the path integral framework in extended path space. Instead of directly applying Monte-Carlo methods to estimate the path integral, we separate the integral into numeric part and analytic part. The analytic part always takes three sampling dimensions and will be replaced by an analytically-preintegrated value. The remaining numeric part of the path integral is then estimated by Monte-Carlo methods. We re-derived the original photon plane estimator in our framework and generalized it to a boarder family of unbiased density estimators named "photon surfaces" through choosing different combination of analytic integral dimensions. By utilizing two dimensions of point sampling on light source as the analytic integration domain, we procure single scattering photon surfaces including ut-/vt-/uv-planes for planar light and uv-surfaces for area light, which solved single scattering problems for area light sources. It also provides insights for us to construct unbiased single scattering photon surfaces for points light by employing the directional sampling variables as analytic integration domains.

Furthermore, we interpreted our unbiased density estimators as path sampling strategies so that we can apply multiple importance sampling among them. By doing this, we can also combine our photon surfaces with previous path sampling strategies to create new, robust density estimators that significantly reduce noise.

1.3 Structure Overview

Among the seven chapters of this thesis, Chapter 2 briefly goes through related research of the problem and the solution we used in this paper.

Chapter 3 introduces the basic concepts of light transport simulation, reviews the rendering equations in both vacuum and participating medium, and presents the standard path integral that we aim at solving.

Chapter 4 reviews the Monte-Carlo algorithms that are widely used in solving the path integral, describes each algorithms from the point of interpreting them as path sampling strategies, introduces the path integral in extended path space, and builds our theory framework.

Chapter 5 shows how we re-derive the photon planes estimator in our framework, goes through the process of deriving photon surfaces in a general form and gives the derivations of each photon surfaces estimators including generalized photon planes ($t_i t_j$-planes), uv-surface, uv-planes, ut-planes, vt-planes and randomly-oriented ut-/vt-planes.

Chapter 6 lists different ways of combining the photon surface estimators yielded in Chapter 5 by multiple importance sampling. In this chapter, we describe how we combine discrete strategies as well as give an idea of combining a continuum of strategies using multiple importance sampling.

We illustrate in Chapter 7 how we implement and validate our theory in both real-time visualization and a physically-based renderer, and then present the rendering results.

Finally, in Chapter 8, we discuss further on our algorithms and the potential problems and solutions. We also look forward to some future research.

Chapter 2

Related Works

In this chapter, we review some prior works on density estimators, refer to several cases in graphics that have been solved by utilizing partial analytic integration and describe broadly on recent techniques that related to our theory. We will also discuss in this chapter how our work relates to these works.

2.1 Density Estimators & High Dimensional Samples

Density estimators have been used in rendering for years because of its ability to render tricky light paths like caustics. However, because of the additional blurring process, there exists a sacrifice of accuracy when utilizing density estimators with Monte-Carlo methods to render volume. To keep the advantage of rendering hard light paths as well as reducing unwanted bias, high dimensional samples are gradually used to develop robust volume density estimators. Based on photon mapping, (Havran et al. 2005) introduced ray map which stores a whole photon path as a sample instead of photon points for global illuminations. (Jarosz et al. 2008a) presented camera query beam, which is a line sample on camera subpath, to replace multiple point-queries with beam query. Then (Jarosz et al. 2011a) generalized the theory to compact line samples on photon subpaths("photon beams"). They later presented nine density estimators deduced from different combination of "beams" and "points" on camera and photon subpaths to further reduce the blurring bias. At the same time, to take the benefits of the convergence guarantees and memory efficiency which provided by progressive photon mapping from (Hachisuka et al. 2008), the "photon beams" was extended by (Jarosz et al. 2011c) as progressive photon beams. More recently, (Bitterli and Jarosz 2017) successfully made the density estimator unbiased through developing higher dimensional photon samples named “photon planes” and “photon volumes”.

Previous work usually interprets these multi-dimensional photon samples by marching and taking limits, while this paper reformulates it from the view of integration. Our work provides a framework that allows us to derive a variety of high dimensional photon samples for density estimators. We pre-integrate part of the path integral over some integration domains, which we refer as “analytic variables”, before Monte-Carlo process. This pre-integrated part sweeps out high dimensional photon samples as well as reduces the blurring dimensions. Prior density estimators can be also explained in this framework, where “photon beams” are expressed as pre-integrating one distance sampling domain on photon subpath while “camera beams” corresponds to one-dimensional integration on camera subpath. Likewise, the “photon plane” can be interpreted as pre-integrating over two conjunction distances on photon subpath and the last distance on sensor subpath. Thinking in our theory framework, the “photon planes” from (Bitterli and Jarosz 2017) is actually a special case of a family of “photon plane” estimators which the pre-integrated distance sampling domain could be any two segments on the photon subpath instead of only using the last two segments. We also developed single scattering photon surfaces for area light source by employing point sampling domains on area light. The variety of choices for pre-integration domains produce a group of unbiased density estimator and make it possible for us to further construct robust density estimator by multiple importance sampling among them.

2.2 Analytic Integration

As mentioned in previous section, one key of our framework is the analytic integration of parts of the path integral. Indeed, analytic integration of part of the integral has been applied in solving a wide range of problems in graphics. (Gribel et al. 2010, 2011; Barringer et al. 2012; Nowrouzezahrai et al. 2014; Billen and Dutré 2016) applied analytic integration in solving visibility problems. (Jones and Perry 2000) used the analytic nature of line sample to approximate antialiasing integral, while (Tzeng et al. 2012) utilized the line samples to compute analytic coverage in depth of field. There are also applications of analytic integration in single scattering (Sun et al. 2005; Pegoraro and Parker 2009), and area lighting (Arvo 1995a,b; Chen and Arvo 2000, 2001; Belcour et al. 2018).

Further more, our work uses the analytical integration together with the Monte-Carlo estimator to approximate the path integral. From this point of view, there are some previous work of combining analytic integration with Monte-Carlo through ratio (Heitz et al. 2018) or using control variates Belcour et al. (2018). According to recent analyses in (Sun et al. 2013; Singh et al. 2017; Singh and Jarosz 2017), these techniques smoothing out the integral by pre-integrating analytically and therefore help to improve the rate of convergence as well as variance reduction. Different from previous work, as we start the derivation with photon mapping which has an extra point when comparing to standard path space, we construct the path integral in the extended path space (Hachisuka et al. 2012, 2017). In this way, the integrand are ensured to be singular and all integrals are able to be computed in closed form.

2.3 Other Techniques

Specially, by analytically integrating parts of the extended path integral, we can also formulate unbiased path sampling strategies like BPT, VPLs and next-event estimation in our framework. More generally, we can consider our method as a way to reparameterize paths with new parameters set, thus the corresponding weight of our estimators are the geometry factor of parametrizations. As a reference to compare similar method, previous work from (Jakob 2013) introduces a reparametrization of specular chains with a corresponding generalized geometry factor.

Chapter 3

Light Transport Basics

Rendering generates realistic images from the description of geometries and illumination conditions of a scene. In general, a renderer achieves this through simulating light transport, such as reflection and refraction, and developing the picture from virtual film with the captured light information. In this chapter, we briefly review the key problem of interest that solved by Rendering and introduce the basic concepts of light transport simulation.

3.1 The fundamentals of light transport

In physically-based rendering, the light transport simulation is strictly conducted under physic laws, like optics.

Light generally refers to electromagnetic radiosity. In rendering, we mainly study a subset spectrum of light that directly causes visual sensation. This spectrum of light is usually referred to as visible light. Also, the light is considered as consisting of particles that carry energy called photons. This wave-like and particle-like property is the wave-particle duality of light.

Classic optics has two branches based on different aspects of light properties. One is the physics (wave) optics which consider lights as electromagnetic radiosities. In physics optics, the light propagates in the way the wave travels. Another branch is geometry (ray) optics in which light is considered to travel in straight lines. Rendering in modern graphics is built on top of the geometry optics. In geometry optics, a ray of light travels as a straight line in media until it hits any surface. It then gets reflected, refracted or absorbed

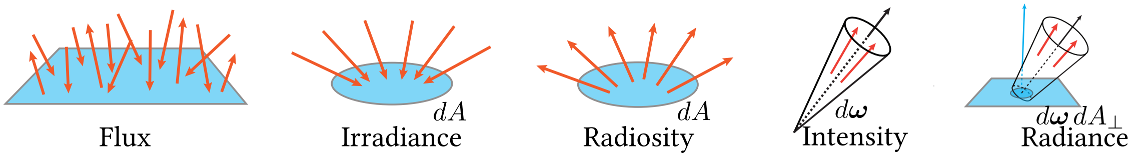

Figure 3.1: Radiometry measurements:

by the surface. This directional-change is referred as a scattering event or a bounce in the scene. Under the geometry optics assumption, phenomenons resulting from wave properties of light such as diffraction, polarization, and interference are not supported in modern graphics.

In this section, we'll introduce basic concepts of light transport in rendering, including the quantities of light, the bidirectional scattering distribution functions (BSDFs), and the phase functions.

3.1.1 Radiometry quantities

In rendering, we compute the amount of energy arrived at each pixel on the image plane. After the light's whole bouncing tour around the scene, we want to be able to quantitatively measure the energy it carries at the moment of hitting the sensor plane. Radiometry quantities measure electromagnetic radiation, such as visible light, which can describe the energy carried by photons. In this subsection, we introduce some basic radiometry quantities that are used in light transport simulation.

Photon power A flow of light consists of photons, and each photon has its own properties such as its currently position $\mathbf{x}$, traveling direction $\boldsymbol{\omega}$, and the amount of energy it packs. Suppose the wavelength of a light is λ, then each photon in this light carries an energy of $\frac{hc}{\lambda}$, where h is the Planck's constant[1] and c is the speed of light in vacuum[2].

Flux The flux represents the power (energy) emitted, received, transmitted, or reflected per unit time, such as the total energy of photons passing through the surface of a light bulb each second. Flux is usually expressed in terms of Φ, and the unit of flux is Joule per second $[\frac{J}{S}]$, also known as Watt [W].

[1] $h \approx 6.63 \times 10^{-34}\ \mathrm{m}^2 \times \mathrm{kg/s}$

[2] $c = 2.99792458 \times 10^8\ \mathrm{m/s}$

Irradiance & Radiosity The irradiance E is the flux per unit area arriving at a surface while the radiosity B is the flux per unit area leaving a surface. Their units are Watt per square meter. Generally, the irradiance and radiosity at a point $\mathbf{x}$ on surface A is written as

$$E(\mathbf{x}) = \frac{d\Phi_i(A)}{dA(\mathbf{x})} \qquad\qquad B(\mathbf{x}) = \frac{d\Phi_o(A)}{dA(\mathbf{x})}$$

Intensity The intensity, noted as I, is the directional density of power per unit solid angle:

$$I = \frac{\mathrm{d}\Phi}{\mathrm{d}\boldsymbol{\omega}}$$

Radiance Radiance L is flux emitted, received, transmitted, or reflected per unit solid angle per unit perpendicular area. It is written as:

$$L = \frac{\mathrm{d}^2\Phi}{\mathrm{d}\boldsymbol{\omega}\,\mathrm{d}A\cos\theta}$$

where θ is the angle between the area plane and the projected plane, and $\mathrm{d}A\cos\theta$ is on projected surface. During light transport simulation, we mainly use radiance term as the energy measurement so that we have the flexibility in both direction and surface area.

3.1.2 The BSDFs and phase functions

The light travels in straight lines before they get scattered, and scatterings happen when the light hits surface or particles in medium. When being scattered, radiance deflected to different directions. How the radiance distribute into these directions is determined by the material the light is interacting with. In graphics, to describe this material related property of surface elements and mediums, we use BSDFs and phase functions respectively.

BSDFs A surface is a boundary that separates different materials. In order to modeling the appearance of surfaces under illumination, we should know how light reflect/refract after hitting the surfaces. Nicodemus (Nicodemus 1965) provided the bidirectional reflectance distribution function (BRDF) to present the directional variation of light reflecting on an opaque surface element. At each bounce point, the BRDF gives how much the light reflected from one direction to another. Later in 1980s, the concept was expanded to

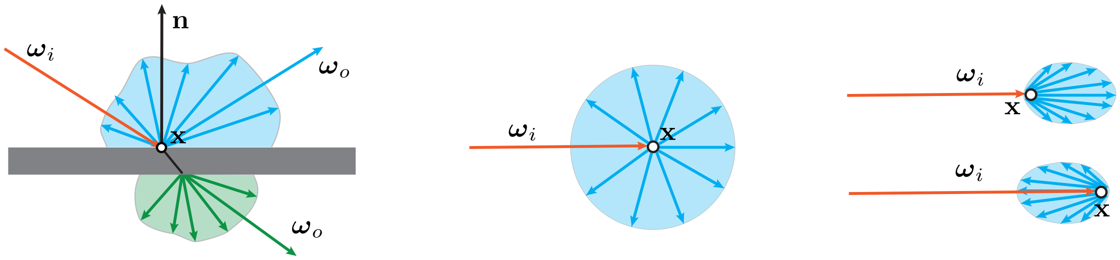

Figure 3.2: BSDFs and Phase Function: The first image on the left is an illustration of bidirectional scattering distribution function where the blue arrows are reluctance directions and the green arrows are transmission directions. Rest images on the right are phase function in the volume: the one oh the left relates to isotropic medium and the two images on the right are anisotropic medium (the top one is forward scattering while the bottom one is backward scattering).

a generalized mathematical function named the bidirectional scattering distribution function (BSDF) (F O Bartell 1981) which tells how light scatter after hit a surface. The BSDFs includes both BRDF and BTDF (bidirectional transmittance distribution function). At a scattering point $\mathbf{x}$, the BSDF of the surface takes the incident direction $\boldsymbol{\omega}_i$, the emitting direction $\boldsymbol{\omega}_o$ and the surface normal $\mathbf{n}$ as input and give us a ratio as output. This ratio represents the portion of the radiance from $\boldsymbol{\omega}_i$ that scatters into $\boldsymbol{\omega}_o$. To be precise, let $L(\mathbf{x},\boldsymbol{\omega}_i)$ and $L(\mathbf{x},\boldsymbol{\omega}_o)$ be incident radiance and outgoing radiance respectively, the BSDF at point $\mathbf{x}$ is

$$\rho_s(\mathbf{x},\boldsymbol{\omega}_i,\boldsymbol{\omega}_o) = \frac{\mathrm{d}L(\mathbf{x},\boldsymbol{\omega}_o)}{L(\mathbf{x},\boldsymbol{\omega}_i)\,|\mathbf{n}(\mathbf{x})\cdot\boldsymbol{\omega}_i|\mathrm{d}\boldsymbol{\omega}}$$

where $\mathbf{n}(\mathbf{x})$ is the surface normal at point $\mathbf{x}$.

Here we list some simple examples of BSDFs: Lambertian BRDF corresponds to uniformly distributed reflecting directions. Specular BSDF relates to surface like mirror and glasses where the reflecting direction is fixed given incident direction. Microfacet model relates to more complex surface, which may be anisotropic when reflecting, like surface of the mac book.

As we mainly interest in light transport in participating media, we won't discuss surface appearance modeling further and will just use ready-made BSDFs for surface rendering.

Phase functions When traveling in participating medium, light will hit particles and scatter into other directions (we will discuss this in detail later in **??**). Analog to BSDFs, phase function is used in describing the property of participating. It stands for the likelihood of light scatter from one direction into another in medium. A simple and straight example of phase function is the isotropic medium where the scattering direction is uniformly random distributed over the sphere (analog to lambertian BRDFs, middle column in ??). We use $\rho_p(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$ to denote the phase function, and for the isotropic medium $\rho_p(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{1}{4\pi}$. Also, the medium can be anisotropic: scattering direction may tend to form obtuse angle between incident direction (backward scattering) or acute angle (forward scattering). There are several models of anisotropic phase function such as Henyey-Greenstein phase function (Henyey and Greenstein 1941), Lorenz-Mie models and Rayleigh scattering model. Since our derivation of volume density estimator mainly concern about the propagation distance and point sample on light/camera, we won't discuss this direction-related property in detail.

Reversion-reciprocity principle Reversion-reciprocity principle or Helmholtz reciprocity says the light ray matches its inverse ray in optic activities like reflecting and refracting. As is shown in Figure 3.3, the randiance scattered from the light ray in direction $\boldsymbol{\omega}_i$ to $\boldsymbol{\omega}_o$ matches the randiance scattered from $\boldsymbol{\omega}_o$ to $\boldsymbol{\omega}_i$ if we switch the incident direction of the light to $\boldsymbol{\omega}_o$. This could also be expressed as:

$$\rho_s(\mathbf{x}_2, \boldsymbol{\omega}_{2i}, \boldsymbol{\omega}_{2o}) = \rho_s(\mathbf{x}_2, \boldsymbol{\omega}_{2o}, \boldsymbol{\omega}_{2i}) \text{ or } \rho_s(\mathbf{x}_2, \boldsymbol{\omega}_{2i} \leftrightarrow \boldsymbol{\omega}_{2o})$$

A simpler way to understand this is the statement "If I can see you, then you can see me." We rely on this principle in rendering so that we can swap camera and light source, or trace light path from camera as well as from light source.

3.2 Light transport in vacuum

Our illustration of light transport start with this simpler assumption where there is no medium in between surfaces, and lights travels in vacuum between scattering events. With this vacuum assumption, in Figure 3.3 the incident radiance $L(\mathbf{x}_2, \boldsymbol{\omega}_{2i})$ at vertex $\mathbf{x}_2$ equals to the outgoing radiance $L(\mathbf{x}_1, \boldsymbol{\omega}_{1o})$ at vertex $\mathbf{x}_1$.

One key problem in light transport simulating is to evaluate the radiance coming along the $\boldsymbol{\omega}_o$ from some point $\mathbf{x}$, which we note as $L(\mathbf{x}, \boldsymbol{\omega}_o)$ and the source of radiance on this ray is emission at point $\mathbf{x}$ and the radiance scattered into direction $\boldsymbol{\omega}_o$ at point $\mathbf{x}$.

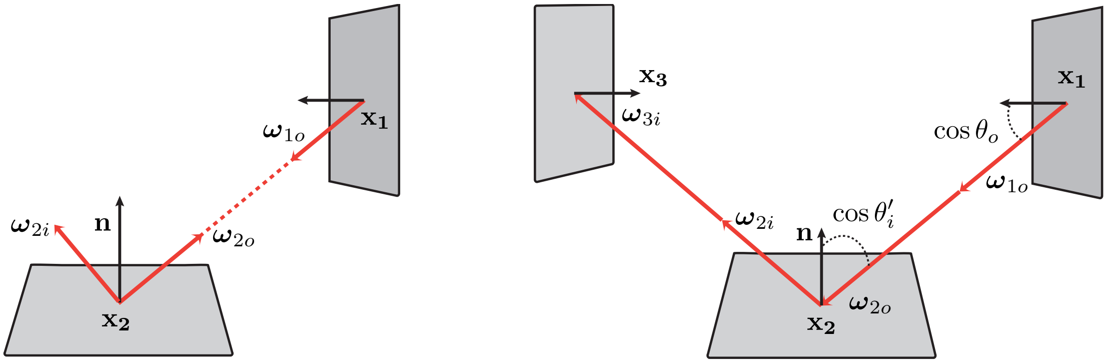
$\boldsymbol{\omega}_{1o}$
$\mathbf{x_1}$
$\boldsymbol{\omega}_{2i}$
$\mathbf{n}$
$\boldsymbol{\omega}_{2o}$
$\mathbf{x_2}$
$\mathbf{x_3}$
$\boldsymbol{\omega}_{3i}$
$\mathbf{x_1}$
$\cos\theta_o$
$\boldsymbol{\omega}_{1o}$
$\boldsymbol{\omega}_{2i}$
$\mathbf{n}$
$\cos\theta_i'$
$\boldsymbol{\omega}_{2o}$
$\mathbf{x_2}$

Figure 3.3: Two point model (on the left) and the three point model (on the right)

3.2.1 The scattering equation

Given the BSDFs (left most image in Figure 3.2) and the Helmholtz reciprocity principle, the scattering radiance from point $\mathbf{x}_2$ along direction $\boldsymbol{\omega}_{2o}$ noted as $L_s(\mathbf{x},\boldsymbol{\omega}_{2o})$ is the sum of radiance get scattered into $\boldsymbol{\omega}_{2o}$ from all the direction over sphere S, moreover,

$$L_s(\mathbf{x}_2,\boldsymbol{\omega}_{2o}) = \int_{S^2} \rho_s(\mathbf{x}_2,\boldsymbol{\omega}_{2i},\boldsymbol{\omega}_{2o}) L(\mathbf{x}_2,\boldsymbol{\omega}_{2i}) |\mathbf{n}(\mathbf{x}_2)\cdot\boldsymbol{\omega}_{2i}| \mathrm{d}\boldsymbol{\omega}_{2i} \tag{3.1}$$

This equation is the scattering equation.

3.2.2 The rendering equation

The outgoing radiance from point $\mathbf{x}$ on a surface along direction $\boldsymbol{\omega}_o$ is identical to the sum of corresponding emission radiance and scattered radiance. Knowing the scattered radiance from Equation 3.1, we add the emittance at point $\mathbf{x}$ and yield the radiance along direction $\boldsymbol{\omega}_o$:

$$\begin{aligned} L(\mathbf{x}_2,\boldsymbol{\omega}_{2o}) &= L_e(\mathbf{x}_2,\boldsymbol{\omega}_o) + L_s(\mathbf{x}_2,\boldsymbol{\omega}_{2o}) && (3.2)\\ &= L_e(\mathbf{x}_2,\boldsymbol{\omega}_{2o}) + \int_{S^2} \rho_s(\mathbf{x},\boldsymbol{\omega}_{2i},\boldsymbol{\omega}_{2o}) L(\mathbf{x},\boldsymbol{\omega}_{2i}) |\mathbf{n}(\mathbf{x}_2)\cdot\boldsymbol{\omega}_{2i}| \mathrm{d}\boldsymbol{\omega}_{2i} && (3.3) \end{aligned}$$

Equation 3.3 is the spherical form of rendering equation, and tackling most light transport simulating problems in graphics is directly related to solving this rendering equation. The rendering equation origins from Kajiya (1986b). It was presented in surface form and discribed in a three-point model (image on the right in Figure 3.3).

3.2.3 Jacobian

Sometimes, it's more convenient to use surface integration form of rendering equation:

$$L(\mathbf{x}_1,\mathbf{x}_2) = G(\mathbf{x}_1,\mathbf{x}_2)[L_e(\mathbf{x}_1,\mathbf{x}_2) + \int_A \rho_s(\mathbf{x}_1\mathbf{x}_2, \mathbf{x}_2\mathbf{x}_3) L_i(\mathbf{x}_2,\mathbf{x}_3) G(\mathbf{x}_2,\mathbf{x}_3) \mathrm{d}\mathbf{x}_3] \qquad (3.4)$$

In the equation above, $G(\mathbf{x}_1, \mathbf{x}_2)$ is the geometry term between two input points. It includes the visibility between points $\mathbf{x}_1$ and $\mathbf{x}_2$ and the Jacobian term that transfer the rendering equation from directional integration to surface integration:

$$G(\mathbf{x}_1,\mathbf{x}_2) = \begin{cases} 0, & \text{if } \mathbf{x}_1, \mathbf{x}_2 \text{ are not mutually visible,} \\ \dfrac{\cos\theta\cos\theta'}{r^2} & \text{otherwise,} \end{cases} \qquad (3.5)$$

where $r = |\mathbf{x}_1 - \mathbf{x}_2|$, $\cos\theta = \dfrac{\mathbf{x}_1 - \mathbf{x}_2}{r} \cdot \mathbf{n}(\mathbf{x}_2)$ and $\cos\theta' = \dfrac{\mathbf{x}_1 - \mathbf{x}_2}{r} \cdot \mathbf{n}(\mathbf{x}_2)$
Instead of parameterizing the radiance by the directions, the surface integration form only parameterize the radiance by the position of points on the surfaces, these two parameterization stand for different integration space, then there should be a Jacobian that map the integration in surface vertex form to directions form. The jacobian equals to the determinatant of the transformation matrix between these two type of parameterization.

3.3 Light transport in participating media

In last section, the radiometry transfers between surfaces and light are considered, however, a big assumption of the surface model is that there is no participating media between the surface and light travels in vacuum between scattering events. Therefore, realistic image with participating media ranging from the medium as sparse as morning fogs to the medium as dense as marbles are not supported.

Participating medium could be thought of as a large number of extremely small particles fill in a volume. For example, the morning fog consists of water droplets or ice crystals. When a ray of light travels pass a medium, a portion of photons in the light hit some particles in the medium being reflected away (out scattering) or a partial of their energy been absorbed during collision (absorption). At the same time, photons from other light may be scattered into this light (in-scattering). Also, the light ray collects radiance emitted by luminous medium into its traveling direction along the way (Emission). Although,

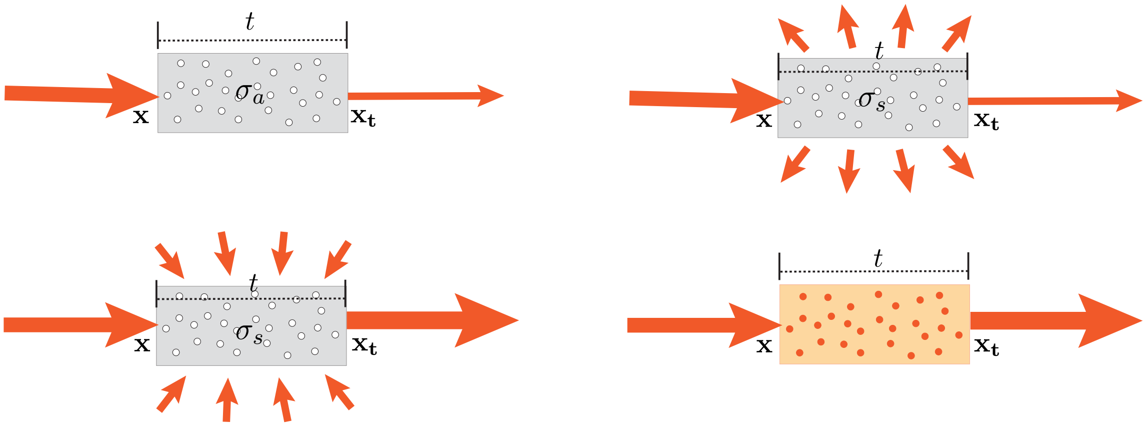

Figure 3.4: Light transport events in participating media: Absorption (top left), out-scattering (top right), in-scattering (bottom left) and emission (bottom right).

interacting with participating medium is, in fact, interacting with particles, in graphics, we don't actually model the particles explicitly. Instead, the participating media are described by statistical properties including density, absorption and scattering coefficients and phase functions.

To analyze the radiance change when a light ray goes through participating medium, in this subsection, we will review the derivation of radiative transfer equation by looking at a differential beam $L(\mathbf{x}, \boldsymbol{\omega})$ going from $\mathbf{x}$ along direction $\boldsymbol{\omega}$ by a distance t to $\mathbf{x}_t$ in medium, like Figure 3.4.

Absorption As is mentioned previously, when going through a volume, the radiance reduces due to absorption. This lose of radiance is measured by the absorption coefficient σ_a. Suppose the medium is purely absorption medium which usually looks transparent and dim, like sunglasses. The radiance at $\mathbf{x}_t$ is noted as $L(\mathbf{x}_t, \boldsymbol{\omega})$ or $L(t)$, then

$$dL(t) = -\sigma_a(t)L(t)\mathrm{d}t. \tag{3.6}$$

sovling this equation gives as the expression of $L(t)$

$$L(t) = C e^{-\int_0^t \sigma_a(t)\mathrm{d}t} \tag{3.7}$$

where by setting $t = 0$ we know $C = L(0) = L(\mathbf{x}_0, \boldsymbol{\omega})$. It can be also observed from Equation 3.7 the radiance decreases by a exponentially factor of absorption coefficient,

Out-scattering Except for the absorption, the radiance also decreases because of the photons' scattering into other directions, this portion of energy lose is measured by the scattering coefficient. Suppose the medium is not absorbing nor luminous, the radiance lose by out-scattering is

$$dL(t) = -\sigma_s(t)L(t)\mathrm{d}t. \tag{3.8}$$

similar as the absorption, solving this equation gives as a exponential decrease factor $e^{\int_0^t \sigma_s(t)\mathrm{d}t}$.

In-scattering As the radiance getting scattered away, when a light passes the volume, there are also photons scattered in from other directions. This increment radiance at point $\mathbf{x}_t$, denoted as $L_s(\mathbf{x}_t, \boldsymbol{\omega})$ is identical to integration of in-scattering radiance from all the directions over the sphere. Also, this increment in radiance is also described in terms of the scattering coefficient

$$\mathrm{d}L(t) = \sigma_s(t)L_s(t)\mathrm{d}t \tag{3.9}$$

where

$$L_s(t) = \int_{S^2} \rho_p(\boldsymbol{\omega}, \boldsymbol{\omega}_i)L(\mathbf{x}_t, \boldsymbol{\omega}_i)\mathrm{d}\boldsymbol{\omega}_i$$

The weight $\rho_s(\boldsymbol{\omega}, \boldsymbol{\omega}_i)$ is the phase function, which tells the likelyhood of a light traveled in direction $\boldsymbol{\omega}_i$ scattered into direction $\boldsymbol{\omega}$. This phase function is an analog of the BSDF term in surface scattering model.

Emission In addition to in scattering radiance, the emitted radiance of the volume itself $L_e(\mathbf{x}, \boldsymbol{\omega})$ also contributes to the change of radiance when light passes,for example liquid in a glow stick. This part of radiance change is expressed by

$$\mathrm{d}L(t) = \sigma_a(t)L_e(\mathbf{x}_t, \boldsymbol{\omega})\mathrm{d}t \tag{3.10}$$

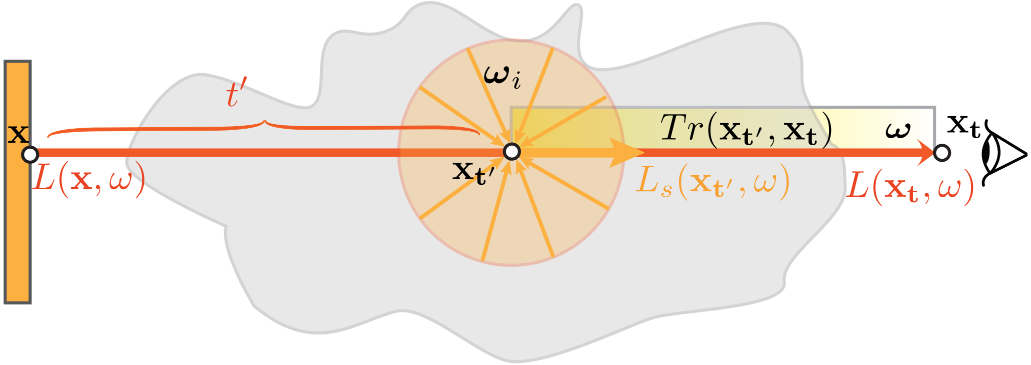

Figure 3.5: Radiative transfer

3.3.1 The radiative transfer equation

Summing up those factors of radiance lose and gain results in a differential form of radiative transfer equation:

$$\mathrm{d}L(t) = \sigma_a(t)L_e(t)\mathrm{d}t + \sigma_s(t)L_s(t)\mathrm{d}t - \sigma_t(t)L(t)\mathrm{d}t \tag{3.11}$$

where σ_t is the extinction coefficient that equals to the sum of scattering coefficient and absorption coefficient. The Equation 3.11 usually write in the form of radiance derivative with respect to distance traveled in the medium, which is referred as radiative transfer equation:

$$L'(t) + \sigma_t(t)L(t) = \sigma_a(t)L_e(t) + \sigma_s(t)\int_{S^2} \rho_p(\boldsymbol{\omega}, \boldsymbol{\omega}_i)L(\mathbf{x}_t, \boldsymbol{\omega}_i)\mathrm{d}\boldsymbol{\omega}_i \tag{3.12}$$

Solving the radiative transfer equation by pattern matching the ODE function $y' + qy = p$ gives us the full volume rendering equation in participating medium:

$$\begin{aligned} L(\mathbf{x}_t, \boldsymbol{\omega}) = L(t) = & \int_0^t Tr(t', t)L_e(t')dt' + Tr(0, t)L(0) \\ & + \int_0^t Tr(t', t)\sigma_s(t')(\int_{S^2} \rho_p(\mathbf{x}_{t'}, \boldsymbol{\omega}, \boldsymbol{\omega}_i)L(\mathbf{x}_{t'}, \boldsymbol{\omega}_i)\mathrm{d}\boldsymbol{\omega}_i)\mathrm{d}t' \end{aligned} \tag{3.13}$$

where the $Tr(t', t)$ is the transmittance term between $\mathbf{x}_{t'}$ and $\mathbf{x}_t$:

$$Tr(t', t) = e^{-\int_{t'}^{t} \sigma_t dt'}, \tag{3.14}$$

and it expresses the combined energy lose result from the absorption and out-scattering. Moreover, we can rewrite this Equation 3.15 in the same form of Equation 3.4 to keep the

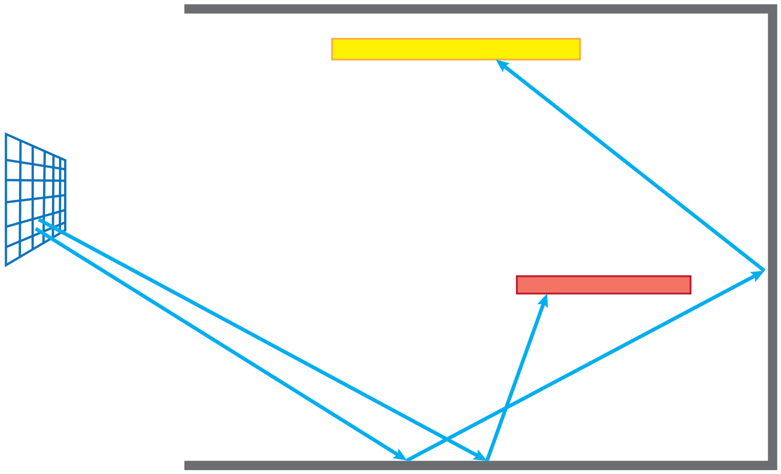

Figure 3.6: Camera plane and light paths

consistency of notations, we have:

$$\begin{aligned} L(\mathbf{x}_t, \boldsymbol{\omega}) = &\int_{\mathbf{x}}^{\mathbf{x}_t} Tr(\mathbf{x}_{t'}, \mathbf{x}_t) L_e(\mathbf{x}_{t'}, \boldsymbol{\omega}) d\mathbf{x}_{t'} + Tr(\mathbf{x}, \mathbf{x}_t) L(\mathbf{x}, \boldsymbol{\omega}) \\ &+ \int_{\mathbf{x}}^{\mathbf{x}_t} Tr(\mathbf{x}_{t'}, \mathbf{x}_t) \sigma_s(\mathbf{x}_{t'}) (\int_{S^2} \rho_p(\mathbf{x}_{t'}, \boldsymbol{\omega}, \boldsymbol{\omega}_i) L(\mathbf{x}_{t'}, \boldsymbol{\omega}_i) \mathrm{d}\boldsymbol{\omega}_i) \mathrm{d}\mathbf{x}_{t'} \end{aligned} \tag{3.15}$$

3.4 The Measurement equation

The rendering equations of surface(Equation 3.4) and medium(Equation 3.15) make it possible to compute the radiance coming from specific directions by simulating light transport. Then, to render a image, a virtual camera looking at the scene is introduced. Usually, the virtual camera consists of a grid of sensors each relating to a pixel in rendered image. The way they work is that, for each location on a sensor, it accumulate the radiance passing through it over all directions and use the results as the pixel intensity value. Radiance from each direction are not directly add up but weighted by an acceptance weight noted as $W_e(\mathbf{x}, \boldsymbol{\omega})$. This acceptance weight called importance describes how much of the radiance, that came from $\boldsymbol{\omega}$ and arrived at $\mathbf{x}$ on sensor, are actually captured by the sensor plane.

Thus, the intensity value of a pixel is calculated as follows:

$$I = \int_A \int_{H^2} W_e(\mathbf{x}, \boldsymbol{\omega}) L(\mathbf{x}, \boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \mathrm{d}\mathbf{x} \tag{3.16}$$

This equation is called the measurement equation,

3.5 The Path integral framework

According to the rendering equations, the radiance arriving at sensor plane can be computed by recursively tracing rays. In this way, new rays are traced into scattered directions when the former ray deflected by surface/medium until it hit the light source. Usually, we call a chain of traced ray from light source to camera(or from camera to light) as a light path, noted as $\bar{\mathbf{z}}$. The length of a light path varys from one to infinity large (it is possible that photon bounce around the scene but never hit the camera plane). Unfortunately, this recursive form makes it difficult to compute the pixel intensity, since the radiance can always be expanded to another integral. Also, if we solve this numerically, the number of light path grows exponentially due to this recursive structure.

However, another way to solve this, is transforming the measurement equation into a path integral formulation which origins from Veach (1997). The path integral formulation is written in the form

$$I = \int_\Omega f(\bar{\mathbf{z}}) \mathrm{d}\mu(\bar{\mathbf{z}}) \tag{3.17}$$

where Ω is space of all the paths of all lengths, $\mu(\bar{\mathbf{z}})$ is a measure on path $\bar{\mathbf{z}}$ and $f(\bar{\mathbf{z}})$ is the measurement contribution function. Since this paper mainly focus in volume rendering, we will review the path integral frame work briefly in terms of volume.

A light path $\bar{\mathbf{z}}$ with length h is defined to be a chain of vertices: $\mathbf{x}_0\mathbf{x}_1 \ldots \mathbf{x}_{h-1}$. This light path belongs to the sub path space Ω_h which includes all the paths with length h. The measure μ_h on this path is a product measure such that

$$\mathrm{d}\mu_h(\mathbf{x}_0 \ldots \mathbf{x}_k) = \mathrm{d}\mathcal{V}(\mathbf{x}_0) \times \cdots \times \mathrm{d}\mathcal{V}(\mathbf{x}_k)$$
$$\text{or } \mu_h = \underbrace{\mathcal{V} \times \cdots \times \mathcal{V}}.$$

Now the whole path space is defined as the union of path spaces of all possible path length:

$$\Omega = \cup_{i=1}^{\infty} \Omega_h.$$

Suppose D is a set of path of any length, then

$$\mu(D) = \cup_{i=1}^{\infty} \mu(D \cap \Omega_i)$$

For each light path there is a measurement contribution function that returns the intensity contributed by this path. By extracting proper term from volume rendering equation, the measurement contribution function is

$$\begin{aligned} f(\overline{\mathbf{z}}) = {} & L_e(\mathbf{x}_k, \boldsymbol{\omega}_k) G(\mathbf{x}_k, \mathbf{x}_{k-1}) W_e(\mathbf{x}_0, \boldsymbol{\omega}_1) \\ & \prod_{i=k-1}^{1} G(\mathbf{x}_i, \mathbf{x}_{i-1}) Tr(\mathbf{x}_i, \mathbf{x}_{i-1}) f_s(\mathbf{x}_{i+1}, \mathbf{x}_i, \mathbf{x}_{i-1}) \end{aligned}$$

where $f_s(\mathbf{x}_{i+1}, \mathbf{x}_i, \mathbf{x}_{i-1})$ is BSDFs when $\mathbf{x}_i$ is on surface and phase function when in medium.

Now the problem of light transport simulating has been transformed to solving this path integral. In next chapter, we will discuss in detail how this path integral is previously solved, and how we solve it.

Chapter 4

Solving Light Transport Problem in Participating Media

Last chapter, we reviewed the basics of light transports, and landed at simulating light transport by solving the path-integral formulation. To estimate an integration, usually we can use numeric method which divide the domain into sections and sum up value of each slices. However, this is not practical in rendering. The reason is that the radiance term in rendering equations could be recursively expressed by an integration with another radiance term in its integrand. In another word, solving the path integral is relating to integrating over a high dimensional domain.

Another way to estimate the value of integral is using Monte-Carlo estimators. In this section, we first show how to estimate the integral formulation in Monte-Carlo ways, and we will go through some popular Monte-Carlo method that are widely used in rendering including path tracing, bidirectional path tracing and photon mapping. Then we start to build the frame work of our solution on top of photon mapping and the path integral in extended path space.

4.1 The Monte-Carlo estimator

Monte-Carlo methods named after casino in Monte-Carlo use random numbers to solve numerical problems that are difficult with other approaches. Earlier, it was used in the development of atomic bomb and was first introduced in rendering by Cook(Cook et al. (1984)) to solve problems including depth of field and monition blur. Some good example references of using Monte-Carlo to solve rendering problems are Kajiya (1986b) and Veach and Guibas (1997).

When solving rendering problems, the value of interest is the pixel intensity which, by path integral formulation, equals to

$$I = \int_{\Omega} f(\bar{\mathbf{z}}) \mathrm{d}\mu(\bar{\mathbf{z}})$$

Monte-Carlo methods estimate the integral by replacing it with the mean of approximation of the integral at each random sample. We define a random variable $\bar{\mathbf{z}}$ on path space Ω with the probability density function of $p(\bar{\mathbf{z}})$, a path $\bar{\mathbf{z}}_j$ is an instance of $\bar{\mathbf{z}}$. Then $g(\bar{\mathbf{z}}) = \dfrac{f(\bar{\mathbf{z}})}{p(\bar{\mathbf{z}})}$ is a function of $\bar{\mathbf{z}}$ on the path space. According to the definition of the expected value

$$\begin{aligned} \mathbb{E}[g(\bar{\mathbf{z}})] &= \int_{\Omega} g(\bar{\mathbf{z}}) p_{\bar{\mathbf{z}}}(\bar{\mathbf{z}}) \mathrm{d}\mu(\bar{\mathbf{z}}) \\ &= \int_{\Omega} \frac{f(\bar{\mathbf{z}})}{p(\bar{\mathbf{z}})} p(\bar{\mathbf{z}}) \mathrm{d}\mu(\bar{\mathbf{z}}) \\ &= \int_{\Omega} f(\bar{\mathbf{z}}) \mathrm{d}\mu(\bar{\mathbf{z}}) \\ &= I \end{aligned}$$

Then the expected value of $g(\bar{\mathbf{z}})$ is approximated by taking N samples of random variable $\bar{\mathbf{z}}$ and computing the mean, a sample is note as $\bar{\mathbf{z}}_j$ where $1 \leqslant j \leqslant N$.

$$\mathbb{E}[g(\bar{\mathbf{z}})] = \mathbb{E}[\frac{f(\bar{\mathbf{z}})}{p(\bar{\mathbf{z}})}] \approx \frac{1}{N} \sum_{j=1}^{N} \frac{f(\bar{\mathbf{z}}_j)}{p(\bar{\mathbf{z}}_j)}$$

Then, the light transport problem expressed in the form of integral is transferred, by Monte-Carlo, to distribute functions that can be evaluated easily:

$$I = \int_{\Omega} f(\bar{\mathbf{z}}) \mathrm{d}\mu(\bar{\mathbf{z}}) \approx \frac{1}{N} \sum_{j=1}^{N} \frac{f(\bar{\mathbf{z}}_j)}{p(\bar{\mathbf{z}}_j)} = \langle I^N \rangle \tag{4.1}$$

where $\langle I^N \rangle$ notes the expect value of the pixel estimated by N samples. The larger the N is, the closer approximation the I^N is to I. By the law of large number, it's guaranteed that the approximation will converge to the exact value of path integral as $N \to \infty$.

4.1.1 Variance analysis

In order to assess the performance of Monte-Carlo estimator, variance and standard deviation are commonly used. We want to know what the variance of the estimated value look like and what factors influence the variance, so that we can improve the Monte-Carlo estimator further. We note the variance of the Monte-Carlo estimated value with N samples as $s^2[I^N]$ where s stands for the standard deviation. Here we also define a new random variable $Y = g(Z) = f(\overline{\mathbf{z}})/p(\overline{\mathbf{z}})$, and Y_j is an instance of Y that $Y_j = f(\overline{\mathbf{z}}_j))/p(\overline{\mathbf{z}}_j)$. Since the path samples are uncorrelated, the summation property holds for variance, then

$$\begin{aligned} s^2[\langle I^N\rangle] &= s^2[\frac{1}{N}\sum_{j=1}^{N}\frac{f(\overline{\mathbf{z}}_j)}{p(\overline{\mathbf{z}}_j)}] \\ &= \frac{1}{N^2}\sum_{j=1}^{N} s^2[\frac{f(\overline{\mathbf{z}}_j)}{p(\overline{\mathbf{z}}_j)}] \\ &= \frac{1}{N}s^2[Y] \end{aligned} \tag{4.2}$$

and the standard deviation is

$$s[\langle I^N\rangle] = \frac{1}{\sqrt{N}}s[Y] \tag{4.3}$$

According to Equation 4.2 and Equation 4.3, the variance of the estimated value goes down with a speed of $O(\sqrt{N})$, which means reducing the variance by a factor of two requires increasing the sample number by factor of four. Though it converge slowly, one advantage of using Monte-Carlo in rendering is its tolerance of high dimension samples: The numerical method which estimate the integral by summing up area of sections requires N^d samples when the sample goes to d dimensions, while the Monte-Carlo method only requires N.

4.1.2 Variance reduction

Since the radiance convergence rate is relatively slow, variance reduction strategies are used to compensate this down side of Monte-Carlo estimators. One thing we learn from those equation above is that the variance of the Monte-Carlo estimator is influence by the variance of $Y = f(\overline{\mathbf{z}})/p(\overline{\mathbf{z}})$.

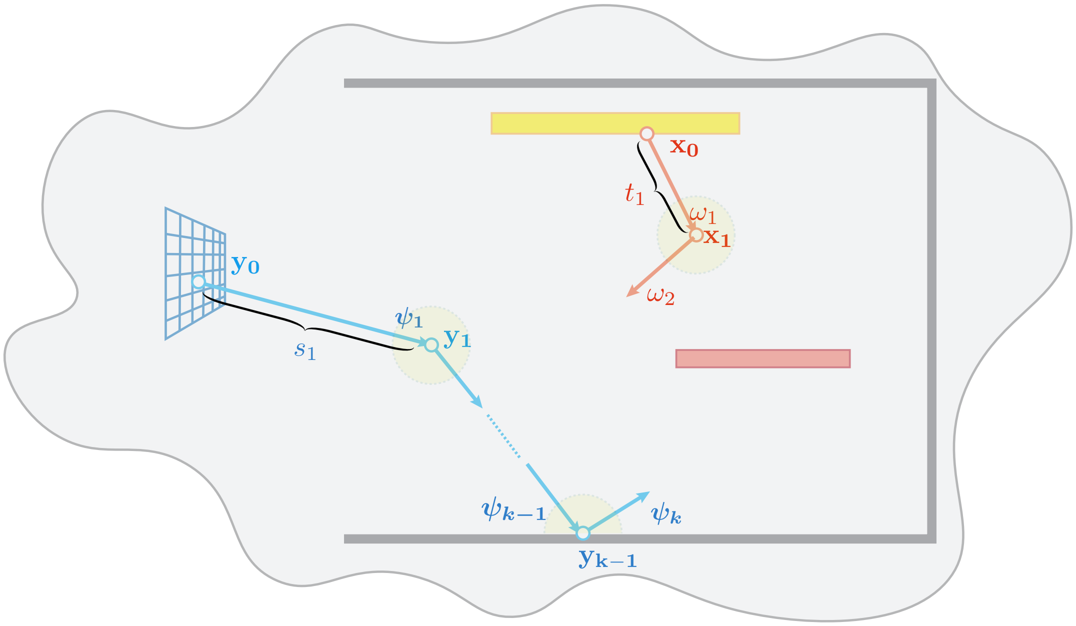

Figure 4.1: Random walks in the process of sampling a light path.

Importance sampling It can be observed from the variance expression that if Y becomes closer to constant, the variance will be smaller. As we have the freedom to choose from what distribution we sample the path, one way to reduce variance is making the sampling density function as close to the measurement contribution function as possible. Intuitively, this could be think as putting more samples in the area where the measurement contributes more. If we perfectly sample the path such that $p(\overline{\mathbf{z}}) = cf(\overline{\mathbf{z}})$, then variance becomes

$$s^2[\frac{f(\overline{\mathbf{z}})}{cp(\overline{\mathbf{z}})}] = s^2[\frac{\cancel{f(\overline{\mathbf{z}})}}{c\cancel{f(\overline{\mathbf{z}})}}] = s^2[\frac{1}{c}] = 0 \tag{4.4}$$

Equation 4.4 proves that if the probability density function is carefully chosen such that it perfectly match the shape of measurement contribution function, in theory we can reduce the variance of Monte-Carlo estimator to zero. However, computing this constant c contains solving $\int cf(\overline{\mathbf{z}})\mathrm{d}\mu(\overline{\mathbf{z}}) = 1$, and this seems to form a dead lock. However, although it is difficult to make the PDFs exactly the same shape as the measurement contribution function, if we have some knowledge of the structure of f, we can make the PDFs as close to f as possible to reduce the variance.

4.2 Sampling a light path

As mentioned previously, the Monte-Carlo estimator use random variables to solve the numeric problem. In this light transport simulation problem the random variables refer to the light path $\bar{\mathbf{z}}$. Actually, a light path is a high dimension variable that consists of a chain of random processes including sampling on light source/camera plane, sampling emission direction/importance direction. sampling propagation distances in medium, sampling scattering directions and a random process on whether terminate the path or not.

In this subsection, we will review the random processes that used to generate a light path, random variables in each random process and the probability density function of a sampled path.

4.2.1 Random variables

Sampling on light source A sampled path starts from a random point on light source and then propagates along a sampled emission direction. If the light source is not a point light, then starting point is usually uniformly sampled with respect to the area/volume of light. By importance sampling, its propagation direction is sampled proportion to the distribution of emission function. For example, for a area light emitting cosine distributed light, we usually chose to sample with $p(\boldsymbol{\omega}_l) = \cos\theta$, where θ is the angle between sampled direction $\boldsymbol{\omega}_l$ and surface normal $\mathbf{n}$.

Sampling on camera A path sample could also starts from a random point on sensor plane and trace along a sampled direction. Similarly, The point on the sensor plane could simply generated from uniform distribution, since the importance is invariant to the point on aperture. When choosing out going direction of a ray, we could choose to sample with respect to the importance function W_e or the reconstruction filter.

Sampling scattering directions After a light or a ray is generated. It travels along sampled direction in the medium. It may be defected into another direction by medium before hitting any surface, or it scattered into another direction after hitting a surface. The direction that the light get scattered into is sampled with respect to the distribution of phase functions for scattering event happens in medium and BSDFs for surface scattering.

Sampling propagation distance Whether the light get scattered in medium or not is determined by the process of sampling a propagation distance, usually referred as free fly distance, in the medium. It works in this way: After a propagation direction is sampled, we sampled a distance t that the light could travel before deflect in the medium. If t is smaller than the distance between $\mathbf{x}_i$ and the nearest surface, then the propagation distance equals to t and scattering point $\mathbf{x}_{i+1}$ is in medium, otherwise the propagation distance equals to the distance from $\mathbf{x}_i$ to nearest surface and $\mathbf{x}$ lays on surface. When sampling the free fly distance, we usually choose to sample proportion to the transmittance term Equation 3.14.

Terminate a path A path terminate after sliding into light source for path starting from camera or camera senor for path traced from light source. A path also terminate when it exits the scene. It is possible that a path keep bounce around in the scene and never stops, one way to avoid this is to add a random process after each bounce on whether terminate this path or not. This random process of termination decision is called Russian. With the internal goal of making the path PDF as close to the contribution function as possible, the distribution of this random variable is carefully designed to follow the shape of contribution function diving all the other terms used in previous random process.

4.2.2 Path probability density function

For a path $\bar{\mathbf{z}} = \mathbf{x}_0\mathbf{x}_1 \ldots \mathbf{x}_l$, the probability density function is a joint probability of generating all the points along this path

$$p(\bar{\mathbf{z}}) = p(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_l)$$

Traced from light source, this probability could be decomposed into products of conditional probabilities like

$$p(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_l) = p(\mathbf{x}_0)\, p(\mathbf{x}_1 \mid \mathbf{x}_0) \ldots p(\mathbf{x}_l \mid \mathbf{x}_{l-1} \ldots \mathbf{x}_1\mathbf{x}_0)$$

Different decomposition of the joint probability relates to different strategy of generating the path sample. In the following sections, we will review a few path sampling strategies including path tracing, bidirectional path tracing and smoothly move towards our solutions from photon mapping.

4.3 Path tracing

One straight forward way of generating a path sample is tracing ray starting from a point on camera $\mathbf{y}_0$ and recursively shooting new rays at scattering points $\mathbf{y}_j$ until the path hit the light source, get out of the scene or terminated by the Russian Roulette. Following the process of how a path is generated by path tracing, the probability density function of a sampled path with length $k+1$ is expressed as $\bar{\mathbf{z}} = \mathbf{y}_0 \dots \mathbf{y}_k$ and its probability density function is

$$p(\bar{\mathbf{z}}) = p(\mathbf{y}_0 \dots \mathbf{y}_k) = p(\mathbf{y}_0)p(\mathbf{y}_1 \mid \mathbf{y}_0)\dots p(\mathbf{y}_k \mid \mathbf{y}_{k-1} \dots \mathbf{y}_1 \mathbf{y}_0) \tag{4.5}$$

Path tracing does not call for a random process of sampling a point on the light source, thus the path may never hit the light source. This made path tracing perform badly in the scene with point light sources in which it is impossible to trace a ray that hit the light source. Another simple way to sample a path is reverse form of path tracing which start the random walk from the light, named light tracing, however, since this strategy don't sample a point on sensor plane, light scattered from specular surfaces hardly get into the camera plane directly, thus it can not handle the scenes with specular surfaces.

4.3.1 Next event estimate

A a technic named Next Event Estimate (NEE) is used to improve path tracing as well as light tracing so that they can handle most of the light path type. With next event estimate, a sample on the light source $\mathbf{x}_l$ is generated and directly connected to the vertices along the sampled path, with this setting, a path with length $k+1$ is expressed as $\bar{\mathbf{z}} = \mathbf{y}_0 \dots \mathbf{y}_{k-1}\mathbf{x}_0$ the probability of a path is decomposed in the following way:

$$\begin{aligned} p(\bar{\mathbf{z}}) &= p(\mathbf{y}_0 \dots \mathbf{y}_{k-1}\mathbf{x}_0) \\ &= p(\mathbf{y}_0)p(\mathbf{y}_1 \mid \mathbf{y}_0)\dots p(\mathbf{y}_{k-1} \mid \mathbf{y}_{k-2} \dots \mathbf{y}_1 \mathbf{y}_0)p(\mathbf{x}_0) \end{aligned} \tag{4.6}$$

Similarly, if we do next event estimate for light tracing, which directly connect each point along the path with a random point on camera sensor, we will have the PDF for a path $\bar{\mathbf{z}} = \mathbf{y}_0\mathbf{x}_{k-1} \dots \mathbf{x}_0$ equals to

$$\begin{aligned} p(\bar{\mathbf{z}}) &= p(\mathbf{x}_0 \dots \mathbf{x}_{k-1}\mathbf{y}_0) \\ &= p(\mathbf{x}_0)p(\mathbf{x}_1 \mid \mathbf{x}_0)\dots p(\mathbf{x}_{l-1} \mid \mathbf{x}_{l-2} \dots \mathbf{x}_1 \mathbf{x}_0)p(\mathbf{y}_0) \end{aligned} \tag{4.7}$$

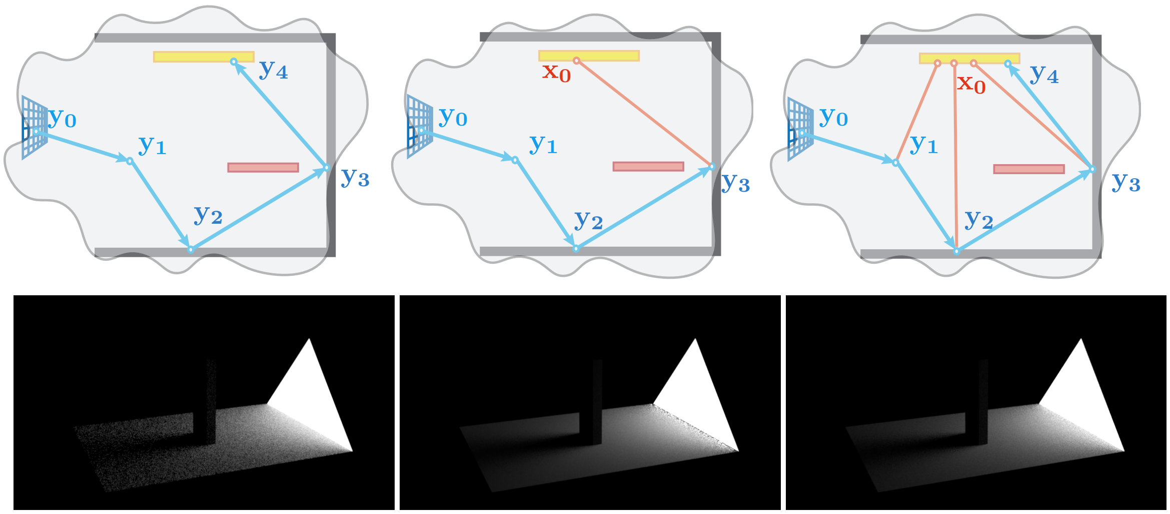

Figure 4.2: Path tracing, next event estimation (NEE), multiple importance sampling among path tracing and NEE (first row) and rendering examples (second row).

Although the next event estimator prevent the path tracing from never hitting a light source, it performs badly when the scattering point gets close to the emission surface due to the geometry term in Equation 3.5. Fortunately, path tracing without next event estimate performs well under this situation since it's easy for a sampled scattering direction to slide in to an emission surface near by. Since path tracing with and without NEE show advantage in different area over the other, a combination of these two strategies is necessary and beneficial.

4.4 Multiple importance sampling

In the last section, we want to combine the path sampling strategy which uses NEE with the strategy that don't use NEE to improve the path tracing. Observing that only applying straight averaging to combine two strategies will keep the bad effects from both sides, to address this, multiple importance sampling (MIS) is used.

Multiple importance sampling is a technique introduced by Veach and Guibas (1995) which uses the combination of multiple sampling strategies to improve the Monte-Carlo estimating where the integrand is high-dimension, discontinues or has singularities. Readers who wants more detail of Multiple importance sampling can go to Veach and Guibas (1995) to read the full derivation and analysis. Here we just pattern match our path integral problem into the general interpretation of multiple importance sampling. Suppose m is the number of strategies we use to evaluate the integral, n_i is the number of samples we

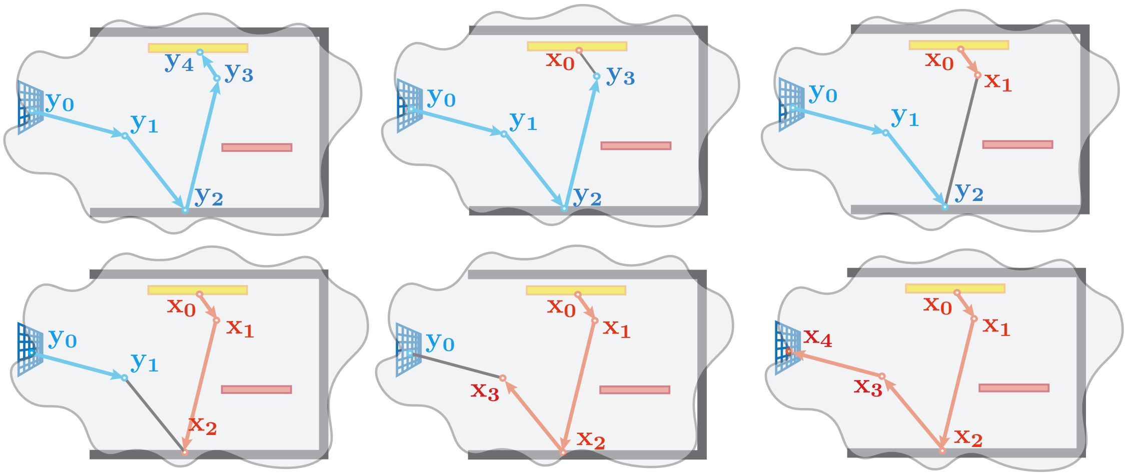

Figure 4.3: Bidirectional path tracing

take from each strategy, then the robust estimate of I with Multiple Importance Sampling should be

$$I = \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} w_{i,j} \frac{f(\overline{\mathbf{z}}_{i,j})}{p_i(\overline{\mathbf{z}}_{i,j})} \tag{4.8}$$

where $\overline{\mathbf{z}}_i, j$ represents the jth sample we take from strategy i, and $w_{i,j}$ is the MIS weight of jth sample from ith strategy. By balance heuristic, the weight is

$$w_{i,j} = \frac{n_i p_i(\overline{\mathbf{z}}_{i,j})}{\sum_{k=1}^{m} n_k p_k(\overline{\mathbf{z}}_{i,j})} \tag{4.9}$$

In the weight expression above, $p_k(\overline{\mathbf{z}}_{i,j})$ is the probability of generating exactly the same path as $\overline{\mathbf{z}}_{i,j}$ by strategy k.

To obtain robust estimation of pixel intensity, we can combine NEE with path tracing by multiple importance sampling. As is illustrated in section 4.3. a path $\overline{\mathbf{z}}$ can be generated by path tracing with a probability of $p_{PT}(\mathbf{y}_0 \ldots \mathbf{y}_k)$ and also by NEE with a probability of $p_{NEE}(\mathbf{y}_0 \ldots \mathbf{y}_{k-1}\mathbf{x}_0)$. At each evaluating point, we take one sample from NEE and one sample from simple path tracing, and weighed contribution from each path by Equation 4.9.

4.5 Bidirectional path tracing

Except for generating the path sample from one side, we can also generating the path samples by tracing a light subpath with length l from light and a sensor subpath from camera side with length $h-l-1$, and then connecting each vertices on light subpath with vertices on camera subpath. This could also be think of as decomposing the probability of path $\bar{\mathbf{z}}$ from two directions instead of decomposing in only one direction (path tracing/-light tracing). This path sampling method is referred as Bidirectional Path Tracing (BPT). If we consider the bidirectional path tracing as a path sampling strategy, then the path probability density function could be written as:

$$\begin{aligned} p(\bar{\mathbf{z}}) &= p(\mathbf{x}_0 \ldots \mathbf{x}_l \mathbf{y}_{k-1} \mathbf{y}_0) \\ &= p(\mathbf{x}_0 \ldots \mathbf{x}_l) p(\mathbf{y}_{k-1} \cdots \mathbf{y}_0) \\ &= p(\mathbf{x}_0) p(\mathbf{x}_1 \mid \mathbf{x}_0) \ldots p(\mathbf{x}_l \mid \mathbf{x}_{l-1} \ldots \mathbf{x}_1 \mathbf{x}_0) p(\mathbf{y}_{k-1} \mid \mathbf{y}_{k-2} \ldots \mathbf{y}_1 \mathbf{x}_0) \cdots p(\mathbf{y}_0) \end{aligned} \tag{4.10}$$

By varying l in $[0, h]$, a light path $\bar{\mathbf{z}}$ with length of h can be decomposed in h+1 ways where $l = 0$ is actually a light tracing strategy and $l = h$ relates to simple path tracing. Since these different decomposition of generating the same path sample relates to different composition of path probability, we can think each decomposition as a type of path sampling strategy. Each strategy has advantage over others in some area, and then utilize MIS to reach a more robust estimation of path integral with bidirectional path tracing.

4.6 Photon mapping

Although bidirectional path tracing provides robust estimation of path integral, there are still some light paths, like specular-diffuse-specular paths, that are tricky for bidirectional path tracing. For example, it's still challenge to render caustics under the water.

A two-pass global illumination algorithm named Photon Mapping was developed by Henrik Wann Jensen et al. to approximate solutions to rendering equation. By photon mapping, rendering caustics is possible. The photon mapping algorithm work in this way: In first pass, it deploy photons from light into the scene to represent indirect illumination and store their info like position and energy in data structures for later use. In the second pass, rays are traced from camera to query photons. Usually, at each query point, it use a blurring kernel to collect photons and return the average radiance of photons within the kernel. This blurring kernel add additional bias into the estimation, however, by properly

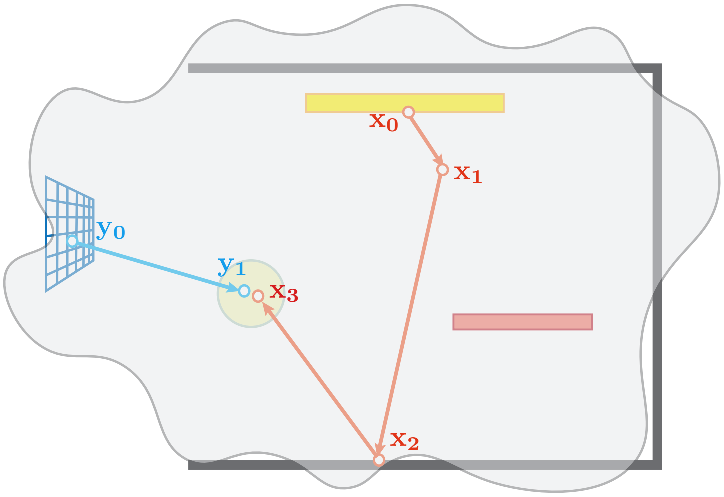

Figure 4.4: Photon mapping

choosing the blurring kernel and carefully control the bias, this sacrifice is worthy to take.

A path in photon mapping with length h is expressed as $\mathbf{x}_0 \ldots \mathbf{x}_l \mathbf{y}_k \ldots \mathbf{y}_0$ where $\mathbf{x}_l \mathbf{x}_k$ are blended by the blurring kernel. Then probability of a light path with length h is decomposed:

$$p(\overline{\mathbf{z}}) = p(\mathbf{x}_0 \ldots \mathbf{x}_l)\, p(\mathbf{y}_k \ldots \mathbf{y}_0) \tag{4.11}$$

Different from bidirectional path tracing, the photon mapping estimator has an additional term $K(\mathbf{x}_l, \mathbf{y}_k)$ relating to blurring in measurement contribution function. It can also be think of as the probability of taking a point within the kernel and its value identical to the inverse volume of the kernel. Another important point is, photon mapping is demonstrated in a different path space because it has an additional point at the connection of two subpaths while bidrectional path tracing couples subpaths directly by shadow connection. This additional point makes it tricky to describe a path generated by photon mapping in standard path space and additional care is needed. Then the extended path space (Georgiev et al. 2012; Hachisuka et al. 2017) which can express photon mapping in path integral framework is used. By the path integral in extended path space, photon mapping is able to MIS with path tracing as well as bidirectional path tracing, etc.

Next we will introduce the extended path space and describe photon mapping in the form of extended path integral. Although the photon mapping itself is biased method, we will show how we make the it unbiased with our photon surfaces estimator by carefully configure the blurring kernel. Noticing that, for the sake of simplicity, we assume all the medium used in this paper are homogeneous. We can easily expand the theory to sup-

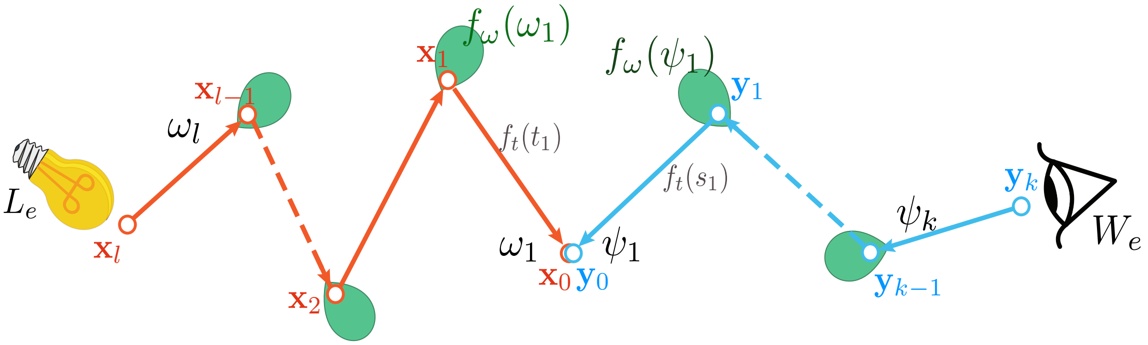

Figure 4.5: Path integral in extended path space

port heterogeneous by Woodcock/delta tacking once we have solution for homogeneous medium.

4.7 Path integral in extended path space

Now we introduce the framework we use to derive our photon surfaces estimators in this paper starting from redefining the path integral in extended path spaceHachisuka et al. (2017).

Recall that, with the path integral frame work provided by Veach (1997), the intensity of a pixel is evaluated by an integration:

$$I = \int_{\mathbf{\Omega}} f(\bar{\mathbf{z}}) d\mu(\bar{\mathbf{z}}) \tag{4.12}$$

where $\mu(\bar{\mathbf{z}})$ is the measure of a full path from light source to camera plane, $\mathbf{\Omega}$ is the space of all possible paths, and $\bar{\mathbf{z}}$ is the measurement contribution function along a full path.

Change of notation We derive our estimator by photon mapping, therefore, initially we defined a full light path in extended path space as $\bar{\mathbf{z}} = \bar{\mathbf{x}}\bar{\mathbf{y}}$ where $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ refers to the photon subpath and the camera subpath respectively Hachisuka et al. (2017). In order to keep the simplicity of the equations, starting from this section, we reverse the index numbering of the vertex on light paths to start from the connection of photon and camera subpaths. As is shown in Figure 4.5, by this way of indexing, $\mathbf{x}_l$ refers to the point on light, $\mathbf{y}_k$ refers to the pointon sensor plane, and $\mathbf{x}_0$ and $\mathbf{y}_0$ are vertices at the connection of photon subpath and camera subpath.

Path Shown in figure Figure 4.5, on photon subpath, photon goes from $\mathbf{x}_i$ to $\mathbf{x}_{i-1}$ along direction $\boldsymbol{\omega}_i$ by distance t_i while a query beam goes from $\mathbf{y}_i$ to $\mathbf{y}_{i-1}$ along direction $\boldsymbol{\psi}_i$ by distance s_i. These can be expressed in formula:

$$\mathbf{x}_0 = \mathbf{x}_l + \prod_{i=l}^{1} t_i \boldsymbol{\omega}_i \qquad \mathbf{y}_0 = \mathbf{y}_l + \prod_{i=k}^{1} s_i \boldsymbol{\psi}_i \tag{4.13}$$

Then we describe our path by directions and distance as:

$$\bar{\mathbf{z}} = \mathbf{x}_l \overline{\boldsymbol{\omega}}\, \bar{t}\, \overline{\boldsymbol{\psi}}\, \bar{s}\, \mathbf{y}_k$$

where

$$\overline{\boldsymbol{\omega}} = \boldsymbol{\omega}_l \ldots \boldsymbol{\omega}_1 \qquad \bar{t} = t_l \ldots t_1 \qquad \overline{\boldsymbol{\psi}} = \boldsymbol{\psi}_k \ldots \boldsymbol{\psi}_1 \qquad \bar{s} = s_k \ldots s_1$$

We refer the offset vector between the end points of two subpaths as $\mathbf{g}$ where $\mathbf{g} = \mathbf{x}_0 - \mathbf{y}_0$.

Space Let $\boldsymbol{\Omega}_h$ denotes the whole set of full paths with length h, where $0 < h < \infty$. Then

$$\boldsymbol{\Omega} = \cup_{i=1}^{\infty} \boldsymbol{\Omega}_h$$

is the path space that represents the union of spaces of all path lengths.

Measurement We define a measure μ_h on a paths set D where $D \subseteq \boldsymbol{\Omega}_h$ as a product measure such that

$$\begin{aligned}\mu_h(D) = \int_D \mathrm{d}\mathscr{A}(\mathbf{x}_l) \times \underbrace{\mathrm{d}\boldsymbol{\omega}_l \times \cdots \times \mathrm{d}\boldsymbol{\omega}_1}_{l \text{ times}} \times \underbrace{\mathrm{d}t_l \times \cdots \times \mathrm{d}t_1}_{l \text{ times}} \times \\ \underbrace{\mathrm{d}\boldsymbol{\psi}_l \times \cdots \times \mathrm{d}\boldsymbol{\psi}_1}_{k \text{ times}} \times \underbrace{\mathrm{d}s_l \times \cdots \times \mathrm{d}s_1}_{k \text{ times}} \mathrm{d}\mathscr{A}(\mathbf{y}_l)\end{aligned}$$

In another word:

$$\begin{aligned}\mathrm{d}\mu_h(\mathbf{x}_l \overline{\boldsymbol{\omega}}\, \bar{t}\, \overline{\boldsymbol{\psi}}\, \bar{s}\, \mathbf{y}_k) = \mathrm{d}\mathscr{A}(\mathbf{x}_l) \times \underbrace{\mathrm{d}\boldsymbol{\omega}_l \times \cdots \times \mathrm{d}\boldsymbol{\omega}_1}_{l \text{ times}} \times \underbrace{\mathrm{d}t_l \times \cdots \times \mathrm{d}t_1}_{l \text{ times}} \times \\ \underbrace{\mathrm{d}\boldsymbol{\psi}_l \times \cdots \times \mathrm{d}\boldsymbol{\psi}_1}_{k \text{ times}} \times \underbrace{\mathrm{d}s_l \times \cdots \times \mathrm{d}s_1}_{k \text{ times}} \mathrm{d}\mathscr{A}(\mathbf{y}_l)\end{aligned} \tag{4.14}$$

To keep things brief, we aggregated the equation above by defining a parameter set $\overline{\xi} = \{\mathbf{x}_l, \overline{\boldsymbol{\omega}}, \overline{t}, \overline{\boldsymbol{\psi}}, \overline{s}, \mathbf{y}_k\}$ whose elements are the integration domains in Equation 4.14:

$$\mathrm{d}\mu_h(\overline{\xi}) = \prod_{\xi_i \in \overline{\xi}} \mathrm{d}\xi_i$$

We extend our paths set D to include any possible finite length path, and the measurement on D is the sum of the measure on the path sets of each length:

$$\mu(D) = \sum_{i=1}^{\infty} \mu(D \cap \boldsymbol{\Omega}_i)$$

Measurement Contribution Function The measurement contribution function along a full path $\boldsymbol{\Omega}$ is the product of the contribution of light and camera subpaths coupled by a scattering phase function and a world space 3D blurring kernel:

$$f(\bar{\mathbf{z}}) = f(\bar{\mathbf{x}}) K(\mathbf{g}) f_{\boldsymbol{\omega}}^{1,1} f(\bar{\mathbf{y}})$$

In the equation above, the contribution from photon subpath is

$$f(\bar{\mathbf{x}}) = \prod_{i=l}^{1} f_{\omega}(\boldsymbol{\omega}_i) f_t(t_i)$$

where the $f_{\omega}(\boldsymbol{\omega}_i)$ is emission function when $i = l$, scattering phase function when $\mathbf{x}_i$ is in the medium and BRDF when $\mathbf{x}_i$ is on a surface.

$$f_{\omega}(\boldsymbol{\omega}_i) = \begin{cases} L_e(\mathbf{x}_i, \boldsymbol{\omega}) \cos\theta & i = l \\ \sigma_s \rho_p(\boldsymbol{\omega}_{i+1}, \boldsymbol{\omega}_i) & i \text{ is in the medium.} \\ \rho_s(\boldsymbol{\omega}_{i+1}, \boldsymbol{\omega}_i) & i \text{ is on the surface.} \end{cases}$$

And the $f_t(t_i)$ is the transmittance term

$$f_t(t_i) = Tr(t_i) V(\mathbf{x}_i, \mathbf{x}_{i-1}) \qquad \text{with} \qquad Tr(t) = e^{-t_i \sigma_t}$$

The contribution from camera subpath is computed similar as that of photon subpath since they are symetric except that the camera subpath emits importance $W_e(\mathbf{y}_k, \boldsymbol{\psi}_k)$ instead of emittance.

Recall that we have integration domain set $\overline{\xi}$, which is also a set of variables that determine the path. For the sake of later derivation, we express our measurement contribution function in terms of ξ:

$$f(\overline{\mathbf{z}}(\overline{\xi})) = f(\overline{\xi}) K(\mathbf{g}(\overline{\xi})) f_{\boldsymbol{\omega}}^{1,1}$$

where $f(\overline{\xi}) = f(\bar{\mathbf{x}}) f(\bar{\mathbf{y}}) = \prod_{\xi \in \overline{\xi}} f(\xi_i)$, with

$$f(\xi_i) = \begin{cases} f_\omega(\xi_i) & \text{if } \xi_i \in \{\mathbf{x}_l, \mathbf{y}_k, \overline{\boldsymbol{\psi}}, \overline{\boldsymbol{\omega}}\} \\ f_t(\xi_i) & \text{if } \xi_i \in \{\overline{s}, \overline{t}\} \end{cases}$$

Then the path integral becomes:

$$I = \int_{\mathbf{\Omega}} f(\overline{\xi})\, K(\mathbf{g}(\overline{\xi}))\, f_{\boldsymbol{\omega}}^{1,1}\, \mathrm{d}\overline{\xi} \tag{4.15}$$

Monte-Carlo estimator in photon mapping The integration in Equation 4.15 can be estimated by Monte-Carlo estimator that

$$I \approx \frac{f(\overline{\xi})\, K(\mathbf{g}(\overline{\xi}))\, f_{\boldsymbol{\omega}}^{1,1}}{p(\overline{\xi})}$$

in which $p(\overline{\xi})$ is the joint probability of getting the specific value of variables in $\overline{\xi}$.

Chapter 5

The Photon Surfaces Estimator

5.1 Method overview

We only focus on unbiased density estimator in this paper. However, starting with photon mapping usually means there is a blurring kernel that makes the estimator biased. With the goal of deriving unbiased estimators, we shrink the 3D blurring kernel to a 3D Dirac delta kernel where the full light path contributes to the pixel only when its photon and camera subpaths exactly attach at $\mathbf{x}_0\mathbf{y}_0$, which means $\mathbf{g}(\bar{\xi}) = \mathbf{0}$. In this way, we know that the light path created by photon mapping in extended space is identical to a full light path going from the light source to the camera plane in standard path space. The Dirac delta kernel in three-dimension equals to the product of three one-dimensional Dirac delta functions along three orthogonal axes:

$$K(\mathbf{g}) = \boldsymbol{\delta}^3(\mathbf{g}) = \boldsymbol{\delta}^1(x(\mathbf{g}))\boldsymbol{\delta}^1(y(\mathbf{g}))\boldsymbol{\delta}^1(z(\mathbf{g}))$$

where $x-$, $y-$ and $z-$ represent three axes of standard Cartesian coordinates and $x(\mathbf{g}), y(\mathbf{g}), z(\mathbf{g})$ are the projections of vector $\mathbf{g}$ on those axes. According to Equation 4.13, $\mathbf{g}$ is determined by all the variables in $\bar{\xi}$:

$$\mathbf{g}(\bar{\xi}) = \mathbf{x}_0(\bar{\xi}) - \mathbf{y}_0(\bar{\xi}) = \left(\mathbf{x}_l + \sum_{i=l}^{1} t_i \boldsymbol{\omega}_i\right) - \left(\mathbf{y}_k + \sum_{i=k}^{1} s_i \boldsymbol{\psi}_i\right).$$

Here comes the key problem, in practice, with the delta kernel, it is impossible to sample a path that contributes non zero value to the pixel intensity by Monte-Carlo method. Our solution to this is splitting our extended space integration in Equation 4.15 into two parts:

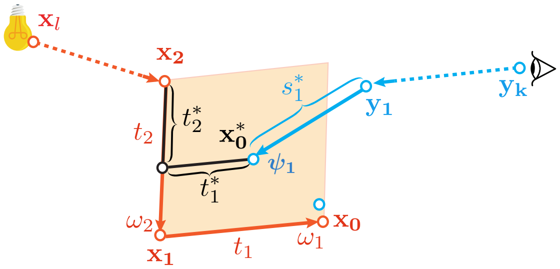

Figure 5.1: The original 0D photon plane estimator

One takes three integration dimensions, which referred as $\overline{\xi_a}$, from $\overline{\xi}$. We analytically integrate on $\overline{\xi_a}$ and alleviate the delta kernel out. The other part contains all the rest integration dimensions named $\overline{\xi_n}$, which is identical to $\overline{\xi}/\overline{\xi_a}$. Now we can express our measurement contribution function as a product of contribution depending on $\overline{\xi_a}$ and the contribution depending on $\overline{\xi_n}$:

$$f(\overline{\xi}) = f(\overline{\xi_a})f(\overline{\xi_n})$$

The extended path integral in Equation 4.15 can therefore be written as:

$$I = \int_{\mathbf{\Omega}} f(\overline{\xi_n}) \underbrace{\int_{\mathbf{\Omega}_a(\overline{\xi_n})} f(\overline{\xi_a})\, \boldsymbol{\delta}^3(\mathbf{g}(\overline{\xi_a}))\, f_{\boldsymbol{\omega}}^{1,1}\, \mathrm{d}\overline{\xi_a}}_{I_a} \mathrm{d}\overline{\xi_n}, \tag{5.1}$$

where $\mathbf{g}(\overline{\xi_a})$ is a shorthand for $\mathbf{g}(\overline{\xi_a}, \overline{\xi_n})$ because we only manipulate on the analytical part. With this new decomposition of path contribution, we use Monte-Carlo after analytic integration to evaluate Equation 5.1, then

$$I = \frac{f(\overline{\xi_n})I_a(\overline{\xi_n})}{p(\overline{\xi_n})}$$

where $I_a(\overline{\xi_n})$ is the analytically-preintegrated part in Equation 5.1 with in the brace, and $p(\overline{\xi_n})$ is the joint probability density function of sampling a random variable set $\overline{\xi_n}$.

Next section, we will show how to derive a photon 0D-plane estimator in our framework and extended it to more general cases.

5.2 Re-derive 0D-photon planes

The 0D-photon plane estimator Bitterli and Jarosz (2017) employs edge $\mathbf{x}_2\mathbf{x}_1$ and $\mathbf{x}_1\mathbf{x}_0$ along photon subpath to form a photon plane and use edge $\mathbf{y}_0\mathbf{y}_1$ as query beam. In our framework, this is equivalent to choosing $\overline{\xi_a} = \{t_1, t_2, s_1\}$ as analytic integration dimensions, and its analytic path integral I_a is

$$I_a = \int_{\mathbf{\Omega}_a} f_t(t_2)\, f_t(t_1)\, f_t(s_1)\, \boldsymbol{\delta}^3(\mathbf{g}(\overline{\xi_a}))\, f_{\boldsymbol{\omega}}^{1,1}\, \mathrm{d}t_2 \mathrm{d}t_1 \mathrm{d}s_1. \tag{5.2}$$

In order to analytically compute this integration, we express the delta function in terms of $\overline{\xi_a}$. With this specific choice of $\overline{\xi_a}$, the $\mathbf{g}(\{t_2, t_1, s_1\})$ is an one-to-one function on $\{t_2, t_1, s_1\}$:

$$\mathbf{g}(\overline{\xi_a}) = \underbrace{(\mathbf{x}_2 + \boldsymbol{\omega}_2 t_2 + t_1\boldsymbol{\omega}_1)}_{\mathbf{x}_0(t_2,t_1)} - \underbrace{(\mathbf{y}_1 + s_1\boldsymbol{\psi}_1)}_{\mathbf{y}_0(s_1)} = \mathbf{x}_2 - \mathbf{y}_1 + \mathbf{A}\overline{\xi_a}$$

where the matrix $\mathbf{A} = [\boldsymbol{\omega}_2, \boldsymbol{\omega}_1, \boldsymbol{\psi}_1]$. Moreover, $\mathbf{g}(\overline{\xi_a}) = \mathbf{0}$ only have one root for this choice of $\overline{\xi_a}$, then by the function composition property of Dirac delta function, we can replace the delta function of $\mathbf{g}$ with a delta function of $\overline{\xi_a}$ as follow:

$$\boldsymbol{\delta}^3(\mathbf{g}(\overline{\xi_a})) = \frac{\boldsymbol{\delta}^3(\overline{\xi_a} - \overline{\xi_a^*})}{\left|\frac{\partial \mathbf{g}}{\partial \overline{\xi_a}}(\overline{\xi_a^*})\right|} = \frac{\boldsymbol{\delta}^3(\overline{\xi_a} - \overline{\xi_a^*})}{\mathbf{J}^{\mathbf{g}}_{\overline{\xi_a}}(\overline{\xi_a^*})} \tag{5.3}$$

where $\overline{\xi_a^*} = \{t_2^*, t_1^*, s_1^*\}$ is the root of equation $\mathbf{g}(\overline{\xi_a}) = \mathbf{0}$. In this photon plane case, it is the intersection point between the query beam and photon plane, as shown in the figure Figure 5.1. The denominator in Equation 5.3 is the change-of-variable Jacobian $\mathbf{J}^{\mathbf{g}}_{\overline{\xi_a}}$ from $\overline{\xi_a}$ to $\mathbf{g}$, and it is equals to the determinant of matrix $\mathbf{A}$:

$$\mathbf{J}^{\mathbf{g}}_{t_2,t_1,s_1}(\overline{\xi_a^*}) = \det(\mathbf{A}) = |(\boldsymbol{\omega}_2 \times \boldsymbol{\omega}_1) \cdot \boldsymbol{\psi}_1|. \tag{5.4}$$

Replacing the delta function in Equation 5.2 with Equation 5.3 allows us to preintegrate over $\overline{\xi_a}$ and sweep out the delta kernel. Then we yielded

$$I_a = \frac{f_{\boldsymbol{\omega}}^{1,1}\, f_t(t_2^*) f_t(t_1^*) f_t(s_1^*)}{\left|(\boldsymbol{\omega}_2 \times \boldsymbol{\omega}_1) \cdot \boldsymbol{\psi}_1\right|}. \tag{5.5}$$

The equation above exactly matches the expression of 0D photon plane estimator in Bitterli and Jarosz (2017) accounting for different notations.

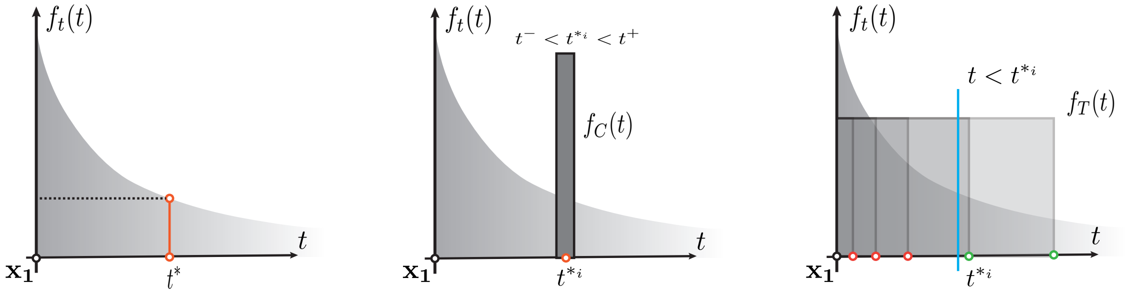

Figure 5.2: Expected value (right), collision estimator (middle) and tracklength estimator (left): Expected value directly compute the transmittance; collision estimator check if the sample t^{*i} falls in the interval $[t^-, t^+]$; track-length estimator check if the sample go beyond some distance t.

5.2.1 The transmittance estimators

Shown in Equation 5.5, when evaluating contribution of a path, there are transmittance related terms like $f_t(t_1^*)$ that defers for each hitting point. In fact, the parametric 2D photon plane formed by sweeping all possible combination of t_1, t_2 is an semi-infinite plane. Its costly to evaluate the transmittance using expected value (Spanier 1966) for each hitting points. The original photon mapping method estimates the transmittance with "collision" estimator (Spanier and Gelbard 1969). (Jarosz et al. 2011a) replace the expected value of transmittance with an unbiased estimation produced with "track-length" estimator (Spanier 1966), which replaced the semi-infinite "long" beam with a finite "short" beam. Later, Bitterli and Jarosz (2017) also use track-length estimator to substitute semi-infinite plane with finite planes.

To be precise, here we briefly review the collision estimation, track-length estimation and expected value in an one dimension setting (Figure 5.2). Let light beams emit from $\mathbf{x}_1$ and travel along direction $\boldsymbol{\omega}$ in a medium. The energy lose of light beam is expressed in terms of transmittance, and the relation between the transmittance and the distances t is described by Equation 3.14. To simulate the transmittance, we can evaluate its value directly at each location t^* by Equation 3.14, or use samples who has constant transmittance value but distributed proportional to the transmittance term. t^{*i} denotes the ith sampled distance along $\boldsymbol{\omega}$.

Expected value Shown in Figure 5.2, when using expect value, it return the analytic value of the transmittance computed by Equation 3.14 at each query distance.

The collision estimator Collision estimator use sample point with constant contribution value as an estimation of beam. We can think it as concentrating all the energy of the infinite photon beam to the sampled location; since the probability density function we take for distance sampling is proportion to the transmittance function, large number of samples ensure the estimation converge to the expected value. However, in practise, a point estimation is replaced by a blurring kernel, and the energy is equally distributed within the kernel. Let t^- and t^+ be the lower and upper bound of the kernel, for one sampled distance t^{*i}, its contribution is

$$f_{Ci}(t) = \begin{cases} \frac{1}{|t^+ - t^-|\sigma} & t^- \leqslant t^{*i} \leqslant t^+, \\ 0 & \text{otherwise.} \end{cases}$$

The collision estimator has two main problem: It requires large number of samples to converge. Also, it introduces bias by blurring.

The track-length estimator Another way of estimate a light beam in the medium is track-length estimator, which draws a short beam with constant intensity value for each sample, this short beam is expressed as:

$$f_{Ti}(t) = \begin{cases} 1 & 0 \leqslant t \leqslant t^{*i}, \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

Then when evaluating the transmittance at distance t^*:

$$\begin{aligned} Tr(t^*) &\approx \sum_{1}^{n} f_{Ti}(t^*)\, p(t^{*i}) \\ &= \sum_{1 \leqslant i \leqslant n,\, t^{*i} > t^*} \sigma_t e^{-\sigma_t t^{*i}} \end{aligned} \tag{5.7}$$

when $n \to \infty$, Equation 5.7 becomes

$$\lim_{n\to\infty} \sum_{1 \leqslant i \leqslant n,\, t^{*i} > t^*} \sigma_t e^{-\sigma_t t^{*i}} = \int_{t^*}^{\infty} \sigma_t e^{-\sigma_t t} \mathrm{d}t \tag{5.8}$$

$$= e^{-\sigma_t t^*} \tag{5.9}$$

Since our photon surfaces requires more than one dimension of transmittance estimation. Here, we illustrate, in a canonical setup, how each combinations of transmittance estimators look like in two-dimensional cases. As shown in top left corner of Figure 5.3,

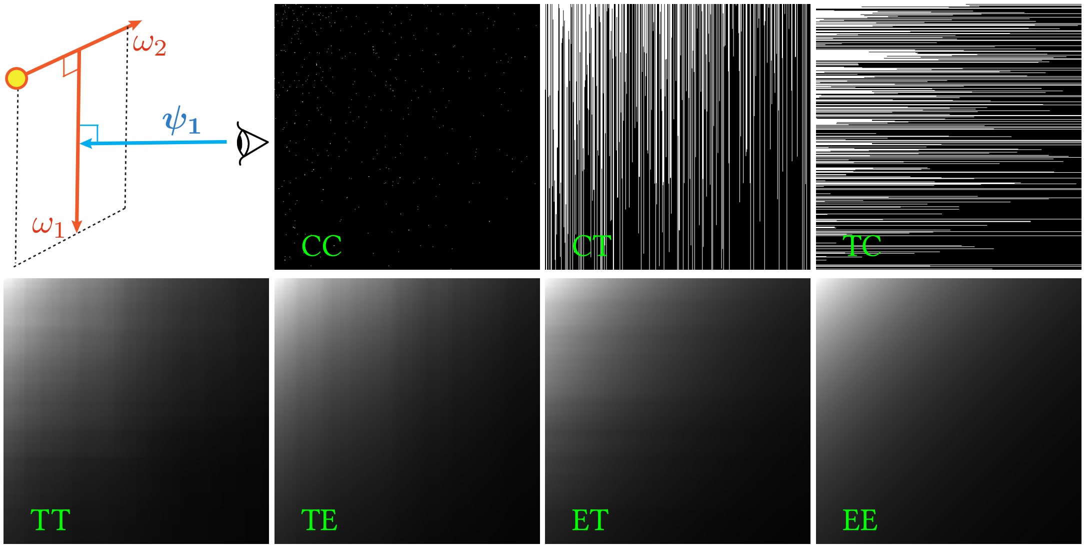

Figure 5.3: Transmittance estimators: The first and second letters in green label represent the transmittance estimators used along direction $\boldsymbol{\omega}_2$ and $\boldsymbol{\omega}_1$ respectively. C: Collision estimator; T: Track-length estimator; E: Expected value of transmittance.

a photon is firstly emitted from light source along direction $\boldsymbol{\omega}_2$ and then it scatters into direction $\boldsymbol{\omega}_1$, which is perpendicular to $\boldsymbol{\omega}_2$; then a camera view the scene from direction $\boldsymbol{\psi}_1$ which is perpendicular to the plane formed by $\boldsymbol{\omega}_2$ and $\boldsymbol{\omega}_1$. Suppose we always use expected transmittance along $\boldsymbol{\psi}_1$, then use different density estimators, including collision estimator (C), track-length estimator (T) and expected transmittance (E), along $\boldsymbol{\omega}_2$ and $\boldsymbol{\omega}_1$. This gives us nine possible combinations. We draw seven of them out in Figure 5.3.

Since the transmittance estimators is in an orthogonal dimension to our derivation of Jacobian for photon surfaces, we can directly use track-length estimators for transmittance evaluation.

5.3 Generalize photon plane to photon surfaces

We have showed how to derive photon plane estimator in our generalized framework section 5.2, now we want to extend this to a family of unbiased estimators by selecting different combination of the analytic variables. Although there are many choices of integration dimensions that can be used as $\overline{\xi_a}$, what we focus on, in this thesis, is a subset of this family which takes two integration dimensions ξ_{a_1}, ξ_{a_2} from the photon subpath and always uses $\xi_{a_3} = s_1$ as the third integration dimension. By doing this, at analytic integration stage, all the other path integral dimensions except $\overline{\xi_a}$ are fixed and the $\mathbf{x}_0$ is only influenced by two variable, ξ_{a_1} and ξ_{a_2}. Then integrating over ξ_{a_1} and ξ_{a_2} sweeps all possible locations of $\mathbf{x}_0$ and produces a two dimensional parametric surface $\mathbf{x}_0(\xi_{a_1}, \xi_{a_2})$, which named "photon surface". Similarly, all the possible positions of point $\mathbf{y}_0$ influenced by ξ_{a_3} form a query beams. Integrating over $\overline{\xi_a}$ is equivalent to finding the intersections between the photon surfaces and the query beam, and each intersection point relates to a full light path sample.

With our constrain on the choice of analytic integration domain, the vector $\mathbf{g}$ with respect to analytic variable can be written generally as

$$\mathbf{g}(\overline{\xi_a}) = \mathbf{x}_0(\xi_{a_1}, \xi_{a_2}) - \mathbf{y}_0(s_1) \tag{5.10}$$

and solving equation $\mathbf{g}(\overline{\xi_a}) = \mathbf{0}$ is identical to finding the intersection between query beams and the photon surface. It is obvious that if the surface have some curvature properties, a photon sphere for example, there will be more than one intersection between a surface and a beam. Recall that in section 5.2, we want to transform the delta function of $\mathbf{g}$ directly to a function of $\overline{\xi_a}$. That requires dividing the function domain into parts before the change-of-variables happens to ensure that each part has the one-to-one property. Then, for a function $\mathbf{g}$ with more than one root, the delta function on it equals to the sum of delta functions on $\overline{\xi_a}$ of each parts:

$$\boldsymbol{\delta}^3(\mathbf{g}(\overline{\xi_a})) = \sum_r \frac{\boldsymbol{\delta}^3(\overline{\xi_a} - \overline{\xi_a^{*r}})}{\left|\frac{\partial \mathbf{g}}{\partial \overline{\xi_a}}(\overline{\xi_a^{*r}})\right|}, \tag{5.11}$$

where $\overline{\xi_a^{*r}}$ is the rth root of equation $\mathbf{g}(\overline{\xi_a}) = 0$. The denominator of Equation 5.11 is referred to as Jacobian term and is represented by $\mathbf{J}^{\mathbf{g}}_{\overline{\xi_a}}$ in this thesis. From Equation 5.10, we can compute the Jacobian as follow:

$$\begin{aligned}
\mathbf{J}^{\mathbf{g}}_{\overline{\xi_a}} &= \left|\frac{\partial \mathbf{g}}{\partial \overline{\xi_a}}(\overline{\xi_a^{*r}})\right| = \left|\det\left[\frac{\partial \mathbf{g}(\overline{\xi_a^{*r}})}{\partial \xi_{a_1}}, \frac{\partial \mathbf{g}(\overline{\xi_a^{*r}})}{\partial \xi_{a_2}}, \frac{\partial \mathbf{g}(\overline{\xi_a^{*r}})}{\partial s_1}\right]\right| \\
&= \left|\det\left[\frac{\partial \mathbf{x}_0(\xi_{a_1}^{*r}, \xi_{a_2}^{*r})}{\partial \xi_{a_1}}, \frac{\partial \mathbf{x}_0(\xi_{a_1}^{*r}, \xi_{a_2}^{*r})}{\partial \xi_{a_2}}, \frac{\partial \mathbf{y}_0(s_1^{*r})}{\partial s_1}\right]\right| \\
&= \left|\left(\frac{\partial \mathbf{x}_0(\xi_{a_1}^{*r}, \xi_{a_2}^{*r})}{\partial \xi_{a_1}} \times \frac{\partial \mathbf{x}_0(\xi_{a_1}^{*r}, \xi_{a_2}^{*r})}{\partial \xi_{a_2}}\right) \cdot \frac{\partial \mathbf{y}_0(s_1^{*r})}{\partial s_1}\right|.
\end{aligned} \tag{5.12}$$

In Equation 5.12 above, $\frac{\partial \mathbf{x}_0(\xi_{a_1}^{*r}, \xi_{a_2}^{*r})}{\partial \xi_{a_1}}$ and $\frac{\partial \mathbf{x}_0(\xi_{a_1}^{*r}, \xi_{a_2}^{*r})}{\partial \xi_{a_2}}$ can be considered as two tangent vector of the parametric surface $\mathbf{x}_0(\xi_{a_1}, \xi_{a_2})$ at point $(\xi_{a_1}^{*r}, \xi_{a_2}^{*r})$, then their cross product is a normal of the surface. This surface normal $\mathbf{n}$ is not normalized because it includes information of

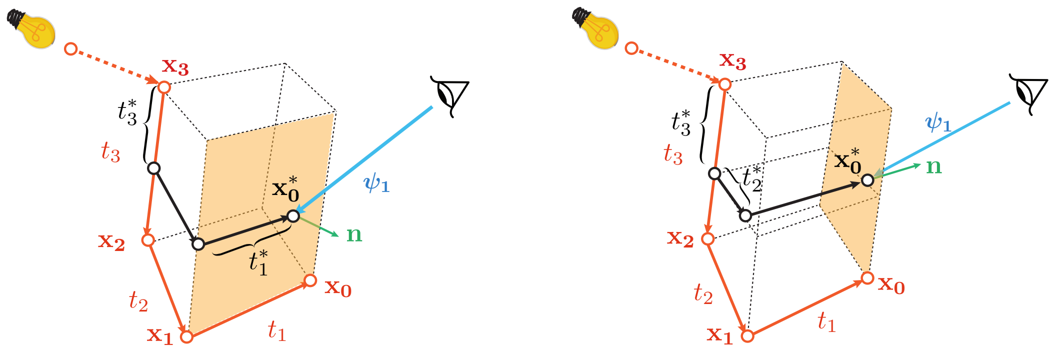

Figure 5.4: Generalized photon planes

the angle between two tangent vectors. Since all the estimators in this paper has a Jacobian term that transform function of $\overline{\xi_a}$ to function of $\mathbf{g}$, and according to Equation 5.12, all the Jacobian of our photon surfaces estimators can be expressed as a dot product between photon surface normal and the query beam's direction:

$$\mathbf{J}^{\mathbf{g}}_{\overline{\xi_a}} = \left|\mathbf{n}(\overline{\xi_a}) \cdot \boldsymbol{\psi}_1\right|$$

Then, by following the same recipe of deriving the 0D photon plane estimator, we procure the general expression of our unbiased density estimators,

$$I_a = \sum_r \frac{f_{\boldsymbol{\omega}}^{1,1} f(\overline{\xi_a})}{\left|\mathbf{n}(\overline{\xi_a^{*_r}}) \cdot \boldsymbol{\psi}_1\right|} \tag{5.13}$$

In the next subsection, we will instantiate different choices of ξ_{a_1} and ξ_{a_2} and introduce corresponding photon surfaces estimators including planes in thelight path with more than 2 bounces in medium, and photon surfaces for light path that has one bounce off the area light source.

5.4 Generalized photon planes

The original photon plane assign t_1 and t_2 to ξ_{a_1} and ξ_{a_2}, now we can extend this to a broader group of photon plane estimators by assigning arbitrary two distances variable t_i, t_j $(l \geqslant i > j \geqslant 1)$ on photon subpath to ξ_{a_1} and ξ_{a_2}. To derive the Jacobian term, we start with the offset vector **g**:

$$\begin{aligned}\mathbf{g}(\overline{\xi_a}) &= \underbrace{\mathbf{x}_l + t_i\boldsymbol{\omega}_i + t_j\boldsymbol{\omega}_j + \sum_{b\neq i,j} t_b\boldsymbol{\omega}_b}_{\mathbf{x}_0(t_i,t_j)} - \underbrace{(\mathbf{y}_1 + s_1\boldsymbol{\psi}_1)}_{\mathbf{y}_0(s_1)} \\ &= \left(\mathbf{x}_l - \mathbf{y}_k + \sum_{b\neq i,j} t_b\boldsymbol{\omega}_b\right) + \mathbf{A}\overline{\xi_a}, \qquad (5.14)\end{aligned}$$

where $A = [\boldsymbol{\omega}_i, \boldsymbol{\omega}_j, \boldsymbol{\psi}_1]$, and all possible positions of (t_i, t_j) sweep out a photon plane, which is referred as $t_i t_j$-plane. Given the **g** function, by following Equation 5.12, we know the Jacobian of the general photon plane estimator is

$$\mathbf{J}^{\mathbf{g}}_{t_i,t_j,s_1} = \left|\left(\frac{\partial\mathbf{x}_0^*}{\partial t_i} \times \frac{\partial\mathbf{x}_0^*}{\partial t_j}\right) \cdot \frac{\mathbf{y}_0^*}{\partial s_1}\right| = \det(A) = \left|(\boldsymbol{\omega}_i \times \boldsymbol{\omega}_j) \cdot \boldsymbol{\psi}_1\right|$$

Then the precomputed integrand for aribitary choice of photon plane estimator is

$$I_a = \frac{f^{1,1}_{\boldsymbol{\omega}} f(t_i^*) f(t_j^*) f(s_1^*)}{\left|(\boldsymbol{\omega}_i \times \boldsymbol{\omega}_j) \cdot \boldsymbol{\psi}_1\right|} \qquad (5.15)$$

5.5 Photon surfaces for area light source

5.5.1 uv-surfaces

Except for picking the dimensions relating to distance variables to compute analytic integral, for the light path starting from area light source, we can also choose the two-dimensional variable $\mathbf{x}_l$ as integration domain ξ_{a_1}, ξ_{a_2}. This two-dimensional variable $\mathbf{x}_l$ relates to the point sampling process on the area light. Notice that we made an assumption that all the area light sources we use are Ramen surfaces. With this assumption, we can say there must be some orthogonal two-dimension parametrization (u, v) that represent all the points in the 2D space of the light source, and this surface could be expressed as $\mathbf{x}_l(u, v)$. By assigning u, v to ξ_{a_1} and ξ_{a_2}, the offset vector becomes

$$\mathbf{g}(\overline{\xi_a}) = \underbrace{\mathbf{x}_l(u, v) + \sum_{i=1}^{k} \boldsymbol{\omega}_i t_i}_{\mathbf{x}_0(u,v)} - \underbrace{(\mathbf{y}_k + \boldsymbol{\psi}_1 s_1)}_{\mathbf{y}_0(s_1)}$$

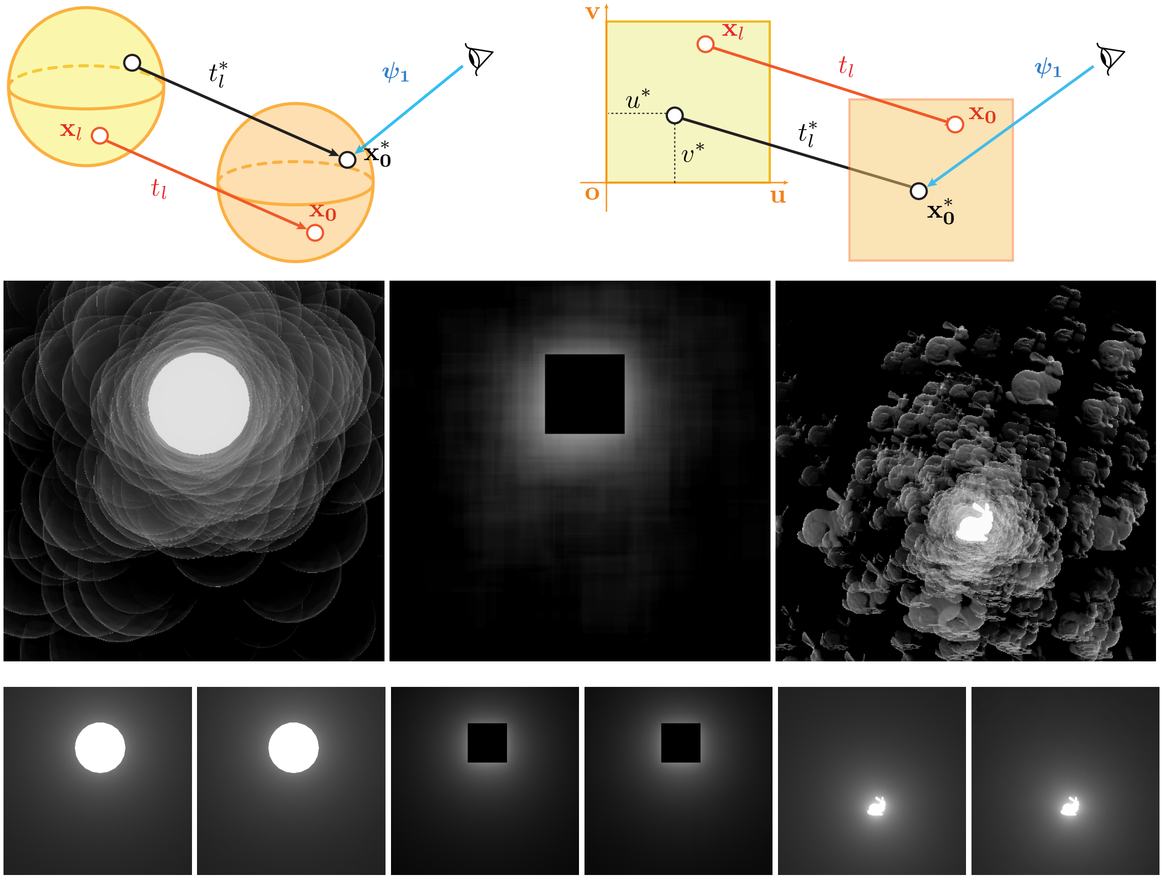

Figure 5.5: uv-surfaces for area lights: Images in first row illustrate how a path is generated by a uv-surface. The second row presents examples of uv-surfaces for a sphere (left), a quad (middle) and a bunny mesh (right) light sources through three rendering results with low sample amounts. The third row shows validations of uv-surfaces by comparing rendering from uv-surfaces estimators (right) with path tracing (left) in an infinite scene with sphere light, quad light and mesh light(bunny).

We can tell from the expression above that the photon surfaces $\mathbf{x}_0(u, v)$ has the same shape as the light surface $\mathbf{x}_l(u, v)$ because $\boldsymbol{\omega}_i t_i$ are linear transformations. We name this group of estimators as uv-surfaces. Following the process of deriving Jacobian that changes variables from u, v, s_1 to $\mathbf{g}$, we yield:

$$\mathbf{J}^{\mathbf{g}}_{u,v,s_1} = \left|\mathbf{n}^* \cdot \boldsymbol{\psi}_1\right|$$

where $\mathbf{n}^*$ is the normalized surface normal at the intersection point between query beams and the uv-surface. Replace the contribution term in Equation 5.13 with the emission function, and put in the Jacobian term, we get the pre-integrated throughput of uv-surfaces:

$$\langle I_a \rangle_{\mathbf{x}_l} = \frac{f^{1,1}_{\boldsymbol{\omega}} L_e(\mathbf{x}^{*r}_l, \boldsymbol{\omega}_l) \cos\theta}{\left|\mathbf{n}^{*r} \cdot \boldsymbol{\psi}_1\right|}$$

where $\mathbf{x}^{*r}_l$ is the corresponding point, on the light source, of $\mathbf{x}^{*r}_0$.

An interesting way to think about this is that here we effectively are performing standard volumetric photon mapping, but instead of it being blurred by an arbitrary 3D kernel, the kernel is the shape of the light source.

5.5.2 ut/vt-planes

It is obvious that the original 0D photon plane estimator requires at least two distance variables alone the photon subpath. Therefore, for the first bounce off surface, we can not draw a photon plane by following the same procedure. However, we can employ some other integration domain, to sweep out parametric photon surfaces. For the first bounce off the light source, we can take the only distance variable and one dimension of the emission direction to sweep out a photon cone as well as take both dimensions of direction sampling to get a photon sphere (Jiao 2018). Also, in previous section, we introduced uv-surfaces which is available for the first bounce off the light source. In addition to uv-surfaces, we can also take u, t_l and v, t_l as the analytic variables. But for most curvature light source, it's not intuitive what photon surfaces will we get by integrating out those dimensions. Nevertheless, it is easier to tell what the u, v axes are for a planar light. As the planar area light source are widely used in rendering, it is worthwhile to derive this ut-/vt- photon plane estimator for it. This section, we will show how to derive ut- and vt- photon planes for planar light sources, which can handle single scattering events for planar light.

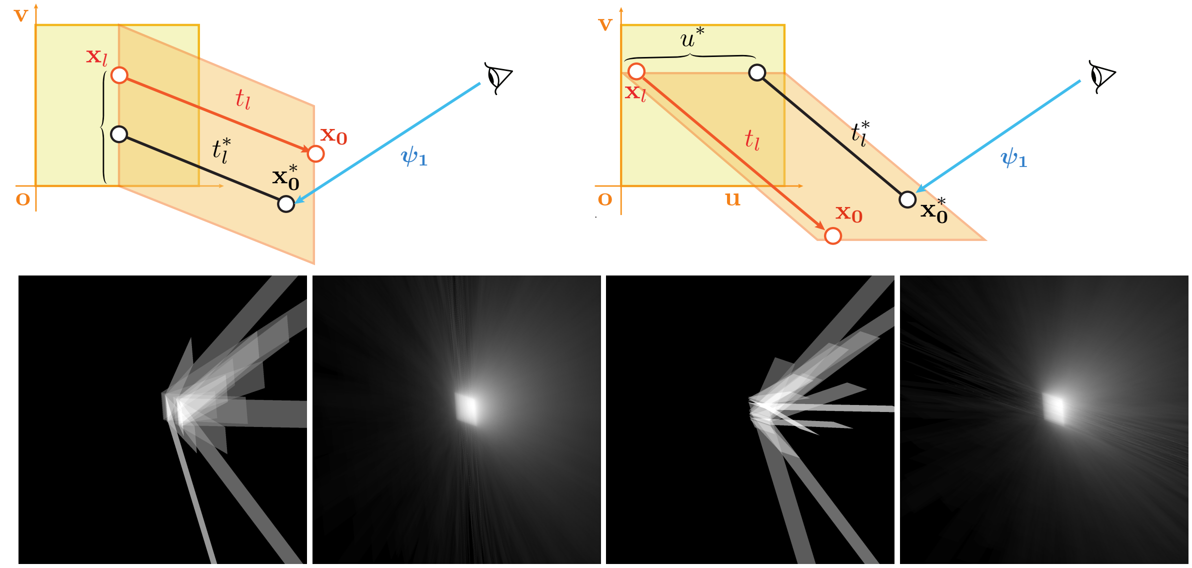

Figure 5.6: ut-planes and vt-planes for planar lights: Images in first row illustrate how paths are generated by ut-planes (left) and vt-planes (right). The second row shows the ut-/vt-planes rendering in a infinite large scene with homogeneous medium and a planar light.

For simplicity, we starts with quad light source. Let two unit vectors $\mathbf{u}, \mathbf{v}$ go along two orthogonal edges of the light, and the origin point $\mathbf{o}$ be a corner. Any point on the same plane as the light source can be expressed by (u, v) where u is the length of $\mathbf{ox}_l$'s projection on $\mathbf{u}$ and v is the length of $\mathbf{ox}_l$'s projection on $\mathbf{v}$. We take u, t_j as analytic integrate variables, where $1 \leqslant j \leqslant l$ then we express $\mathbf{g}$ in the following form:

$$\begin{aligned}
\mathbf{g}(\overline{\xi_a}) &= \underbrace{\mathbf{o} + u\mathbf{u} + v\mathbf{v} + t_j\boldsymbol{\omega}_j + \sum_{i=1, i\neq j}^{l} t_i\boldsymbol{\omega}_i}_{\mathbf{x}_0(u, t_l)} - \underbrace{(\mathbf{y}_1 + s_1\boldsymbol{\psi}_1)}_{\mathbf{y}_0(s_1)} \\
&= \mathbf{o} + v\mathbf{v} + \sum_{i=1, i\neq j}^{l} t_i\boldsymbol{\omega}_i - \mathbf{y}_1 + \mathbf{A}\overline{\xi_a}
\end{aligned}$$

where $\mathbf{A} = [\mathbf{u}, \boldsymbol{\omega}_j, \boldsymbol{\psi}_1]$ and $\overline{\xi_a} = [u, t_j, s_1]$. The 2D parameteric surface is a plane since u, t_j are two distance variables. Then, the corresponding Jacobian term equals to the determinant of matrix $\mathbf{A}$:

$$\mathbf{J}^{\mathbf{g}}_{u, t_j, s_1} = \left|(\mathbf{u} \times \boldsymbol{\omega}_j) \cdot \boldsymbol{\psi}_1\right|$$

Put this Jacobian term into the demoninator of Equation 5.13 and substitude the contribution term with emission at the corespondent point of $\mathbf{x}_0^*$ on light source, we yield the

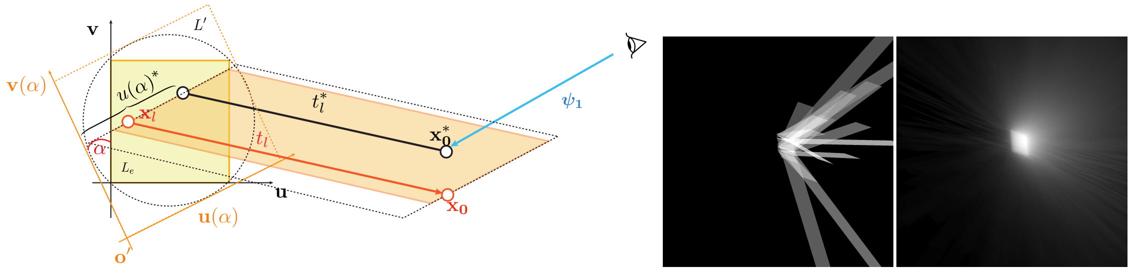

Figure 5.7: Randomly-oriented planes for planar light: left most is an illustration of how the randomly-oriented plane is formed and how a path is generated from it. Two images on the right are simple renderings (low sample count on the left, high sample count on the right) of one randomly-oriented plane estimator in a infinite large scene with participating and a quad light.

pre-integrated contribution:

$$\langle I_a\rangle_{ut_j} = \frac{f_{\boldsymbol{\omega}}^{1,1} L_e((u^*,v),\boldsymbol{\omega}_l)\cos\theta f(t_j)}{\left|(\mathbf{u}\times\boldsymbol{\omega}_j)\cdot\boldsymbol{\psi}_1\right|} \tag{5.16}$$

Similarly, we take v, t_j as the analytic variables and pre-integrate to procure the vt-plane estimator

$$\langle I_a\rangle_{vt_j} = \frac{f_{\boldsymbol{\omega}}^{1,1} L_e((u,v^*),\boldsymbol{\omega}_l)\cos\theta f(t_j)}{\left|(\mathbf{v}\times\boldsymbol{\omega}_j)\cdot\boldsymbol{\psi}_1\right|} \tag{5.17}$$

Then the group of estimator which $j = l$ are single scattering photon planes, and $j = l-1$ are double scattering photon planes.

This ut- and vt- plane estimator can also be extended to any planar light source. Take a disk light source L as an example, we wrap it with a quad L'. The parameterization takes two unit vector along the edge of L' as axes and the corner of L' as the origin point of this coordinates. Then we define a virtual light source on L' such that emission equals to zero when the point (u,v) is outside L:

$$L_e(\mathbf{x}_l(u,v),\boldsymbol{\omega}_l) = \begin{cases} L_e(\mathbf{x}_l,\boldsymbol{\omega}_l) & \text{if } (u,v)\in L, \\ 0 & \text{otherwise} \end{cases}$$

5.5.3 Randomly-oriented ut/vt-planes

With previous redefinition of planar light source, in fact, we can have a continuum set of ut-/vt- planes since there are infinity many orthogonal parameterization of light source through rotation. Here we show the derivation of randomly-oriented ut/vt-plane based on a quad light.

Starting with the local parameterization of light source described in subsection 5.5.2 where **u** and **v** goes along two edges of the light and **o** is the corner connecting these two edges, we rotate the local parameterization clockwise by α°around the geometry center of the quad light and yield new parameterization $(u(\alpha), v(\alpha))$. α is the angle between new axes $\mathbf{u}(\alpha)$ and the old axes' vector **u**. When $\alpha = 0$, $(u(\alpha), v(\alpha))$ refers to the original local parameterization. We redefine a virtual quad light L' to wrap the origin quad light L, and set the origin of the axes to one corner of this virtual quad light. Then we can derive new ut/vt-plane on this rotated parameterization by following the same recipe in subsection 5.5.2. Then the pre-integrated contribution of a randomly-oriented $ut/vt-$ is:

$$\langle I_a \rangle_{u(\alpha)t_j} = \frac{f_{\boldsymbol{\omega}}^{1,1} L_e'((u(\alpha)^*, v(\alpha)), \boldsymbol{\omega}_l) \cos\theta f(t_j)}{\left|(\mathbf{u}(\alpha) \times \boldsymbol{\omega}_j) \cdot \boldsymbol{\psi}_1\right|} \tag{5.18}$$

$$\langle I_a \rangle_{v(\alpha)t_j} = \frac{f_{\boldsymbol{\omega}}^{1,1} L_e((u(\alpha), v(\alpha)^*), \boldsymbol{\omega}_l) \cos\theta f(t_j)}{\left|(\mathbf{v}(\alpha) \times \boldsymbol{\omega}_j) \cdot \boldsymbol{\psi}_1\right|} \tag{5.19}$$

In next section, we'll show how to utilize this continuum set of photon planes in multiple importance sampling and reduce the singularities from photon plane estimators.

Chapter 6

Multiple Importance Sampling

From the Jacobian term in the denominator of Equation 5.13, we know that, when the angle between $\mathbf{n}(\overline{\xi_a^{*r}})$ and $\boldsymbol{\psi}_1$ becomes 90°, the value of the analytic integration evaluated by the photon surface will be infinitely large. In another words, the closer the angle is to 90°, the larger the path contribute is to the pixel intensity. For example, the photon sphere has bright silhouette, as well as photon planes become super bright when viewed from glancing angles. Those effects bring singularities to rendering results. The main purpose of this chapter is providing solutions to alleviate this noisy effects.

Observing that each intersection point between query beam and photon surface produces a standard light path in Figure 4.5, we interpret our photon surface estimators as sampling strategies in standard space. Moreover, as each photon surface estimator has advantages over others in some area, to address the singularity problem, we can use multiple importance sampling to combine the benefits of estimators.

In this section, we'll show how to alleviate the noise of 2+ bounce by applying multiple importance sampling among generalized photon planes, and how we reduce the singu-

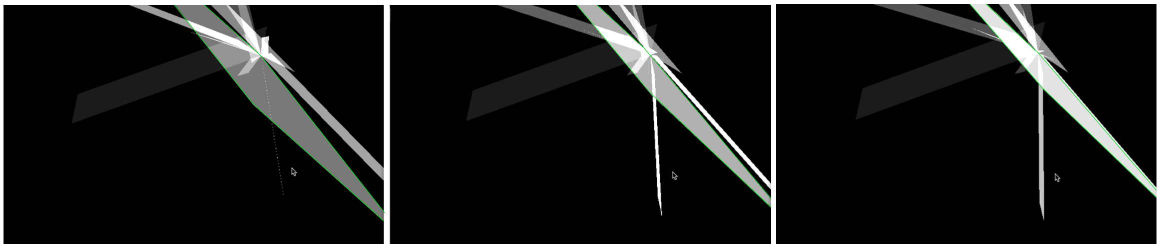

Figure 6.1: An example of plane singularity: Here we show the brightness change of a photon plane (highlight with green lines) when we rotate the viewing angle.

larities of single scattering photon surfaces by combining ut-,vt- and uv planes. We also demonstrate a method to reduce the ut/vt- planes' singularities by multiple importance sampling among a continuum of strategies.

6.1 Combining a Discrete Collection of Strategies

Multiple importance sampling combines discrete collection of strategies by summing up the weighted contribution computed from each strategy. Usually, for a strategy j, the mis weight is defined to be the ratio between its PDF, $p_j(\overline{\mathbf{z}})$, and the sum of the PDFs of all strategies that can produce the same path:

$$w_j(\overline{\mathbf{z}}) = \frac{p_j(\overline{\mathbf{z}})^\beta}{\sum_{i=1}^{m} p_i(\overline{\mathbf{z}})^\beta}$$

where $\beta = 1$ gives balance heuristic and $\beta = 0$ relates to straight average. However, our photon surface estimators is derived by pre-integrating three selected dimensions in the path integral. Thus, the remaining dimensions differs according to different choice of analytic integration domains, which made it hard to directly compute the weight from path PDFs. Instead we use the inverse of path throughput as a proxy of path PDFs.Jendersie (2018):

$$w_j(\overline{\mathbf{z}}) = \frac{I_j(\overline{\mathbf{z}})^{-\beta}}{\sum_{i=1}^{m} I_i(\overline{\mathbf{z}})^{-\beta}} \tag{6.1}$$

where $I_i(\overline{\mathbf{z}})$ is the path throughput in Equation 5.1 that computed by ith strategy.

6.1.1 MIS Thrid-plus Bounces Photon Planes

With a complete light path who has more than three edges on the photon subpath, there are at least t_1t_2-, t_1t_3- and t_2t_3-planes available for MIS. Take strategy t_1t_2-plane for example, replace the analytic integration in Equation 5.1 by Equation 5.15 and put it into

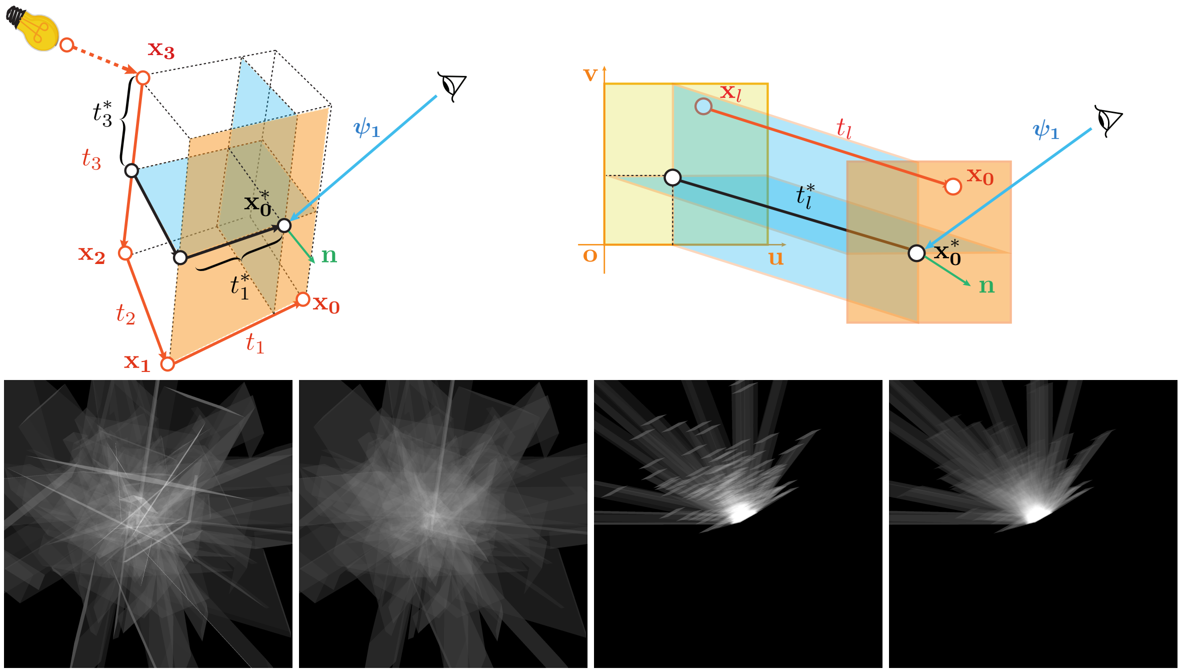

Figure 6.2: MIS discrete startegies: Top left image is an illustration of combining three generalized photon planes ($t_1 t_2$-planes, $t_1 t_3$-planes and $t_2 t_3$-planes). Top right image shows how we combine three single scattering photon surfaces (ut-/vt-planes and uv-planes) for planar light source. Images on the bottom is comparison between straight (left) average and MIS (right).

Equation 6.1, then its weight is:

$$
\begin{aligned}
w_{t_1 t_2}(\bar{\xi}) &= \frac{\frac{p(\bar{\xi}/\{t_1,t_2,s_1\})\mathbf{J}^{\mathbf{g}}_{t_1,t_2,s_1}}{\cancel{f(\bar{\xi})}}}{\frac{p(\bar{\xi}/\{t_1,t_2,s_1\})\mathbf{J}^{\mathbf{g}}_{t_1,t_2,s_1}+p(\bar{\xi}/\{t_2,t_3,s_1\})\mathbf{J}^{\mathbf{g}}_{t_2,t_3,s_1}+p(\bar{\xi}/\{t_1,t_3,s_1\})\mathbf{J}^{\mathbf{g}}_{t_1,t_3,s_1}}{\cancel{f(\bar{\xi})}}} \\
&= \frac{p(\bar{\xi}/\{t_1,t_2,s_1\})\mathbf{J}^{\mathbf{g}}_{t_1,t_2,s_1}}{p(\bar{\xi}/\{t_1,t_2,s_1\})\mathbf{J}^{\mathbf{g}}_{t_1,t_2,s_1}+p(\bar{\xi}/\{t_2,t_3,s_1\})\mathbf{J}^{\mathbf{g}}_{t_2,t_3,s_1}+p(\bar{\xi}/\{t_1,t_3,s_1\})\mathbf{J}^{\mathbf{g}}_{t_1,t_3,s_1}}
\end{aligned}
\tag{6.2}
$$

6.1.2 MIS Single Scattering Photon Planes

According to section 5.1, for a complete light path who only has one edge along photon subpath, we can combine ut-, vt- and uv-plane estimators. Similarly, when MIS these three strategies, the weight is computed as:

$$
\begin{aligned}
w_{uv}(\bar{\xi}) &= \frac{p(\bar{\xi}/\{u,v,s_1\})\mathbf{J}^{\mathbf{g}}_{u,v,s_1}}{p(\bar{\xi}/\{u,t_1,s_1\})\mathbf{J}^{\mathbf{g}}_{u,v,s_1} + p(\bar{\xi}/\{u,t_1,s_1\})\mathbf{J}^{\mathbf{g}}_{u,t_1,s_1} + p(\bar{\xi}/\{v,t_1,s_1\})\mathbf{J}^{\mathbf{g}}_{v,t_1,s_1}} \\
&= \frac{p(t_l)p(\boldsymbol{\omega}_l)\mathbf{J}^{\mathbf{g}}_{u,v,s_1}}{p(t_l)p(\boldsymbol{\omega}_l)\mathbf{J}^{\mathbf{g}}_{u,v,s_1} + p(v|u)p(\boldsymbol{\omega}_l)\mathbf{J}^{\mathbf{g}}_{u,t_1,s_1} + p(u|v)p(\boldsymbol{\omega}_l)\mathbf{J}^{\mathbf{g}}_{v,t_1,s_1}}
\end{aligned}
\tag{6.3}
$$

Although in theory, for the light path with more than one edge on photon subpath, we could MIS ut-,vt-,uv-planes with $t_i t_j$-planes, the increasing need for shooting shadow ray for occlusion checking makes it less efficient. Therefore, we just use ut-,vt-planes for the first and second bounce.

6.1.3 MIS Second Bounce Photon Planes

Initially we only have the 0D-photon plane at the second bounce off light source, with ut-/vt-photons, we are allowed to do MIS among photon planes at the second bounce off area light. For this specific bounce, we choose to multiple improtance sampling among ut_l-, $ut_{l-1}-$ and $t_l t_{l-1}$-planes.

6.2 Combining a Continuum of Strategies

As mentioned in subsection 5.5.3, there are a continuum set of ut-/vt- planes that can be used for multiple importance sampling, we apply this contiuum MIS in the single scattering event off a planar light source. Recall that an $u(\alpha)$t-plane is yield by rotating the coordinates system by $\alpha°$. Thus we have $\mathbf{u}(\alpha) = \sin\alpha\mathbf{u} + \cos\alpha\mathbf{v}$. Through Equation 5.16 and Equation 5.1, we know the score of a path returned by it is

$$
\begin{aligned}
\langle I_a \rangle_{u(\alpha)t_l} &= \frac{f(\overline{\xi_n})}{p(\overline{\xi_n})} I_a(u(\alpha), t_l, s_1) \\
&= \frac{f(\overline{\xi_n}) f^{1,1}_{\boldsymbol{\omega}} f_t(t_l^*) L_e((u(\alpha)^*, v(\alpha)), \boldsymbol{\omega}_l)}{p(\overline{\xi_n})\mathbf{J}^{\mathbf{g}}_{u(\alpha),t_l,t_1}}
\end{aligned}
$$

where

$$
\mathbf{J}^{\mathbf{g}}_{u(\alpha),t_l,t_1} = \left|(sin\alpha\mathbf{u} + \cos\alpha\mathbf{v}) \cdot \boldsymbol{\psi}_1\right| \tag{6.4}
$$

Firstly, we consider leveraging multiple importance sampling among m discrete estimators, corresponding to m randomly-rotated parameterizations of the original light source. Similar to the last section, we use the reciprocal estimator throughput as a proxy for the path PDF. We define the term $I(\overline{\mathbf{z}}, \alpha_i)$ to represent the path throughput returned by a randomly-oriented plane which makes an angle of α with the ut-plane. If we have m strategies, by balance heuristic the pixel intensity goes

$$I = \frac{1}{N}\sum_{j}^{N} \frac{I(\overline{\mathbf{z}}_j, \alpha)^{-1}}{\frac{1}{m}\sum_{i=1}^{m} I(\overline{\mathbf{z}}_i, \alpha_i)^{-1}} I(\overline{\mathbf{z}}_j, \alpha) \tag{6.5}$$

Replacing the path throughput with section 6.2 and cancelling terms give us

$$I = \frac{1}{N}\sum_{j}^{N} \frac{\mathbf{J}^{\mathbf{g}}_{u(\alpha),t_l,s_1}}{\frac{1}{m}\sum_{i=1}^{m} \mathbf{J}^{\mathbf{g}}_{u(\alpha_i),t_l,s_1}} I(\overline{\mathbf{z}}_j, \alpha) \tag{6.6}$$

Performing MIS between the uncountably infinite possible parameterizations of the light source corresponds to taking limit of the expression as the number of strategies m goes toward infinity. Recall the property of the Riemann sum is

$$\lim_{m\to\infty} \frac{b-a}{m}\sum_{i=1}^{m} f(x) = \int_a^b f(x)\mathrm{d}x \tag{6.7}$$

$$\lim_{m\to\infty} \frac{1}{m}\sum_{i=1}^{m} f(x) = \frac{1}{b-a}\int_a^b f(x)\mathrm{d}x. \tag{6.8}$$

We can then compute the denominator of Equation 6.6 by

$$\lim_{m\to\infty} \frac{1}{m}\sum_{i=1}^{m} \mathbf{J}^{\mathbf{g}}_{u(\alpha_i),t_l,s_1} = \frac{1}{\pi}\int_0^{\pi} \mathbf{J}^{\mathbf{g}}_{u(\alpha),t_l,s_1}\mathrm{d}\alpha \tag{6.9}$$

Inserting Equation 6.4 into Equation 6.9 we obtain

$$\begin{aligned}
&\frac{1}{\pi}\int_0^{\pi} \left|((\sin\alpha\mathbf{u} + \cos\alpha\mathbf{v}) \times \boldsymbol{\omega}_1) \cdot \boldsymbol{\psi}_1)\right|\mathrm{d}\alpha \\
&= \frac{1}{\pi}\int_0^{\pi} |k_1 \sin\alpha + k_2 \cos\alpha|\mathrm{d}\alpha \\
&= \frac{2}{\pi}\sqrt{k_1^2 + k_2^2}
\end{aligned} \tag{6.10}$$

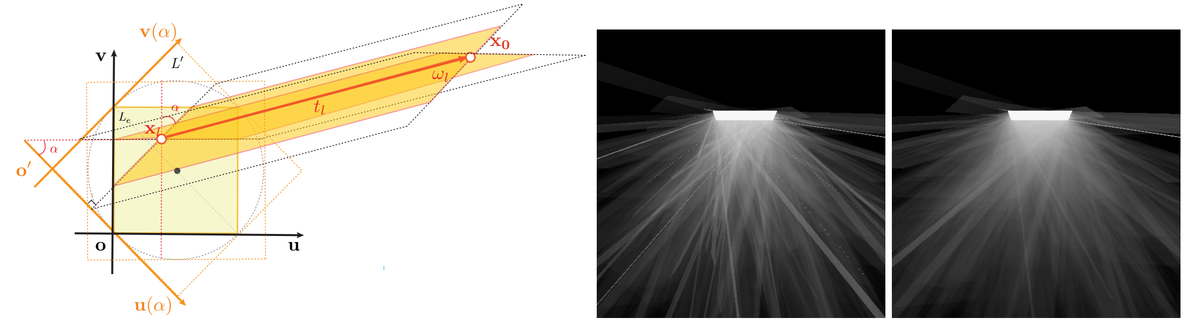

Figure 6.3: MIS a continuum of startegies

where $k_1 = (\mathbf{u} \times \boldsymbol{\omega}_l) \cdot \boldsymbol{\psi}_1$, $k_2 = (\mathbf{v} \times \boldsymbol{\omega}_l) \cdot \boldsymbol{\psi}_1$. This gives us the continuously MIS'd estimator

$$\langle I_a \rangle_C \approx \frac{1}{N} \sum_j^N \frac{\mathbf{J}^{\mathbf{g}}_{u(\alpha),t_l,s_1}}{\frac{2}{\pi}\sqrt{k_1^2 + k_2^2}} I(\overline{\mathbf{z}}_j, \alpha) \tag{6.11}$$

Considering this MIS'd estimator as a whole, the demoninator can be regard as its new Jacobian $\frac{2}{\pi}\sqrt{((\mathbf{u} \times \boldsymbol{\omega}_l) \cdot \boldsymbol{\psi}_1)^2 + ((\mathbf{v} \times \boldsymbol{\omega}_l) \cdot \boldsymbol{\psi}_1)^2}$. This Jacobian only goes to zero when the query beam is parallel to the emission direction while an individual plane estimator goes to zero at any glancing angle. To alleviate the only singularity it have, we can further improved it by MIS with uv-planes.

Chapter 7

Implementation and Results

7.1 Implementation

In order to validate our photon surface estimators, we first implemented it in a simplified real-time renderer with orthogonal camera. Then, for the efficiency comparison with previous unbiased Monte-Carlo estimator, such as path tracing and photon plane Bitterli and Jarosz (2017), we implemented photon surfaces in an open-source renderer Tungsten Bitterli (2018). In this section, we will briefly describe these two implementation.

7.1.1 Real-time implementation

Aim at directly viewing photon surface estimators and validating them by comparing converged images, the real-time implementation makes an assumption that the scene is consist of a light source in infinite large space that filled with homogeneous medium. This assumption makes it possible for us to ignore the occlusion and surfaces first while only focusing on medium. The rendered images are returned by an orthogonal camera, and the camera subpath is limited to only one edge. We added the flexibility of configuring the properties of the participating media, setting the path number, switching emission functions, light shapes and setting up MIS options. This real-time implementation supports $t_1 t_2$-, $t_1 t_3$- and $t_2 t_3$-planes for multiple scattering; can draw $\mathrm{u}t_l$-, $\mathrm{v}t_l$-, uv-, randomly orientated $\mathrm{u}t_l$- planes and uv-surfaces for single scattering (first bounce off light source). For multiple importance sampling at first bounce off the light source, this real-time renderer could combine $\mathrm{u}t_1$-, $\mathrm{v}t_1$- and uv- planes, MIS between randomly orientated $\mathrm{u}t_l$-

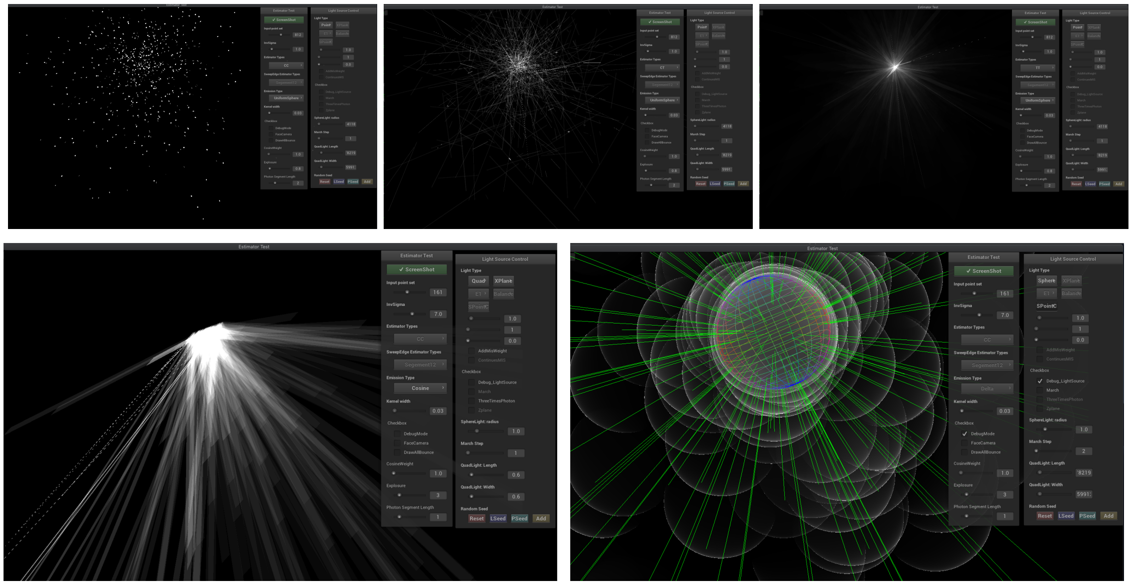

Figure 7.1: Screen shots of real-time implementations.

planes and uv- planes. Moreover, this real-time implementation supports MIS in higher bounces(three-plus bounces) between t_1t_2-, t_1t_3- and t_2t_3-planes.

The photon subpaths are traced on CPUs, expressed as chains of vertices and sent to GPUs. For the sake of efficiency, the program only generates the photon surfaces for selected bounces. Most of the photon surfaces are generated by geometry shaders while some of them are connected in vertex shaders. Then, the intensity values returned by photon subpaths are finalized in fragment shaders.

7.1.2 Tungsten implementation

After having our photon surface estimators validated in simple real-time setups, we had them implemented in a full functional renderer Tungsten, which supports other unbiased Monte-Carlo estimators as well as the previous photon plane estimator. In Tungsten renderer, the surface shading and occlusion test are enabled, which allows us to render actual scenes with our estimators.

Generating photon surfaces Once a photon path is traced, the integrator iterates over all bounces and generates one kind of “photon surface” that specified in the configuration file. If the selected estimator is one of the MIS’d ones, at each bounce, we will uniformly choose one photon surface among the alternative strategies to draw out. This corresponds to the one-sample strategy of MIS. We store those primitives and their bounding box in a uniform grid or a BVH structure, depending on the integrator setting in the input file. For the uv-surfaces, since they share exactly the same shape with the light source, we only store the bounding box and a transform matrix for each photon uv-surface.

Evaluating path In rendering pass, we evaluate each path at the intersections between the query beam and all the photon surfaces. For the purpose of MIS, at each hitpoint, we take the throughput returned by the photon surface that it hits and multiply it by the weight computed from the path throughputs of all estimators. The ways of computing the weights are given in **??**. However, in order to make the implementation more time-efficient, in practice, we only compute a subset of path throughput influenced by different Jacobians.

Handling occlusion When shooting camera rays, some photon surface estimators require us to trace additional shadow rays for visibility test. The reason for this is we integrated three dimensions out, and that makes some photon surface estimators not produce exactly the same path as the one which generated the photon surface. The number of additionally required shadow rays depends on the farthest analytic variable counting from the $\mathbf{x}_0$. Because the photon path generated by a photon surface and the path producing such photon surface only share the same subset of path edges from $\mathbf{x}_l$ to the farthest analytic variable the estimator takes. For example, the uv-surfaces take the position of $\mathbf{x}_l$ on area light as variable, therefore, wherever the hitpoint is, we need to trace a shadow ray all the way back to the light source. As shooting shadow rays are time-consuming, we only choose to include, in implementation, the photon surface estimators whose analytic integration dimensions are as close to $\mathbf{x}_0$ as possible.

In order to make a fair comparison with previous 0D photon plane, additional care should be taken when tracing the photon subpath. Evaluating the path by MIS'd $t_1 t_2$-, $t_1 t_3$- and $t_2 t_3$-planes requires at least three scattering events in the medium, while by the 0D-Photon planes only needs two. If we trace 100 photon subpath in a scene, the number of paths with at least two scattering events is larger than the path with at least three scattering events. To make the sample numbers equal, every time the photon subpath hit a surface, in addition to reflecting and transmitting by following BSDFs, we also force it to pretend not hitting any surface and continue to draw two segments in the medium. The occlusion test allows us to do so without introducing extra radiance, and the path probability is truncated to the hitting point by the surfaces.

7.2 Results

To evaluate our methods, we conducted equal time comparisons between our photon surface estimators and previous methods on a variety of scenes from (Bitterli 2016). All the images in equal time comparison were rendered with the same renderer and hardware (Core i-5 6500, 4 cores, 3.2Ghz). For one particular scene, each estimator went through 100 five-minutes render passes with different seeds, creating 100 rendered images, then the variance of the 100 images is used as the quantitative score of the performance. Since both our photon surfaces estimators and the 0D Photon Plane estimators, as well as path tracing, are unbiased methods, the variances of the 100 images are actually the mean square errors(MSE). However, as the photon beams is a bias method, its variance is just an underestimated version of the error. In order to show the improvement from photon surface estimators, here we mainly focused on rendering results of the first scattering event off the light and the third scattering off the light source.

Figure 7.2 shows the performance of the 3-Planes estimator which combines $t_2 t_1$-, $t_3 t_1$- and $t_3 t_2$- together. This group of images compare the MIS'd 3-Planes estimators (middle column), the straight average of 3-Plane estimators (leftmost column) and the original 0D Plane from Bitterli and Jarosz (2017) (rightmost column). This group of images only rendered out third bounce of the light source. It can be observed that the MIS'd 3-Plane estimator significantly reduced the singularities of planes.

Figure 7.3 compares the MIS'd ut-,uv- and uv- estimators (middle column), the straight average of ut-,uv- and uv estimators (leftmost column) and the photon beams Jarosz et al. (2011a) at the first bounce of light source. This group of comparison contains six scenes ranging from outdoor gas station (Row 4) to indoor living room (Row 6) with light oriented in different directions. This group shows how the rendering problem of single scattering event is solved by unbiased estimators (photon surfaces) with a higher convergence rate than the biased method (photon beams). Scenes in Row 2 and Row 3 have window shades just behind the light sources, which might influence the performance of the plane. However, the results turn out that even with dense occluders on the light source, the photon plane estimators still perform significantly better than beams. Row 4 is an outdoor scene and the light source is coming from the top, this row present the advantage of the MIS on eliminating the singularities of planes.

Figure 7.4 compared the MIS'd photon planes (middle column) with path tracing (leftmost column) in simulating single scattering in the media. We noticed that, although the convergence rate of photon planes are higher than path tracing, they visually relate to different kind of noises. By zooming the image (rightmost column), we observed that the path tracing methods have more high-frequency noises while the MIS'd photon planes estimator has more low-frequency noises. That is reasonable because a photon plane may cover many pixels, and this allows more pixels to share the same random number, therefore its rendered image is visually smoother when being viewed closely. This leads to a discussion on which one is better: high-frequency noises or low-frequency noises, and how we can quantitatively evaluate them and make a judgment.

Figure 7.5 shows the advantage of using the MIS among a continuum of strategies (randomly-oriented planes) over just taking averting these strategies. The variance shows that the rate of convergence when applying MIS is about the twice to three times of the strait averaging. To further confirm how much these randomly oriented planes improve the efficiency, there should also be a comparison between MISing among randomly-oriented planes and uv-planes with MISing among just ut-/vt- and uv- planes. However, there is a debate on how much uv-plane should we include together with randomly-oriented planes, ten percent, one third or fifty percent? Some further tests remain in this.

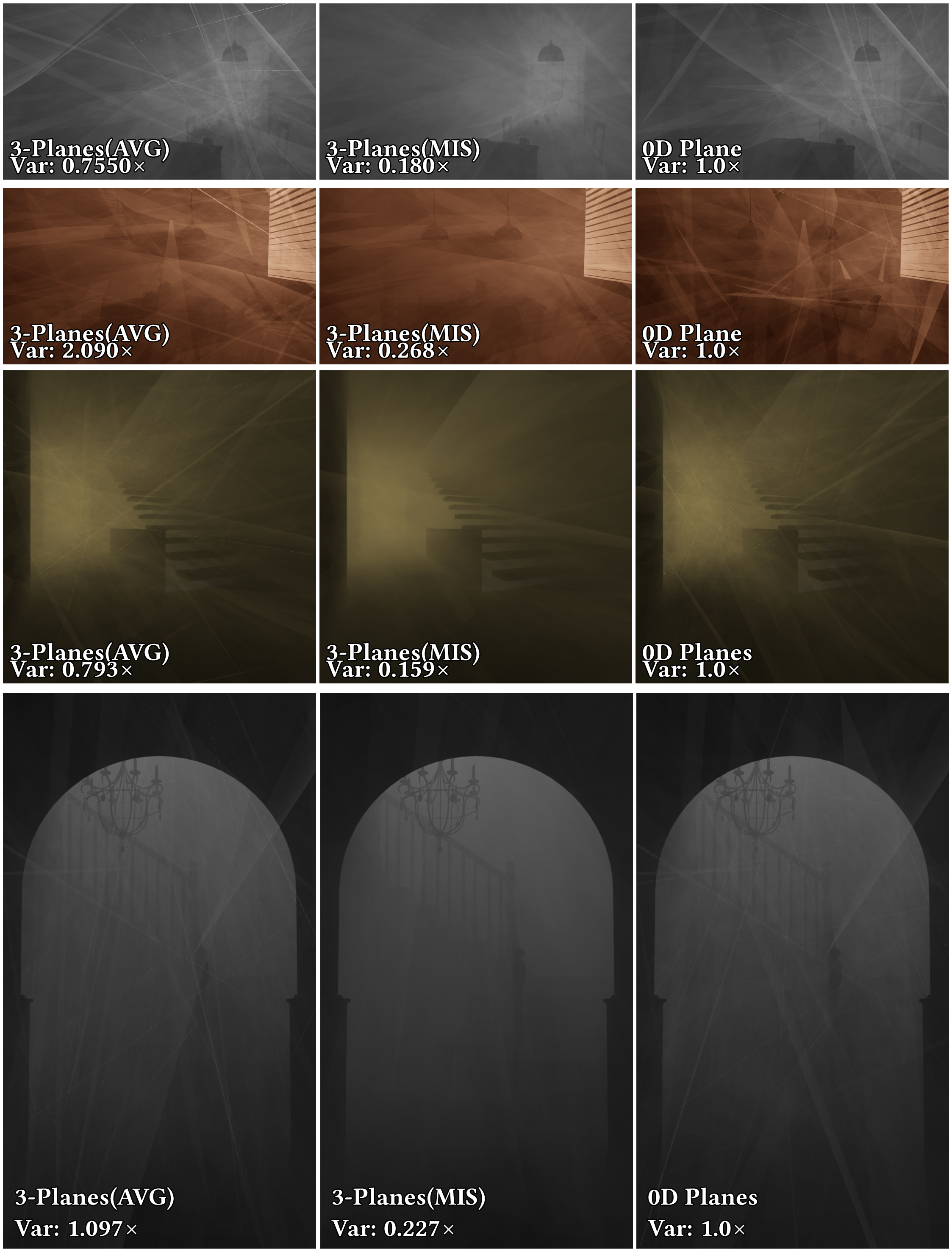

Figure 7.2: Straight average v.s. MIS 3-Planes v.s. 0D Photon Plane: We compare our methods, including straight average of 3-Planes (right column) and MIS of 3-Planes (middle column), with the original 0D photon plane (left column) at the third bounce off light source.

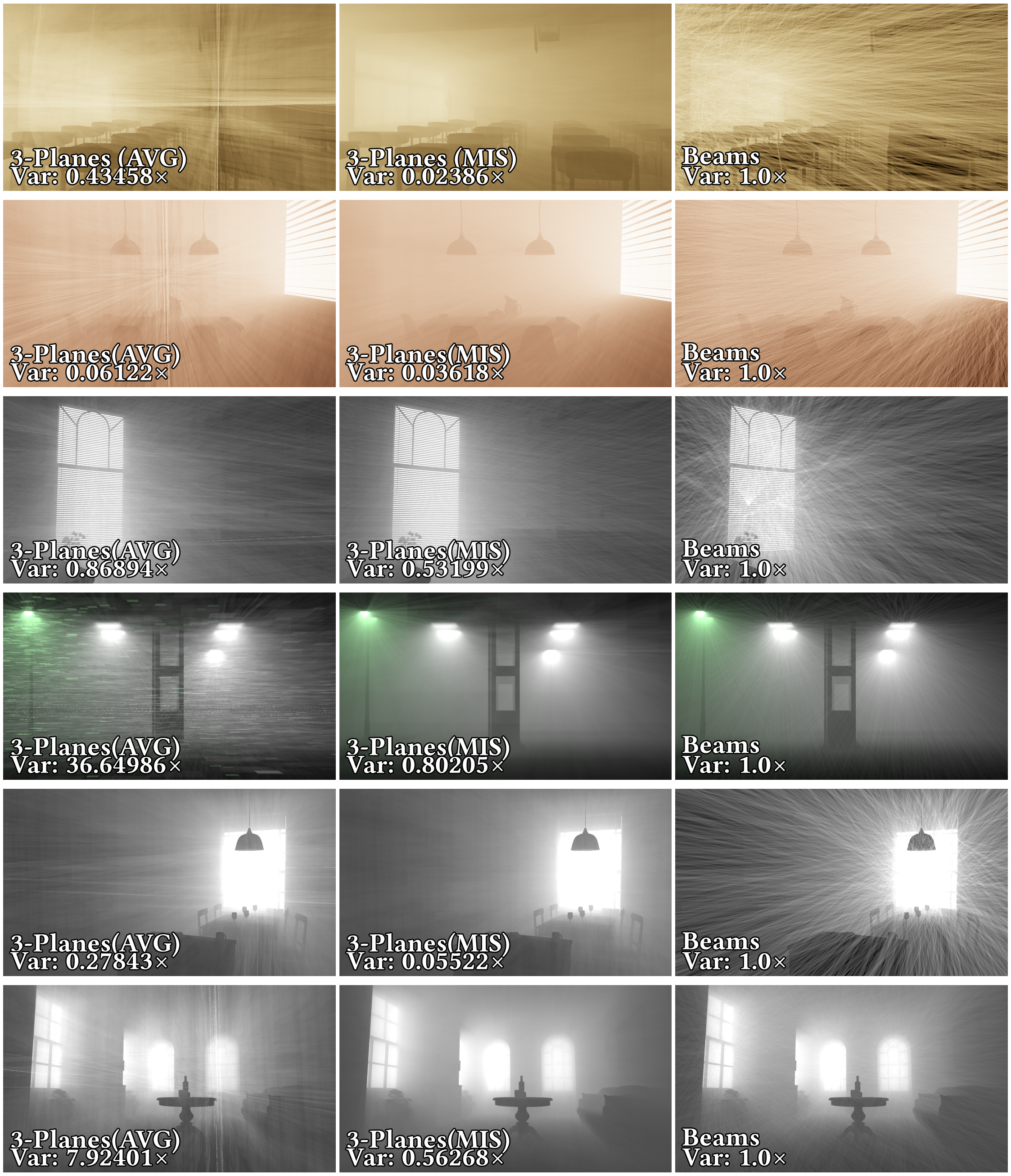

Figure 7.3: Single scattering comparison: straight-averaged ut-/uv-vt-planes v.s. MIS'd ut-/uv-vt-planes v.s. photon beams: We compare our methods for single scattering event off light source, including straight-averaged ut-/uv-vt-planes (right column) and MIS'd ut-/uv-vt-planes (middle column), with photon beams (left column).

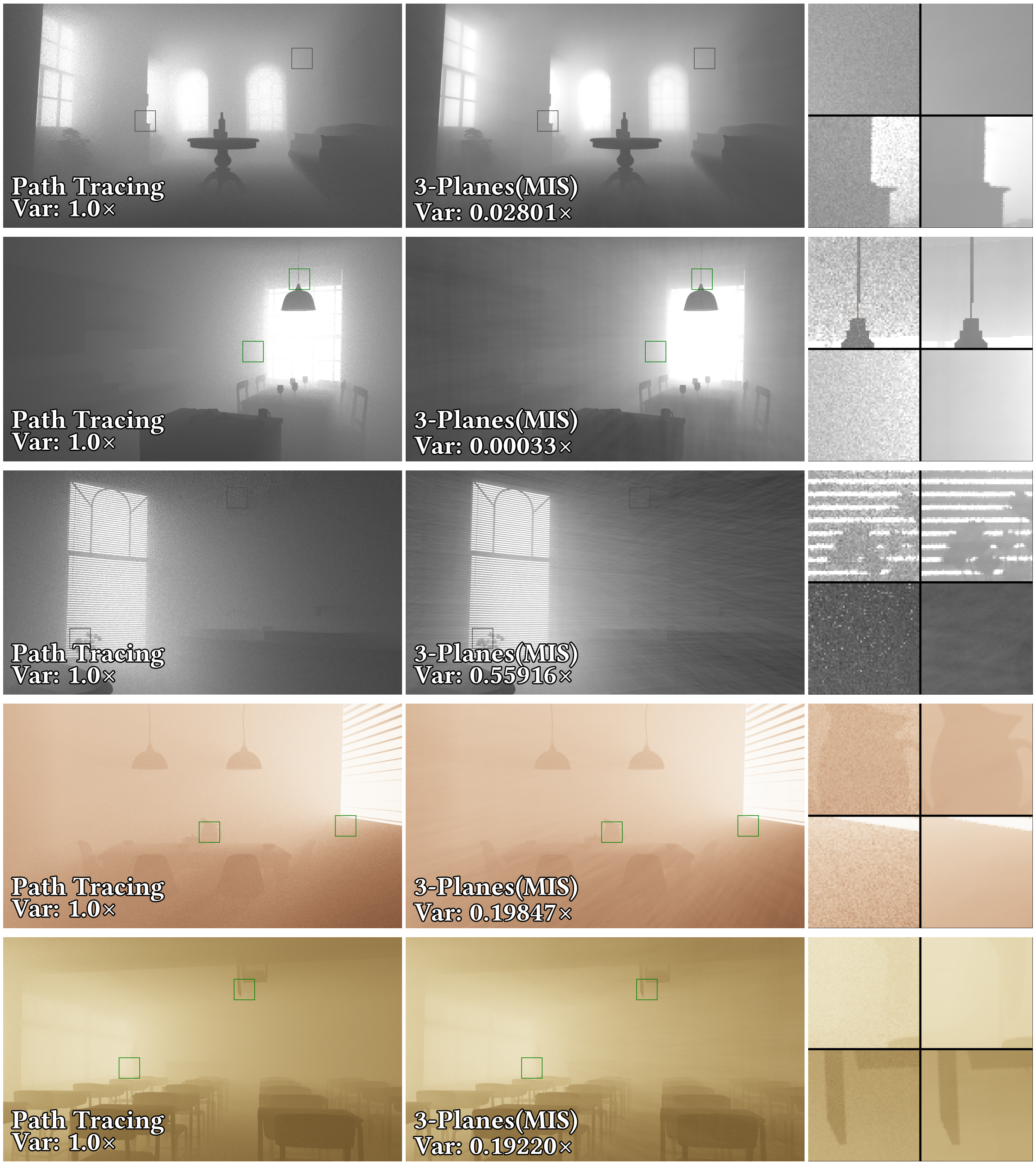

Figure 7.4: Single scattering comparison: MIS'd ut-/uv-/vt-planes v.s. path tracing: We compare our method, MIS'd ut-/uv-/vt-planes (middle column) with path tracing (right column) as well as showing zoomed in comparisons between these two methods (left most two columns).

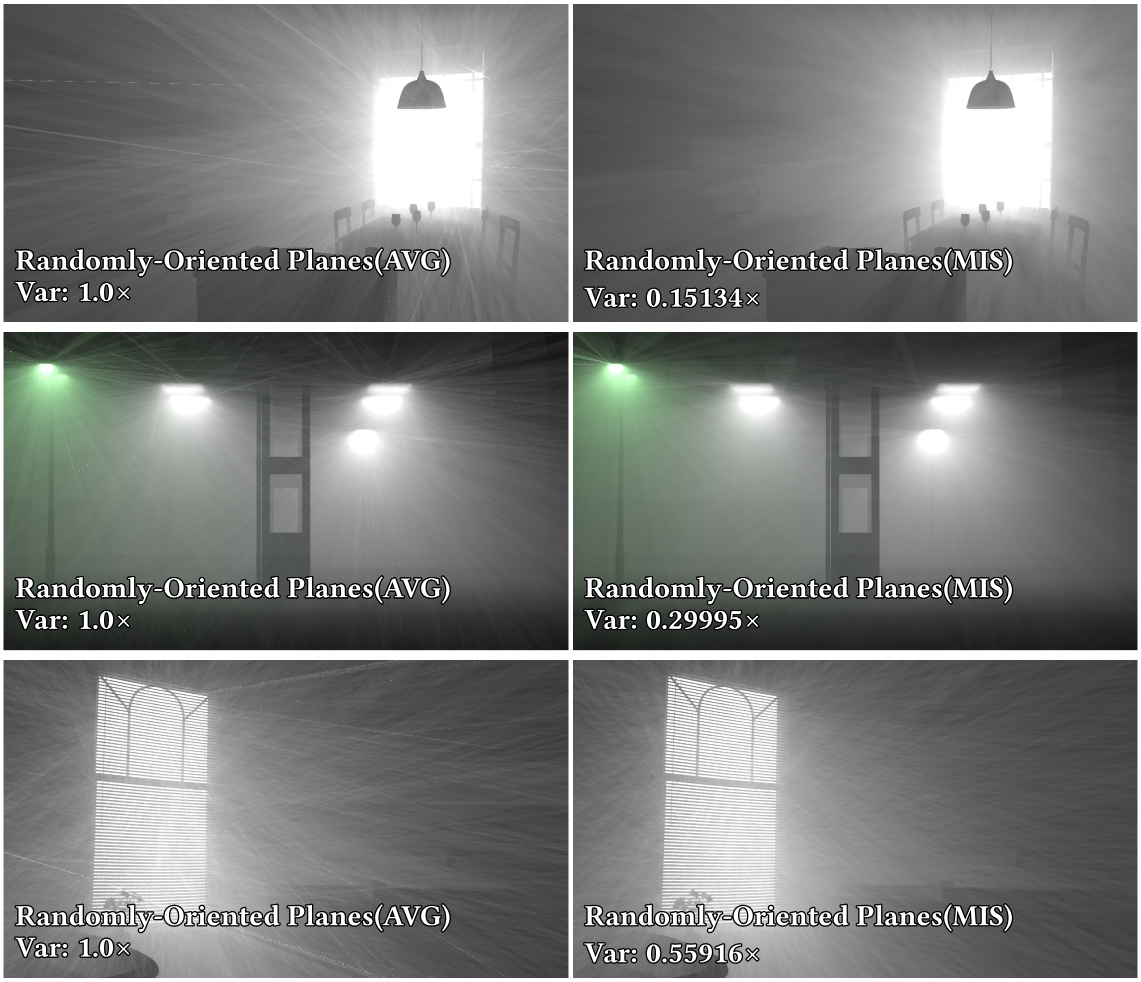
Randomly-Oriented Planes(AVG)
Var: 1.0×
Randomly-Oriented Planes(MIS)
Var: 0.15134×
Randomly-Oriented Planes(AVG)
Var: 1.0×
Randomly-Oriented Planes(MIS)
Var: 0.29995×
Randomly-Oriented Planes(AVG)
Var: 1.0×
Randomly-Oriented Planes(MIS)
Var: 0.55916×

Figure 7.5: We compare among our straight-averaged randomly-oriented photon plane (right) and MIS'd randomly-oriented photon plane (left).

Chapter 8

Conclusion

8.1 Summary

In this paper, we build a new theoretical framework to derive unbiased photon density estimators for the participating media rendering. The main part of our theory utilizes a delta function as blurring kernels to couple a camera subpath and a photon subpath. We then reparameterize the delta kernel with three arbitrary variables and analytically integrate parts of the path integral over the variables. This analytical integration sweeps out the delta kernel and produces photon surfaces. By choosing different sampling dimensions as the analytic integration variables, we produce a group of unbiased density estimators. We re-derive the original photon plane Bitterli and Jarosz (2017) in our framework and got exactly the same equation. Then, we give a more general recipe of deriving photon surface estimators, such that in addition to distance variables, the directional variables and vertex sampling on light could also be included in the pre-integration domains and used to generate photon surfaces. By following this recipe, we extend previous 0D photon planes to a broader group of photon planes. Instead of only using the last two propagation distances as the edges of the photon plane, our $t_i t_j$-plane can use any two propagation distances as the edges of photon planes, therefore providing a family of estimators available for generating exactly the same path. By including point sampling to analytic integration domain, we present single scattering photon surfaces, which give the renderer ability to render light transport that was not supported by previous unbiased density estimators. Furthermore, as each one of the density estimators performs better than others in some area, we considered each unbiased density estimator as path sampling routine and used MIS to combine their advantages and smooth out their weakness. Finally, we achieve noticeable improvement in efficiency of volume rendering.

8.2 Discussion & Future work

Heterogeneous medium Recall that we made the assumption that all the media we work with are homogeneous media. Although our theory could also be revised to fit in heterogeneous conditions, there will be extra variances when applying photon surfaces in heterogeneous media. Since our photon surfaces are 2D samples, paths samples generated from the same photon surfaces estimator may share same direction or distance samples and that does not fit with the idea of using the importance sample to simulate heterogeneous medium. For example, path samples generated from a $t_1 t_2$ photon plane nearly share all the random sample on photon subpath except for t_1 and t_2.

Surface-Medium transport As we used single scattering photon planes to solve light transport problem for the first bounce off light sources, problems of evaluating the first bounce off surfaces by photon density estimator remained unsolved. One potential solution to this is using one or two dimensions in BSDFs samples as pre-integration dimensions and drawing photon cones and spheres at the first bounce off the surface. However, it is obvious that this becomes tricky for specular surfaces, and other discrete BSDFs appearance models.

Another problem in surface-medium transport relates to MIS: for the second bounce off surfaces, we can only generate 0D photon planes, with only one estimator, MIS makes no sense here. It requires more photon surface estimators to apply MIS so that the singularity from the 0D photon plane would not dominate the noise.

One alternative way of applying MIS at second bounce off the surface is using photon surfaces generated from pre-integrating directional sampling domains (photon cones, photon spheres photon cylinders).

Another possible way we could solve low orders (first and second) scattering off surfaces by utilizing ut-/vt-,uv-planes estimators at those bounces. Nonetheless, these estimators become costly when they bounce farther from light source because of their demand for tracing shadow rays.

Point light We derive single scattering photon surfaces for the area lights on which there is an extra sampling dimension. For the point lights without this extra dimension, we need to seek for another sampling dimension when a photon surface is needed for a single scattering event. Our paper focuses on the photon surfaces generated from choosing distance sampling variables and area sampling variables as analytic integration domain, so we didn't provide any single scattering photon surfaces for point light source. However, if we expand our focus on distance and area variables to consider other types of random variables along the path sample, one intuitive choice of the extra dimension of a point light is in the two-dimensional variables of emission direction variable. Then, sweeping out combinations of direction and distance dimensions will generate photon spheres and cones for the first bounce of point light source.

Single scattering Observed from the performance of single scattering photon planes in different scenes, they tend to be the most efficient when the area light is relatively large compared to the camera view. When the area light is small, smaller than a pixel, for example, the performance of single scattering photon planes will perform just as good as previous methods such as path tracing and photon beams. However, one alternative way is considering those kind of small lights as point lights and combine the single scattering photon planes with single scattering spheres and cones (Jiao 2018).

Camera plane On one hand, we have discussed that the size of the area light source will influence the performance of a single scattering photon plane. The reason for this is that large light corresponds to larger photon planes which cover more pixel in camera sensor. On the other hand, earlier in this paper, we limited the derivation of photon surface estimators by forcing two pre-integration domains on the photon subpath and by directly assigning another analytic integration variable to be the last distance on the sensor subpath. If we relax this constraint, there will be more possible query and photon sample combinations, such as using a query surface and a photon beam instead of a query beam and photon surface. For example, if we take a point sample on the sensor as the analytic integration dimensions, similar to uv photon plane, we can get "uv- camera plane". One possible difference between query surface and photon surface is drawing photon surface on camera subpath expect to result in high-frequency noise rather than low-frequency noise because the query plane is not shared by pixels.

Combine with previous methods Since every photon surfaces estimator we derive is unbiased and the path created by camera beams intersecting with photon surfaces are expressed in standard path space, we can think our photon surfaces estimators as path sampling techniques. As mentioned previously, our single scattering photon surfaces have low-frequency noise while path tracing has high-frequency noise. We could think of compensating the noise problem by utilizing MIS among our methods and path tracing, or other previous unbiased methods, to further improve the volume rendering.

Subsurface scattering Another assumption we made is the camera being in the medium, though, in practice, it is not always the case. If the camera is viewing the medium outside the boundary surface, the main interest will be the appearance of a volume boundary, and then this will turn out to be a subsurface scattering problem.

Bibliography

James R. Arvo. Applications of irradiance tensors to the simulation of non-lambertian phenomena. In *Proc. SIGGRAPH*, August 1995a.

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

Rasmus Barringer, Carl Johan Gribel, and Tomas Akenine-Möller. High-quality curve rendering using line sampled visibility. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 31 (6), November 2012. ISSN 0730-0301.

Laurent Belcour, Guofu Xie, Christophe Hery, Mark Meyer, Wojciech Jarosz, and Derek Nowrouzezahrai. Integrating clipped spherical harmonics expansions. *ACM Trans. Graph.*, 37(2), mar 2018. doi: 10.1145/3015459.

Niels Billen and Philip Dutré. Line sampling for direct illumination. *Computer Graphics Forum (Proc. EGSR)*, 35(4), June 2016.

Benedikt Bitterli. Rendering resources, 2016. https://benedikt-bitterli.me/resources/.

Benedikt Bitterli. Tungsten renderer, 2018. https://github.com/tunabrain/tungsten.

Benedikt Bitterli and Wojciech Jarosz. Beyond points and beams: Higher-dimensional photon samples for volumetric light transport. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 36(4), July 2017. doi: 10.1145/3072959.3073698.

Min Chen and James Arvo. Simulating non-Lambertian phenomena involving linearly-varying luminaires. In *Rendering Techniques (Proc. EGWR)*, June 2001.

Min Chen and James R. Arvo. A closed-form solution for the irradiance due to linearly-varying luminaires. In *Rendering Techniques (Proc. EGWR)*, June 2000.

Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '84, pages 137–145, New York, NY, USA, 1984. ACM. ISBN 0-89791-138-5. doi: 10.1145/800031.808590. URL http://doi.acm.org/10.1145/800031.808590.

W. L Wolfe F O Bartell, E. L. Dereniak. The theory and measurement of bidirectional reflectance distribution function (brdf) and bidirectional transmittance distribution function (btdf), 1981. URL https://doi.org/10.1117/12.959611.

Iliyan Georgiev, Jaroslav Křivànek, Tomas Davidovic, and Philipp Slusallek. Light transport simulation with vertex connection and merging. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 31(5), 2012.

Carl Johan Gribel, Michael Doggett, and Tomas Akenine-Möller. Analytical motion blur rasterization with compression. In *Proceedings of HPG*, 2010.

Carl Johan Gribel, Rasmus Barringer, and Tomas Akenine-Möller. High-quality spatio-temporal rendering using semi-analytical visibility. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 30(4), August 2011.

Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. Progressive photon mapping. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 27(5), 2008.

Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. A path space extension for robust light transport simulation. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 31(5), 2012.

Toshiya Hachisuka, Iliyan Georgiev, Wojciech Jarosz, Jaroslav KÅŹivÃąnek, and Derek Nowrouzezahrai. Extended path integral formulation for volumetric transport. In *Proceedings of EGSR (Experimental Ideas & Implementations)*. The Eurographics Association, jun 2017. doi: 10.2312/sre.20171195.

Vlastimil Havran, Jiri Bittner, Robert Herzog, and Hans-Peter Seidel. Ray maps for global illumination. In *Rendering Techniques (Proc. EGSR)*, June 2005.

Eric Heitz, Stephen Hill, and Morgan McGuire. Combining analytic direct illumination and stochastic shadows. pages 2:1–2:11. ACM, 2018. ISBN 978-1-4503-5705-0. doi: 10.1145/3190834.3190852.

L. G. Henyey and J. L. Greenstein. Diffuse radiation in the galaxy. *Astrophysical Journal*, 93:70–83, jan 1941. doi: 10.1086/144246.

Wenzel Jakob. *Light Transport on Path-Space Manifolds*. PhD thesis, Cornell University, August 2013.

Wojciech Jarosz, Matthias Zwicker, and Henrik Wann Jensen. The beam radiance estimate for volumetric photon mapping. *Computer Graphics Forum (Proc. Eurographics)*, 27(2), April 2008a.

Wojciech Jarosz, Matthias Zwicker, and Henrik Wann Jensen. The beam radiance estimate for volumetric photon mapping. *Computer Graphics Forum (Proceedings of Eurographics)*, 27(2):557âĂŞ566, apr 2008b. doi: 10.1111/j.1467-8659.2008.01153.x.

Wojciech Jarosz, Derek Nowrouzezahrai, Iman Sadeghi, and Henrik Wann Jensen. A comprehensive theory of volumetric radiance estimation using photon points and beams. *ACM Trans. Graph.*, 30(1), February 2011a. ISSN 0730-0301.

Wojciech Jarosz, Derek Nowrouzezahrai, Iman Sadeghi, and Henrik Wann Jensen. A comprehensive theory of volumetric radiance estimation using photon points and beams. *ACM Transactions on Graphics (Presented at SIGGRAPH)*, 30(1):5:1âĂŞ5:19, jan 2011b. doi: 10.1145/1899404.1899409.

Wojciech Jarosz, Derek Nowrouzezahrai, Robert Thomas, Peter-Pike Sloan, and Matthias Zwicker. Progressive photon beams. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 30(6), December 2011c.

Johannes Jendersie. Path throughput importance weights. *CoRR*, abs/1806.01005, 2018. URL http://arxiv.org/abs/1806.01005.

Shaojie Jiao. *Photon Surfaces: Volumetric Light Transport Using Multiple Importance Sampled 2D Photon Samples*. Master's thesis, Dartmouth College, October 2018.

Thouis R. Jones and Ronald N. Perry. Antialiasing with line samples. In *Rendering Techniques (Proc. EGWR)*, London, UK, 2000. Springer-Verlag. ISBN 3-211-83535-0.

James T. Kajiya. The rendering equation. *Proc. SIGGRAPH*, 20(4), August 1986a. ISSN 0097-8930.

James T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, August 1986b. ISSN 0097-8930. doi: 10.1145/15886.15902. URL http://doi.acm.org/10.1145/15886.15902.

Eric Lafortune and Yves Willems. Rendering participating media with bidirectional path tracing. *Photorealistic Rendering Techniques (Proc. EGWR)*, 1996.

Eric P. Lafortune and Yves D. Willems. Bi-directional path tracing. In *PROCEEDINGS OF THIRD INTERNATIONAL CONFERENCE ON COMPUTATIONAL GRAPHICS AND VISUALIZATION TECHNIQUES (COMPUGRAPHICS âĂŹ93*, pages 145–153, 1993.

Fred E. Nicodemus. Directional reflectance and emissivity of an opaque surface. *Appl. Opt.*, 4(7):767–775, Jul 1965. doi: 10.1364/AO.4.000767. URL http://ao.osa.org/abstract.cfm?URI=ao-4-7-767.

Derek Nowrouzezahrai, Ilya Baran, Kenny Mitchell, and Wojciech Jarosz. Visibility silhouettes for semi-analytic spherical integration. *Computer Graphics Forum*, 33(1), February 2014.

Vincent Pegoraro and Steven G. Parker. An analytical solution to single scattering in homogeneous participating media. *Computer Graphics Forum (Proc. Eurographics)*, 28 (2), 2009.

Gurprit Singh and Wojciech Jarosz. Convergence analysis for anisotropic monte carlo sampling spectra. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 36(4), jul 2017. doi: 10.1145/3072959.3073656.

Gurprit Singh, Bailey Miller, and Wojciech Jarosz. Variance and convergence analysis of monte carlo line and segment sampling. *Computer Graphics Forum (Proceedings of EGSR)*, 36(4), jun 2017. doi: 10.1111/cgf.13226.

Jerome Spanier. Two pairs of families of estimators for transport problems. *SIAM Journal on Applied Mathematics*, 14(4), 1966. ISSN 00361399.

Jerome Spanier and Ely Meyer Gelbard. *Monte Carlo principles and neutron transport problems*. Addison-Wesley, 1969.

Bo Sun, Ravi Ramamoorthi, Srinivasa G. Narasimhan, and Shree K. Nayar. A practical analytic single scattering model for real time rendering. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 24(3), 2005.

Xin Sun, Kun Zhou, Jie Guo, Guofu Xie, Jingui Pan, Wencheng Wang, and Baining Guo. Line segment sampling with blue-noise properties. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 32(4), July 2013.

Stanley Tzeng, Anjul Patney, Andrew Davidson, Mohamed S. Ebeida, Scott A. Mitchell, and John D. Owens. High-quality parallel depth-of-field using line samples. In *Proceedings of HPG*, 2012. ISBN 978-3-905674-41-5.

Eric Veach. *Robust Monte Carlo methods for light transport simulation.* PhD thesis, Stanford, CA, USA, December 1997.

Eric Veach and Leonidas Guibas. Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques (Proc. EGWR)*, 1994.

Eric Veach and Leonidas Guibas. Optimally combining sampling techniques for Monte Carlo rendering. *Proc. SIGGRAPH*, 29, 1995.

Eric Veach and Leonidas Guibas. Metropolis light transport. *Proc. SIGGRAPH*, 31, 1997.