UNIVERSITY OF CALIFORNIA, SAN DIEGO

Light Portals : Light Transport Variance Reduction

A Thesis submitted in partial satisfaction of the
requirements for the degree of Master of Science

in

Computer Science

by

Phi Hung Le Nguyen

Committee in charge:

Henrik Jensen, Chair
David Kriegman
Jurgen Schulze

2014

The Thesis of Phi Hung Le Nguyen is approved and is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2014

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to thank Henrik Jensen for his usually weekly insights that helped guide me back on course often away from premature optimizations.

I would like to thank David Kriegman for allotting his valuable time to read and evaluate my thesis, and especially for the conversations about his entrepreneur experience at TAAZ he graciously shared.

I would like to thank Jurgen Schulze for allocating his precious time to read and evaluate my thesis, and particularly for the inestimable experience I gained from working in his Immersive Visualization Lab.

I would like to acknowledge the individuals at UCSD's Graphics, Machine Learning, and Vision labs for their occasional conversations that shed new light on useful algorithmic tools and concepts in their respective fields.

ABSTRACT OF THE THESIS

Light Portals : Light Transport Variance Reduction

by

Phi Hung Le Nguyen

Master of Science in Computer Science

University of California, San Diego, 2014

Henrik Jensen, Chair

Monte Carlo integration is a technique that numerically approximates a definite integral using random samples. Existing global illumination algorithms leverage this technique to estimate realistic lighting of 3D scenes. To half that estimation error, four times as many samples are needed. This thesis proposes an automated practical way of analyzing a given 3D scene and generate a modest amount of samples that dramatically reduces the estimation error. The results demonstrate that the estimation improved by several orders of magnitude despite using less samples.

Chapter 1

Introduction

Realistic image synthesis is a domain in the field of computer graphics; its end goal is to generate new images from a complete description of a 3D scene. The description typically consists of the size and location of objects, each object's material properties, the position and emission characteristics of light sources, and a virtual camera. Realism is achieved when the difference between a synthetic image and a photograph of the same scene in reality is not noticeable [Pattanaik et al., 1997].

Photorealistic rendering has many applications in a variety of fields. Visual effects in the film industry is a major driving force for the development of new computer graphics techniques [Sadeghi et al., 2010]. Architectural design benefits immensely from the ability to simulate indoor and outdoor illumination at various times during the day and night [Ward, 1994]. Given the design of a building, ergonomic adjustments to improve comfortability can be validated through simulating the change in wall materials, or even the relocation of walls and windows. Other equally important areas of design such as advertising, appliances, automobiles, consumer electronics, furniture, etc...leverage photorealistic rendering to create accurate imagery of yet-to-be-produced products. Recently, the advancement of hardware and graphics algorithms such as [Kaplanyan and Dachsbacher, 2010] enabled real-time visualization of virtual worlds with comparable realism previously seen only in the film industry.

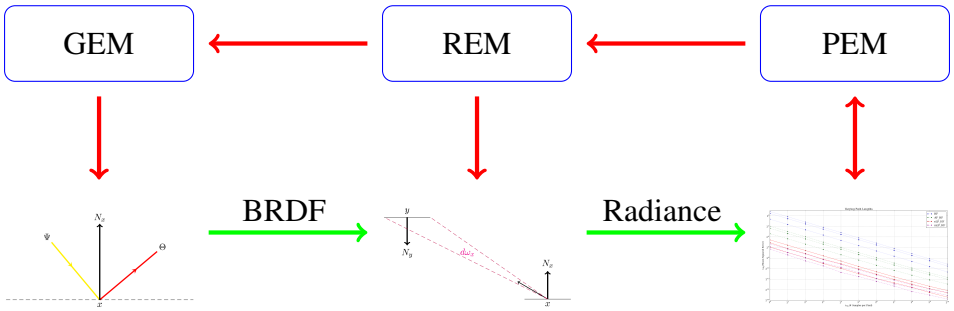


Figure 1.1. Illustration of rendering system stages: local light reflection model, global light transport simulations, and image display algorithms [Greenberg et al., 1997]. GEM (Goniometric Error Metric), REM (Radiometric Error Metric), and PEM (Perceptual Error Metric) can be computed at each stage and fed back into the rendering system to improve image quality and system performance.

A photorealistic rendering system can be subdivided into three separate stages as shown in Figure 1.1. The goal of the first stage is to develop a general purpose wavelength-dependent model or models for arbitrary reflectance functions e.g. subsurface scattering, re-emission, texturing, and surface anisotropy. These compact, data-efficient models are validated through comparing with physical measurements gathered from a gantry.

The middle stage focuses on the creation of global illumination algorithms capable of accurately computing the light energy distribution within complex geometric environments consisting of surfaces with arbitrary reflection functions. The resulting radiometric values are validated against actual measurements of physical environments captured using photometric cameras.

The last stage occurs in the perceptual domain, unlike the earlier stages whose focus is on simulating and comparing physical processes. This stage takes the previously computed radiometric values and synthesizes photorealistic images that are perceptually indistinguishable from real scenes. The quality of a synthetic image is determined using perceptual error metrics; the metrics additionally provides a feedback loop for reducing the magnitude of computations in the earlier stages i.e. improve efficiency of local light reflectance models and global illumination algorithms.

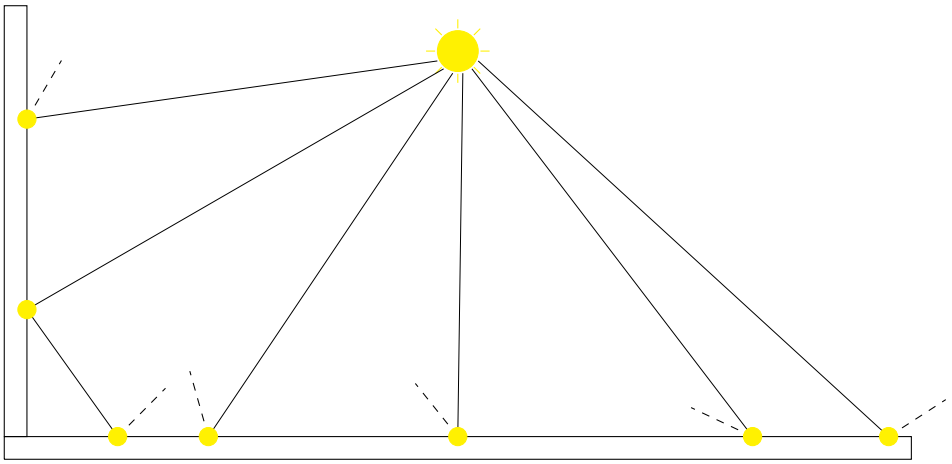


Figure 1.2. Depiction of tracing photons.

With the rendering system in mind, this thesis focuses on a problem that all of the existing global illumination algorithms based on Monte Carlo methods experience: a large number of samples are needed to create a high quality image. This is not a new problem, and several sampling strategies have been proposed over the years to overcome this drawback.

[Jensen, 1995] introduced the use of photon maps as a sampling scheme. The photon map represents a distribution of photons throughout the scene; it is created by emitting a large number of photons for each light source into the scene based upon the emissive characteristics of the light source (see Figure 1.2). A photon is stored at an intersection point when the object it hits is non-specular and non-emitting; explicitly the direction the photon arrived at the surface, its position, and energy level are stored. Based on the density of photons in the map, samples were cast in those directions that contributes mostly to the reflected radiance, which is a radiometric quantity describing the density of light flowing through a given point in a given direction. The downside of this approach is that it requires storing photons all over the scene even in regions that contribute very little. For scenes with strong indirect illumination, a lot of photons are required to remove all the visible noise from the synthesized image.

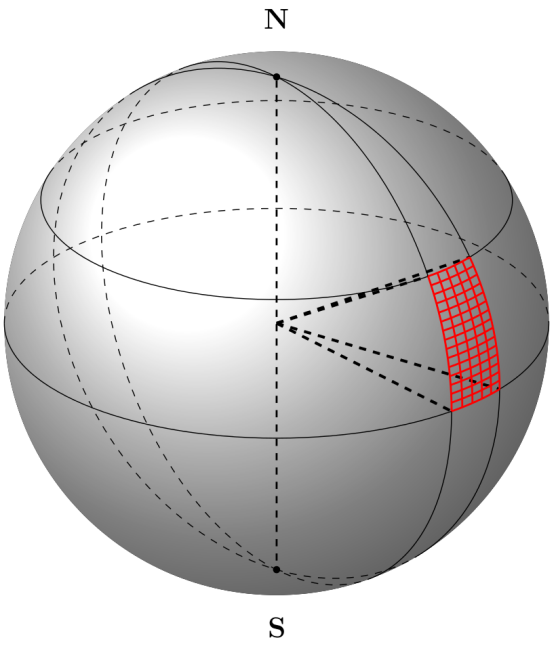


Figure 1.3. An irradiance volume is a sphere that can be decomposed into a hemispherically-mapped grid storing irradiance.

[Greger et al., 1998] proposed the irradiance volume as a trade-off of accuracy for high performance where irradiance is radiance in the reverse direction. Irradiance volumes are a 3D set of diffuse environment maps located in an evenly spaced lattice within the scene. Essentially, irradiance is stored instead of spheres as shown in Figure 1.3. This approach ignores surface interreflections and only handles static lighting i.e. ambient term of lighting model. The irradiance for some point with a normal direction is interpolated from the values stored at the nearest irradiance volumes in the lattice. This approximation fails at transition zones between umbra-penumbra and penumbra-unshadowed regions.

Instead of controlling emission and scattering of photons, [Keller and Wolf, 2000] controls the deposition of the photons. They coined the term importons as photons that are visually important to the scene i.e. affects what is seen by the virtual camera. Starting from the camera sensor, importons are shot into the scene. The deposition of photons are determined based on this distribution. [Keller and Wolf, 2000] illustrates this nicely

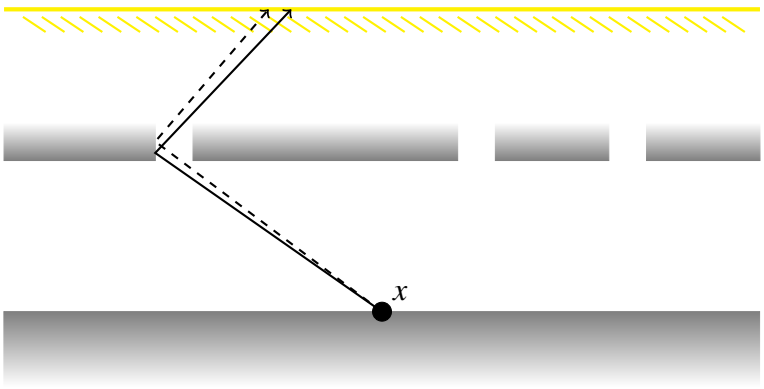


Figure 1.4. MLT tends to get stuck in one region of light for a long time before moving to the other bright regions in the scene.

using a scene with 10×10 rooms each with one light source seen from the top. Using the traditional approach to photon mapping, each room would contribute to the fixed number of photons allocated to the scene. The camera, however, only receives contribution from the rooms closest to it; the use of importons allows for more efficient photon maps to be generated. There are two flaws with this method. Due to energy bleeding, the importance shines through the thin walls resulting in photons being deposited in areas of no importance i.e. behind the wall. The other issue is that this becomes just as inefficient as [Jensen, 1995] when the photons must travel through narrow openings (e.g. tiny windows or door slits) to arrive at important regions in the scene.

Metropolis Light Transport (MLT) is a widely acknowledged adaptive sampling algorithm that handles scenes with narrow openings. It uses the history of the previous sample to choose a new sample on bright regions in the scene. The idea is that since a path hits the light source, nearby paths will most likely hit that light source as well. Nearby paths are created through randomly perturbing the locations of the intersection points. [Veach and Guibas, 1997] explains in detail how to do this in an unbiased way. Unfortunately, since this sampling scheme is based off random walk, it is possible for the algorithm to frequently get stuck in some bright region in the scene as shown in

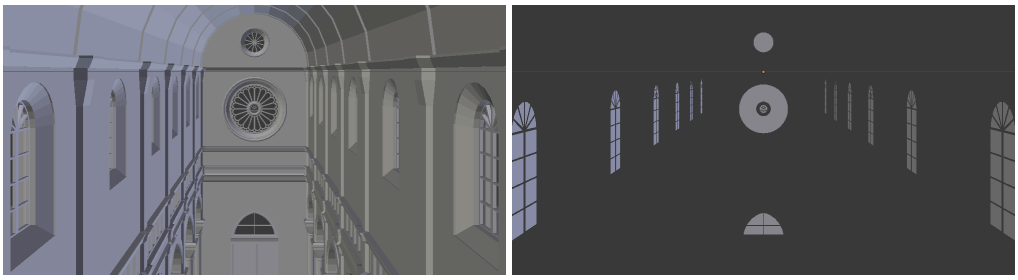


Figure 1.5. Sibenik without windows; one light portal for each opening.

Figure 1.4.

To handle multiple disjoint important regions in the scene, [Hachisuka et al., 2008] uses the entire history of all samples to generate new samples. The idea is to coarsely sample the scene, and then place more samples in areas that have the highest variance according to some metric. Overtly, this uses a generous amount of memory, and the initial sampling may miss small features in the scene.

The most robust memory efficient solution that achieves a dramatic reduction in the number of samples while improving in the quality of the image is manual creation of light portals. A light portal is an invisible planar geometry (e.g. triangles, discs, or planes) artists place at the openings that light enters into the scene [Ward, 1994] (see Figure 1.5). Light is then guided into the scene via sampling these light portals. This solution, however, is not scalable. Suppose a scene has a hundred narrow openings, in the current pipeline, the artist needs to manually place a light portal at each opening. On top of that, the artist needs to be trained in identifying the appropriate place to put the light portals. Placing a light portal at an entrance where there is no light will waste a lot of samples resulting in a final image that is inferior compared to one that does not use any light portals.

1.1 Contribution

This thesis introduces a practical method to automatically identify and create light portals for common architectural scenes.

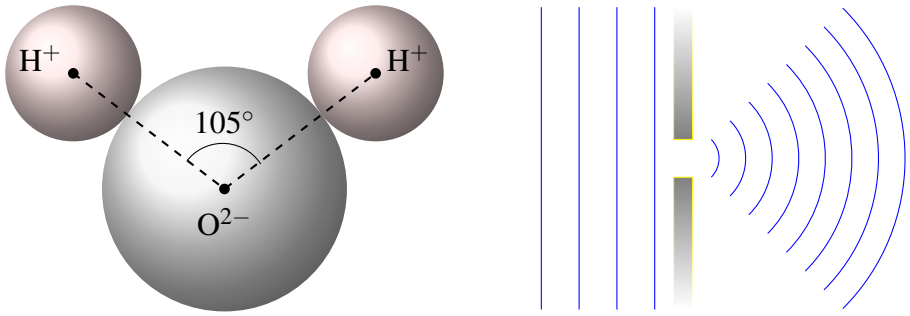
Chapter 2 goes over the fundamentals of light transport assumed in the remainder of the thesis. Chapter 3 briefly examines the classic global illumination algorithm that is used in the results. Chapter 4 describes the internal workings of identifying and fitting a proper light portal. Chapter 5 illustrates the benefits and limitations of the current approach. Finally, the thesis concludes with potential future ventures in chapter 6.

Chapter 2

Light Transport

Light can be modeled either as a wave or a stream of particles; each model captures a different set of light behavior.

The quantum optics model can explain light interactions at the submicroscopic level e.g. electrons and protons. Currently at this level, visualizations focus on displaying the topology and structure of atoms and molecular bonds in a manner that enhances collaborative workspaces [Schulze et al., 2012]. As shown in Figure 2.1a, the colors in the renderings were artistically chosen to stand out and highlight the interactions between molecules. A simplification of the quantum model is the wave model described by Maxwell's equations. This model captures the interaction between light and objects whose size is comparable to the wavelength of light (see Figure 2.1b).



(a) Quantum Model **(b)** Wave Model

Figure 2.1. Illustration of water molecule and diffraction of light.

The geometric optics model is predominantly used in computer graphics due to its simplicity [Dutre et al., 2003]; it assumes:

- The scale of the objects light interacts with is much larger than the wavelength of the light.
- Light travels in a straight line i.e. effects such as diffraction are ignored.
- External factors such as gravity and magnetic fields do not affect light propagation [Einstein, 1911].

2.1 Radiometry

The physical measurement of light energy is in radiometric units. Radiant power is the fundamental radiometric quantity; it is also known as flux commonly denoted as Φ expressed in watts (W). From the definition of watts, this quantity measures how much total energy flows from, to, or through a surface per unit of time.

Irradiance (E) is the incident radiant power on a surface per unit surface area. Radiance is flux per unit projected area per unit solid angle; mathematically, it is a five-dimensional quantity varying over position and direction:

$$L(x,\Theta) = \frac{d^2\Phi}{d\omega dA^{\perp}} = \frac{d^2\Phi}{d\omega dA\cos(\theta)} \tag{2.1}$$

As depicted in Figure 2.2, the cosine term can be thought as a necessary component to uphold the definition of radiance. The projected area is the area of the surface projected perpendicular to the direction of interest. In both illustrations, the normal vector N_x in $\cos(N_x,\Psi) = \theta$ is perpendicular to the horizontal surface x where Ψ is the negated direction of the light. In the scenario at the left, since Ψ is parallel to N_x, $\theta = 1$ thus preserving the definition of radiance. In the scenario at the right, radiance arrives at a

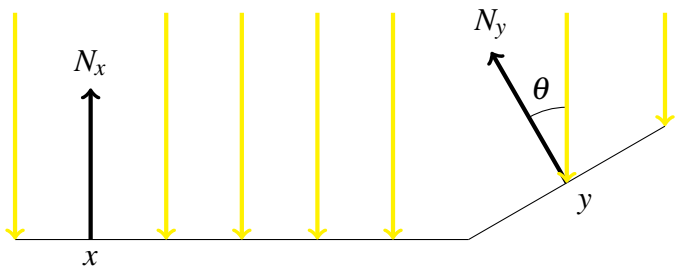


Figure 2.2. Intuition behind cosine term.

grazing angle, so it is smeared out over a large surface. To consistently use the definition of radiance, the larger area needs to be taken into account via θ.

2.2 Rendering Equation

Equation (2.2) is the hemispherical formulation of the rendering equation proposed in [Kajiya, 1986]; it describes the equilibrium distribution of light energy in a scene.

$$\begin{aligned} L(x \to \Theta) &= L_e(x \to \Theta) + L_r(x \to \Theta) \\ &= L_e(x \to \Theta) + \int_{\Omega_x} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi \end{aligned} \tag{2.2}$$

$L(x \to \Theta)$ is the exitant (outgoing) radiance at each surface point x in each direction Θ. $L_e(x \to \Theta)$ represents the radiance emitted by the surface at x in the outgoing direction Θ, and $L_r(x \to \Theta)$ represents the radiance that is reflected by the surface at x in that direction Θ. See Figure 2.3 for a depiction of the hemispherical rendering equation.

Computing $L_r(x \to \Theta)$ requires integrating irradiance over the entire hemisphere at x. The solid angle Ω is used as a measure on the hemisphere; $d\omega_\Psi$ represents an infinitesimal solid angle along direction Ψ towards x. f_r models the interaction between x's surface and the radiance coming from Ψ reflected towards Θ.

Even though the unknown quantity radiance appears on both sides of the equation,

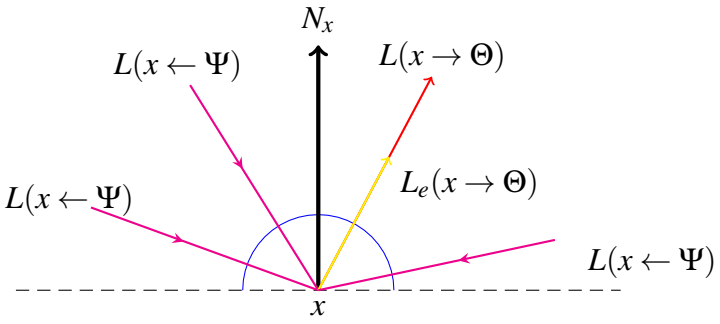


Figure 2.3. Incident radiance integrated over the hemisphere.

it is still computable through exploiting the properties of radiance.

2.2.1 Properties of Radiance

[Dutre et al., 2003] proved that radiance is invariant along straight paths:

$$L(x \rightarrow y) = L(y \leftarrow x).$$

Assuming light is traveling through a vacuum i.e. there is no participating medium, radiance leaving a point x directed towards point y is equal to the radiance arriving at y from x (see Figure 2.4). From the definition of radiance, the total differential power leaving a differential surface area dA_x and arriving at a differential surface area dA_y is

$$L(x \rightarrow y) = \frac{d^2\Phi}{\cos(\theta_x) dA_x d\omega_{x \leftarrow dA_y}}$$

$$d^2\Phi = L(x \rightarrow y) \cos(\theta_x) dA_x d\omega_{x \leftarrow dA_y}$$

where $d\omega_{x \leftarrow dA_y}$ denotes the solid angle subtended by dA_y as seen from x. Similarly, the

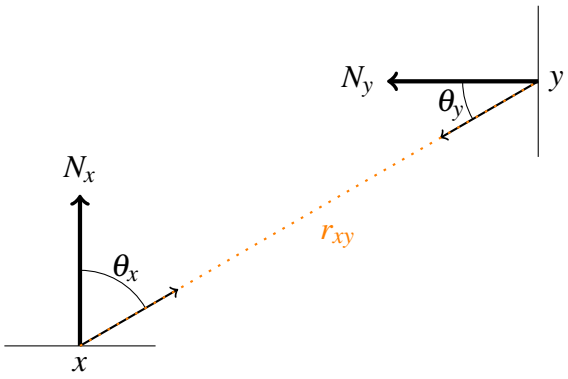


Figure 2.4. Invariance of radiance along straight paths.

power that arrives at dA_y from dA_x is

$$L(y \leftarrow x) = \frac{d^2\Phi}{\cos(\theta_y) dA_y d\omega_{y \leftarrow dA_x}}$$
$$d^2\Phi = L(y \leftarrow x) \cos(\theta_y) dA_y d\omega_{y \leftarrow dA_x}.$$

By definition of the solid angle,

$$d\omega_{x \leftarrow dA_y} = \frac{\cos(\theta_y) dA_y}{r_{xy}^2} \quad \wedge \quad d\omega_{y \leftarrow dA_x} = \frac{\cos(\theta_x) dA_x}{r_{xy}^2}.$$

Assuming energy is not loss or added to the simulation, from the law of conservation of energy,

$$L(x \rightarrow y) \cos(\theta_x) d\omega_{x \leftarrow dA_y} dA_x = L(y \leftarrow x) \cos(\theta_y) d\omega_{y \leftarrow dA_x} dA_y$$
$$L(x \rightarrow y) \cos(\theta_x) \frac{\cos(\theta_y) dA_y}{r_{xy}^2} dA_x = L(y \leftarrow x) \cos(\theta_y) \frac{\cos(\theta_x) dA_x}{r_{xy}^2} dA_y.$$

Thus

$$L(x \rightarrow y) = L(y \leftarrow x)$$

meaning radiance is invariant along straight paths and does not attenuate with distance.

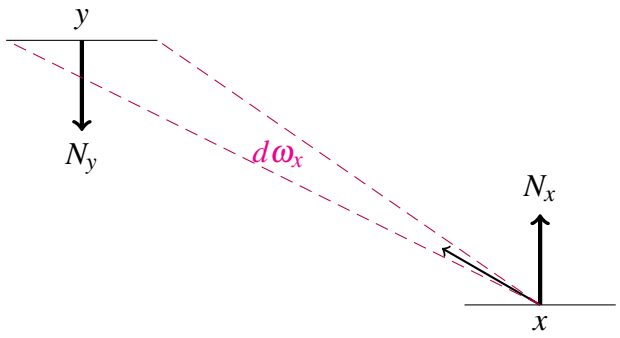


Figure 2.5. Area formulation of the rendering equation.

The second property of radiance states that the response of sensors (e.g. cameras and the human eye) is proportional (dependent on the geometry of the sensor) to the radiance incident upon them; this explains why the color of an object does not change with distance.

Using the above observations, the rendering equation can be evaluated recursively.

2.3 Area Formulation

An alternative formulation of the rendering equation, which is used in the remainder of this thesis, replaces the integration over the hemisphere with integration over all surfaces visible at the point; this is known as the area formulation [Kajiya, 1986]. This formulation defines the ray-casting operation, denoted as $r(x,\Psi)$, as finding the point on the closest visible object along a ray originating at point x and pointing in the direction of Ψ. Formally,

$$r(x,\Psi) = \{y : y = x + t_{\text{intersection}}\Psi\}$$

$$t_{\text{intersection}} = \min\{t : t > 0, x + t\Psi \in A\}$$

$$\forall x,y \in A : V(x,y) = \begin{cases} 1, & \text{if } x \text{ and } y \text{ are mutually visible} \\ 0, & \text{if } x \text{ and } y \text{ are not mutually visible} \end{cases}$$

where A represents all surfaces in the scene, and $V(x,y)$ handles the case where the ray-casting operation does not hit any surface in the scene (see Figure 2.5). Lastly, the differential solid angle can be recast as follows

$$d\omega_{\Psi} = d\omega_{x \leftarrow dA_y} = \cos(N_y, -\Psi)\frac{dA_y}{r_{xy}^2}$$

by definition of the solid angle. The following is the area formulation of the rendering equation

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_A f_r(x, \Psi \to \Theta) L(y \leftarrow -\Psi) V(x,y) G(x,y) dA_y \tag{2.3}$$

where

$$G(x,y) = \frac{\cos(N_x, \Psi)\cos(N_y, -\Psi)}{r_{xy}^2}$$

is called the geometry term.

It should be noted that the hemispherical formulation is efficient for sampling indirect illumination while the area formulation is efficient for direct illumination:

$$\begin{aligned}
L(x \to \Theta) &= L_e(x \to \Theta) + L_r(x \to \Theta) \\
L_r(x \to \Theta) &= L_{\text{direct}} + L_{\text{indirect}} \\
L_{\text{direct}} &= \int_A f_r(x, \overrightarrow{xy} \to \Theta) L_e(y \to \overrightarrow{yx}) V(x,y) G(x,y) dA_y \\
L_{\text{indirect}} &= \int_{\Omega_x} f_r(x, \Psi \to \Theta) L_i(x, \leftarrow \Psi) \cos(N_x, \Psi) d\omega_{\Psi}
\end{aligned}$$

where

$$y = r(x, \overrightarrow{xy}) \quad \wedge \quad L_i(x \leftarrow \Psi) = L_r(r(x, \Psi) \to -\Psi).$$

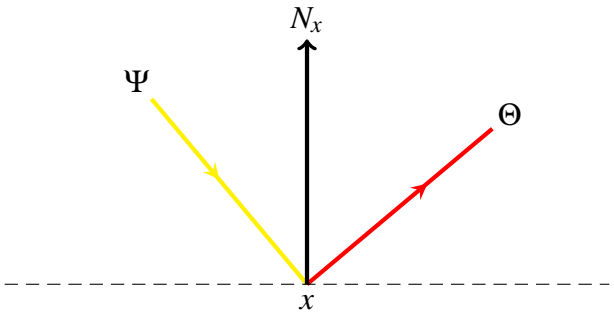


Figure 2.6. Bidirectional reflectance distribution function.

2.4 Intersection between Light and Surfaces

Given the same lighting conditions, materials interact with light in different ways. The most general case is when light enters some surface at point p with incident direction Ψ and leaves the surface at some other point q with exitant direction Θ; the function that defines this behavior is known as the bidirectional subsurface scattering reflectance distribution function. This thesis implements the simpler function known as the bidirectional reflectance distribution function (BRDF) i.e. light incident at a surface exists at the same wavelength, time, and location (see Figure 2.6).

The BRDF at a point x is defined as the ratio of the differential radiance reflected in an exitant direction (Θ), and the differential irradiance incident through a differential solid angle $(d\omega_\Psi)$:

$$f_r(x, \Psi \rightarrow \Theta) = \frac{dL(x \rightarrow \Theta)}{L(x \leftarrow \Psi)\cos(N_x, \Psi)d\omega_\Psi} \tag{2.4}$$

where $\cos(N_x, \Psi)$ is the cosine of the angle formed by the normal N_x at point x and the incident direction Ψ. As mentioned earlier, these material BRDFs are usually characterized empirically; each model must satisfy the following properties:

Range

The BRDF can vary with wavelength and take on any positive value.

Dimension

The BRDF is defined at each point on a surface characterized by the incident and exitant directions.

Helmholtz Reciprocity

The amount of light reflected by the BRDF does not change when the direction of light is reversed:

$$f_r(x, \Psi \rightarrow \Theta) = f_r(x, \Theta \rightarrow \Psi).$$

Independence

The value of the BRDF for a specific incident direction is independent of the irradiance along other incident angles.

Energy Conservation

The total amount of power reflected over all directions must be less than or equal to the total amount of power incident on the surface.

The classic diffuse surface reflect light uniformly over the entire hemisphere i.e. reflected radiance is independent of the exitant direction. The ideal diffuse reflector's BRDF can be derived as follows:

$$\begin{aligned}
L(x \rightarrow \Theta) &= \int_{\Omega_x} f_r(x, \Psi \rightarrow \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi \\
p_d &= \int_{\Omega_x} f_r(x, \Psi \rightarrow \Theta) \cos(N_x, \Psi) d\omega_\Psi \\
p_d &= \int_0^{2\pi} \int_0^{\frac{\pi}{2}} f_r(x, \varphi, \theta) \cos(\theta) \sin(\theta) d\theta d\varphi \\
p_d &= f_r(x) \left[\frac{-\cos(\theta)^2}{2} \right]_0^{\frac{\pi}{2}} \int_0^{2\pi} d\varphi \\
\frac{p_d}{\pi} &= f_r(x)
\end{aligned}$$

where p_d is the fraction of incident energy that is reflected at x. Note that physically-based materials impose $0 \leq p_d \leq 1$. Similarly, the classic diffuse light source emits equal radiance in all directions from all its surface points. The power for the diffuse emitter is defined as

$$\begin{aligned}\Phi &= \int_A \int_\Omega L(x \to \Theta)\cos(\theta) d\omega_\Theta dA_x \\ &= \int_A \int_\Omega L\cos(\theta) d\omega_\Theta dA_x \\ &= L\left(\int_A dA_x\right)\left(\int_\Omega \cos(\theta) d\omega_\Theta\right) \\ &= LA\pi\end{aligned}$$

A perfectly specular surface that reflects or refracts light in one specific direction is another classic material. According to the law of reflection, the incident and exitant light direction make equal angles to the surface's normal and lie in the same plane as the normal. Suppose the light is coming at the specular surface along Ψ and the normal to the surface is N, the incident light is reflected along

$$R = 2(N \cdot \Psi)N - \Psi.$$

To model this physical behavior, delta functions are needed. Fortunately, the evaluation of the BRDF for perfectly specular surfaces can be done using a ray-casting operation

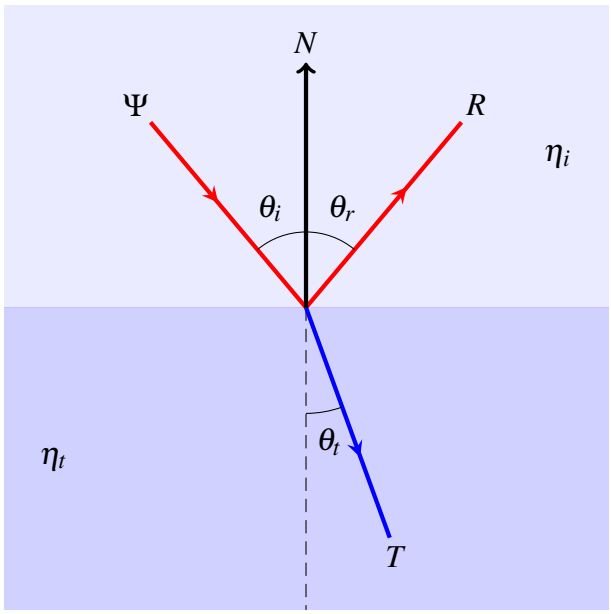


Figure 2.7. Perfect specular reflection and refraction with $\theta_i = \theta_r$.

[Nicodemus, 1977]:

$$
\begin{aligned}
L(x \rightarrow \Theta) &= \int_{\Omega_x} f_r(x, \Psi \rightarrow \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi \\
L_r &= \int_0^{2\pi} \int_0^{\frac{\pi}{2}} f_r(x, (\theta_i, \varphi_i) \rightarrow (\theta_r, \varphi_r)) L_i \cos(\theta_i) \sin(\theta_i) d\theta_i d\varphi_i \\
L_r &= \int_0^{2\pi} \int_0^{\frac{\pi}{2}} p_s 2\delta(\sin^2\theta_r - \sin^2\theta_i)\delta(\varphi_r \pm \pi - \varphi_i) L_i \cos(\theta_i) \sin(\theta_i) d\theta_i d\varphi_i \\
L_r &= \int_0^{2\pi} \int_0^{\frac{\pi}{2}} p_s \frac{\delta(\theta_r - \theta_i)\delta(\varphi_r \pm \pi - \varphi_i)}{\cos(\theta_i)\sin(\theta_i)} L_i \cos(\theta_i) \sin(\theta_i) d\theta_i d\varphi_i \\
L_r &= p_s \int_0^{2\pi} \int_0^{\frac{\pi}{2}} L_i \delta(\theta_r - \theta_i)\delta(\varphi_r \pm \pi - \varphi_i) d\theta_i d\varphi_i \\
L_r &= p_s L_i
\end{aligned}
$$

where p_s is the fraction of incident energy that is reflected at x and δ is the Dirac delta function. Again, note that physically-based materials impose $0 \leq p_s \leq 1$.

According to Snell's law, when light travels in a medium with refractive index η_i

to a medium with refractive index η_t, the refracted ray is given as

$$T = -\frac{\eta_i}{\eta_t}\Psi + N\left(\frac{\eta_i}{\eta_t}(N\cdot\Psi) - \sqrt{1-\left(\frac{\eta_i}{\eta_t}\right)^2(1-(N\cdot\Psi)^2)}\right),$$

which can be derived from the following invariant between the angle of incidence and refraction and the refractive indices of the media

$$\eta_i \sin(\theta_i) = \eta_t \sin(\theta_t).$$

Notice that the term under the square root could be less than zero when there is total internal reflection. This happens when light travels from a dense medium to a rare medium and gets refracted back into the dense medium, which arises at Brewster's angle θ_c such that

$$\eta_i \sin(\theta_c) = \eta_t \sin(\frac{\pi}{2})$$
$$\sin(\theta_c) = \frac{\eta_t}{\eta_i}.$$

To compute the fraction of energy that is reflected and refracted from a perfectly smooth surface, Fresnel equations are needed. The Fresnel equations assume that unpolarized light is either reflected or refracted at a purely specular surface, so $F_r + F_t = 1$ where

$$F_r = \frac{1}{2}\left(r_\parallel^2 + r_\perp^2\right)$$
$$r_\parallel = \frac{\eta_t\cos(\theta_i) - \eta_i\cos(\theta_t)}{\eta_t\cos(\theta_i) + \eta_i\cos(\theta_t)}$$
$$r_\perp = \frac{\eta_i\cos(\theta_i) - \eta_t\cos(\theta_t)}{\eta_i\cos(\theta_i) + \eta_t\cos(\theta_t)}.$$

Chapter 3

Stochastic Path-Tracing

Synthesizing an image is essentially gathering a weighted measure of radiance values incident on the image plane. [Dutre et al., 2003] describes this process as

$$\begin{aligned} L_{\text{pixels}} &= \int_{\text{imageplane}} L(p \rightarrow \text{eye}) h(p) dp \\ &= \int_{\text{imageplane}} L(x \rightarrow \text{eye}) h(p) dp \end{aligned} \tag{3.1}$$

with p being a point on the image plane, and $h(p)$ is a filtering function (see Figure 3.1). $h(p)$ can be a simple box filter such that the final radiance value is the uniform average of the incident radiance values sampled over the area of the pixel. Notice that to evaluate $L(p \rightarrow \text{eye})$, a ray is cast from the eye through p; in this case, that ray hits x. Since radiance is invariant along straight paths, $L(p \rightarrow \text{eye}) = L(x \rightarrow \overrightarrow{xp})$, which can be solved for using the rendering equation. Unfortunately, it is too complex to be analytically evaluated for a generic scene; Monte Carlo integration is a method to solve L numerically.

3.1 Monte Carlo Integration

Monte Carlo is a term that describes mathematical techniques that use statistical sampling to simulate phenomena or evaluate values of functions. To compute the value

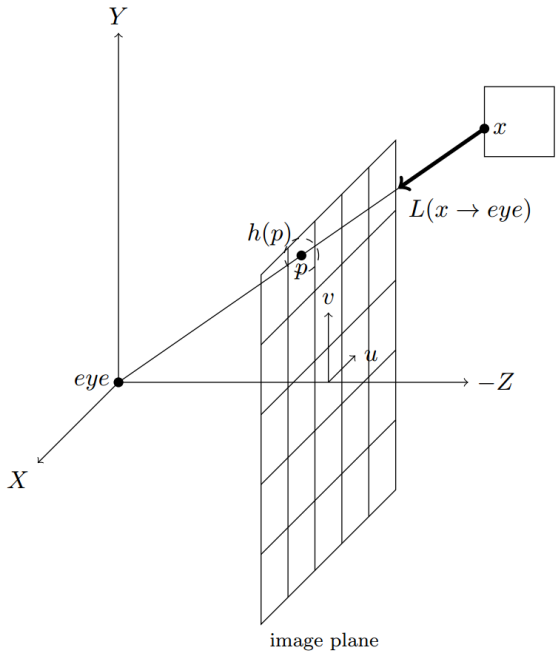


Figure 3.1. Ray-tracing setup.

of the integration of a function with respect to some measure over a domain is to define a random variable such that the expected value of that random variable would be the solution to the problem [Kalos and Whitlock, 2008]. The average over samples drawn from that random variable approximates the solution of the integration. The downside to using Monte Carlo techniques is their slow convergence rate of $1/\sqrt{N}$ where N is the number of samples; this is why Monte Carlo techniques are used as a last resort in tackling complex integrands. Unfortunately, Monte Carlo is the only viable option for solving the rendering equation due to its nonsmooth and high-dimensional characteristics.

Concretely, a Monte Carlo estimator for an integral $I = \int f(x)dx$ is

$$\langle I \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}$$

where N is the total number of samples to use in the estimator, f is the complicated function to integrate over some domain, and p is the probability density function (PDF)

for randomly sampling x_i from the domain to evaluate at f.

A PDF $p(x)$ describes a continuous random variable (versus a discrete one) such that the probability that the variable takes a value x in the interval $[x, x+dx]$ equals $p(x)dx$. To define a valid PDF, the cumulative distribution function (CDF) is needed. A CDF is defined as the probability an event occurs with an outcome whose value is less than or equal to the value y:

$$P(y) = Pr(x \leq y) = \int_{-\infty}^{y} p(x)dx$$

where $P(y)$ is a nondecreasing function and is non-negative over the domain of the random variable. A valid PDF $p(x)$ has the following properties:

- $\forall x, p(x) \geq 0$.
- $\int_{-\infty}^{\infty} p(x)dx = 1$.
- $p(x) = \frac{dP(x)}{dx}$.
- $Pr(a \leq x \leq b) = Pr(x \leq b) - Pr(x \leq a) = \int_a^b p(z)dz$.

Applying the Monte Carlo estimator to the rendering equation yields

$$\langle L_{\text{indirect}}(x \to \Theta) \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{L(r(x, \Psi_i) \to -\Psi_i) f_r(x, \Theta \leftrightarrow \Psi_i) \cos(\Psi_i, N_x)}{p(\Psi_i)} \tag{3.2}$$

$$\langle L_{\text{direct}}(x \to \Theta) \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{L_e(y_i \to \overrightarrow{y_i x}) f_r(x, \Theta \leftrightarrow \overrightarrow{xy_i}) G(x, y_i) V(x, y_i)}{p_L(k_i) p(y_i | k_i)} \tag{3.3}$$

where $p(\Psi_i) = 1/2\pi$ is the probability of picking a random direction on the hemisphere, $p_L(k) = 1/N_L$ is the probability of selecting light source k out of N_L light sources, and $p(y|k) = 1/A_{L_k}$ is the probability of choosing point y on light source k given light source k was selected (see Figure 3.2).

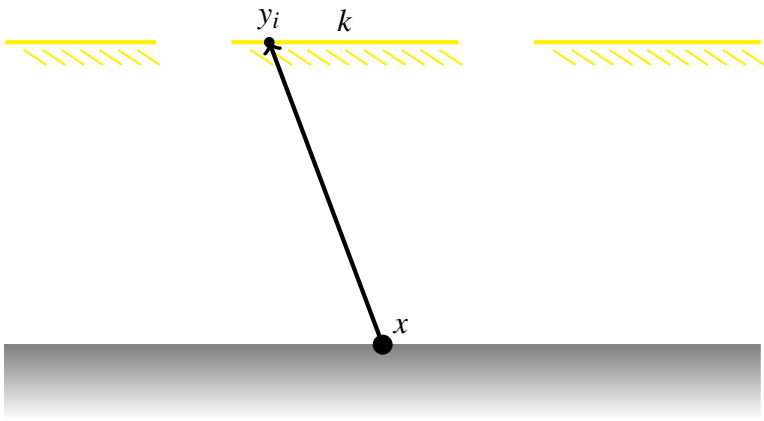


Figure 3.2. Sampling multiple light sources for direct illumination.

Uniformly sampling the hemisphere would result in a PDF of

$$p(\theta,\varphi) = \frac{1}{2\pi}$$

since the surface area of a hemisphere is 2π. [Dutre et al., 2003] has shown that picking samples from a cosine weighted PDF

$$p(\theta,\varphi) = \frac{\cos\theta}{\pi}$$

can further reduce the final variance; this can be accomplished through computing two uniformly distributed samples μ_1 and μ_2 such that

$$\phi_i = 2\pi\mu_1 \qquad \wedge \qquad \theta_i = \cos^{-1}\sqrt{\mu_2}$$

for each sample i on the hemisphere along the Z-axis denoted as e_z. The PDF for a perfectly specular surface is a Dirac delta function. Since the evaluation of f_r consists of computing the reflected or refracted vector, the evaluation of $p(\theta,\varphi)$ is 1.

It should be noted that the randomly generated vector distribution is along e_z.

Rodrigues' rotation formula provides an algorithm to efficiently rotate a vector in space given an axis-angle representation [Gray, 1980]:

$$R = I\cos(\theta) + \sin(\theta)\omega_\times + (1-\cos(\theta))\omega\omega^\top$$

where $R \in SO(3)$ is an orthogonal matrix describing a rotation by an angle θ about a fixed axis specified by the unit vector $\omega \in \mathbb{R}^3$. I is the identity matrix and $\omega_\times$ denotes an antisymmetric matrix with entries

$$\begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}.$$

Since the goal is mapping e_z to N, $\cos(\theta) = e_z \cdot N = N_z$. The desired rotation axis is

$$\widetilde{\omega} = e_z \times N = \begin{bmatrix} -N_y \\ N_x \\ 0 \end{bmatrix} = \sin(\theta)\omega.$$

Thus each vector $v \in \mathbb{R}^3$ can be efficiently rotated as follows

$$\begin{aligned} v_{\text{rot}} &= Rv \\ &= \cos(\theta)Iv + \sin(\theta)\omega_\times v + (1-\cos(\theta))\omega\omega^\top v \\ &= vN_z + \widetilde{\omega}_\times v + \frac{(1-\cos(\theta))\widetilde{\omega}\widetilde{\omega}^\top}{1-\cos^2(\theta)}v \\ &= vN_z + \widetilde{\omega} \times v + \frac{\widetilde{\omega}\widetilde{\omega}^\top}{1+\cos(\theta)}v \\ &= vN_z + \widetilde{\omega} \times v + \widetilde{\omega}\frac{\widetilde{\omega}^\top v}{1+N_z}. \end{aligned}$$

This operation is undefined when $N_z = -1$; one way around this is to solve for $-N$ and invert the resulting vector since the distribution is symmetric.

Clearly these PDFs are uniform and unbiased [Shirley and Wang, 1994]; they are the fastest to evaluate and can be replaced with any other valid PDFs, which may reduce the estimator variance and final image noise or significantly worsen both of them.

3.2 Path Tracing

The idea is to recursively trace a path of randomly selected rays through the scene; the recursive path needs to hit one of the light sources in the scene to contribute a nonzero radiance evaluation. Since the light sources are small compared to the other surfaces, the majority of the paths will miss the light. This process is illustrated in Figure 3.3. Notice that the recursion needs a stopping criterion; otherwise, the generated paths would be of infinite length.

In classic ray-tracing implementations, two techniques are often used to prevent paths from growing too large. The first is to cut off the recursive evaluation after a fixed number of evaluations i.e. the path length is fixed. The other approach is to use an adaptive cut-off path length. Recall that the radiance found at the light source needs to be multipled by cosine factors and BRDF evaluations and divided by PDF values, should this accumulating multiplication factor fall below some threshold, the recursive evaluation is stopped. Unfortunately, both of these approaches will introduce bias to the final image.

3.2.1 Russian Roulette

Russian roulette is a technique that allows exploration of all possible path lengths while keeping the computation manageable [Arvo and Kirk, 1990]. Let $I = \int_0^1 f(x)dx$ be the integral of interest. The standard Monte Carlo integration would use randomly

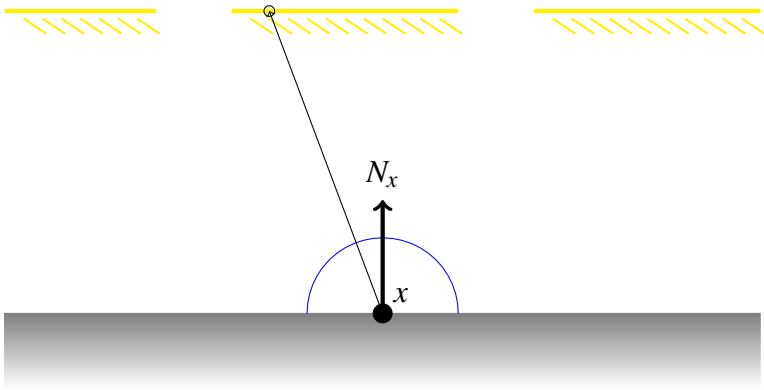


Figure 3.3. Path tracing using hemispherical formulation for indirect illumination.

generated points x_i in the domain of $[0,1]$ and compute the weighted average of all $f(x_i)$. Russian roulette reduces the number of $f(x)$ evaluations needed to estimate I; this is done through scaling $f(x)$ by P horizontally and $1/P$ vertically (see Figure 3.4):

$$I_{RR} = \int_0^P \frac{1}{P} f(\frac{x}{P}) dx$$

where $P \leq 1$. Applying Monte Carlo integration with a uniform PDF $p(x) = 1$ over the domain $[0,1]$ yields:

$$\langle I_{RR} \rangle = \begin{cases} \frac{1}{P} f(\frac{1}{P}) & \text{if } x \leq P \\ 0 & \text{if } x > P. \end{cases}$$

The expected value of $\langle I_{RR} \rangle$ equals I, which demonstrates that this technique is unbiased. In the case of the rendering equation, the recursion stops with probability equal to the absorption probability $\alpha = 1 - P$. Notice that although the samples generated in the interval $[P,1]$ evaluates to 0, this is compensated by weighting the samples in $[0,P]$ with a factor of $1/P$. Specifying α to be small will expand the recursion many times resulting in a more accurate estimator; if α is large, the recursion will end sooner which will increase the variance of the estimator. In practice, global illumination algorithms will evaluate

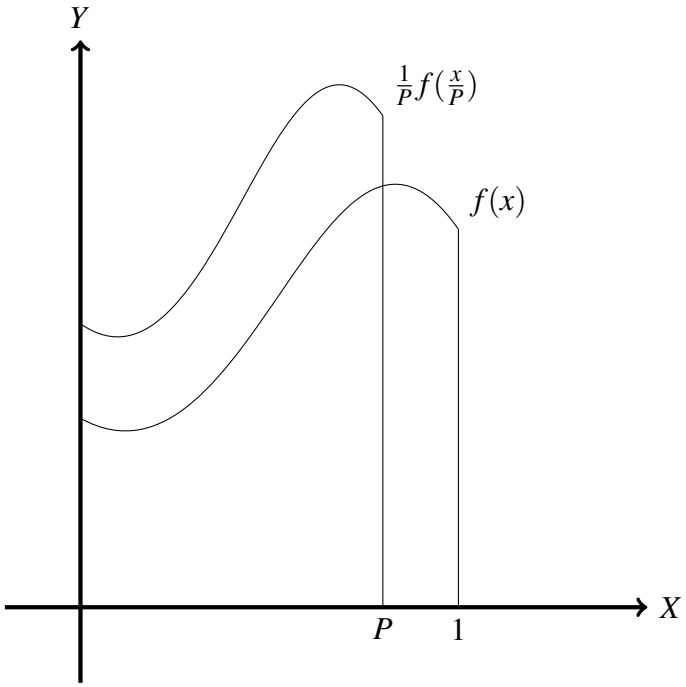


Figure 3.4. Idea behind Russian roulette.

each path up to a fixed minimum path length before terminating with Russian roulette. Any value can be selected for α, but it is commonly set equal to the hemispherical reflectance of the surface's material.

Chapter 4

Light Portals

Existing renderers (e.g. LuxRender, Octane, and RADIANCE) already use light portals to synthesize high quality images quickly. Similar to how the hemispherical formulation allows uniform or cosine weighted PDFs, the area formulation also allows its existing PDF to be replaced with the light portals' more efficient PDF.

The Monte Carlo estimator for the area formulation of the rendering equation requires a few modifications to remain unbiased. Let l_i denote a point on one of the N light portals in the scene randomly chosen according to the distribution $p(l) = (1/N)(1/A_l)$ where A_l is the surface area of the light portal containing l.

$$\langle L_{\mathrm{LP}}(x \to \Theta) \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{L_e(y_i \to \overrightarrow{y_i l_i}) f_r(x, \Theta \leftrightarrow \overrightarrow{x l_i}) G(x, l_i) V(x, y_i)}{p(l_i)} \tag{4.1}$$

where

$$G(x, l) = \frac{\cos(N_x, \Psi) \cos(N_l, -\Psi)}{r_{xl}^2}$$

This estimator is unbiased as long as the light portals cover all the light entrances, which may require the user to verify. One alternative to ensure the estimator unbiased is to combine the hemispherical and area formulations using multiple importance sampling (MIS) to avoid double counting the areas where the integrands overlap [Veach and Guibas, 1995]. Given a set of radiance values $\{L_1, \ldots, L_n\}$ evaluated using n different PDFs, MIS'

balance heuristic allows them to be combined in the following manner to minimize the final variance:

$$L = \sum_{j=1}^{n} w_j L_j \qquad \wedge \qquad w_j = \frac{p_j}{\sum_{k=1}^{n} p_k}.$$

Notice that each PDF is an importance sampling technique on the same domain. Given an area distribution of the light portal $p(l)dA$, the corresponding directional distribution $p(\Psi)d\omega$ is

$$p(\Psi) = p(l)\frac{dA}{d\omega} = p(l)\frac{r_{xl}^2}{\cos(N_l, -\Psi)}.$$

Since light portals capture the flow of light into the scene, evaluating $p(\Psi)$ could result in a division by zero; in this case, the corresponding L_j and p_j can be set to zero.

Using the above framework, existing renderers require artists to provide the light portals. The following sections will describe an automatic way to detect and construct the light portals of a given architectural scene.

4.1 Initial Strategies

As mentioned earlier, a plethora of existing works have tried to improve the sampling scheme of global illumination algorithms. The initial thought was to store all the surface intersections and ray segments of paths that contributed radiance during path tracing [Lafortune and Willems, 1995]. The idea is to analyze the set of surface points, where each point has a distribution of directions, to find abrupt changes in radiance contribution. The assumption is that the shift in radiance indicates direct lighting is being blocked by other surfaces i.e. there is an opening in the scene where light is shining through. This technique was useful for adaptively placing samples on regions with high concentration of radiance. However, to gather reliable approximations of the incoming radiance at each surface point would take longer and require more storage than rendering the scene itself.

4.1.1 Spherical Harmonics

[Greger et al., 1998] seemed most reasonable to importance sample the light entering the scene. Unfortunately, each volume requires a large amount of memory to capture arbitrary directions light could span. One way around this according to [Ramamoorthi, 2009] is to use spherical harmonics because only nine parameters, corresponding to the lowest frequency spherical harmonic basis functions, are required to have a nice approximation of irradiance or reflection from a diffuse surface.

Harmonic functions, the solutions to Laplace's equation, are used in various fields [Sloan, 2008]. Spherical harmonics are the solutions when restricted to a sphere; it has been used to model volumetric scattering, light source emission, atmospheric scattering, environmental reflections for microfacet BRDFs without global shadows, non-diffuse off-line light transport simulations, BRDF representations, image relighting, and image-based rendering with controllable lighting.

The spherical harmonics Y_{lm} are functions of the elevation and azimuthal spherical coordinates (θ, ϕ), which could also be written as cartesian components x, y, and z. $l \geq 0$ is the frequency, and there are $2l+1$ basis functions for each value of l with $-l \leq m \leq l$. The Y_{lm} are the analogue on the unit sphere to the Fourier basis functions on the line or circle; it is defined as

$$N_{lm} = \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}$$

$$Y_{lm}(\theta,\phi) = N_{lm} P_{lm} \cos(\theta) e^{im\phi} \tag{4.2}$$

where N_{lm} is a normalization factor, $i = \sqrt{-1}$, and P_{lm} correspond to the associated Legendre polynomials. Figure 4.1 shows a visualization of the spherical harmonic basis functions. Green denotes positive values while red highlights the negative values.

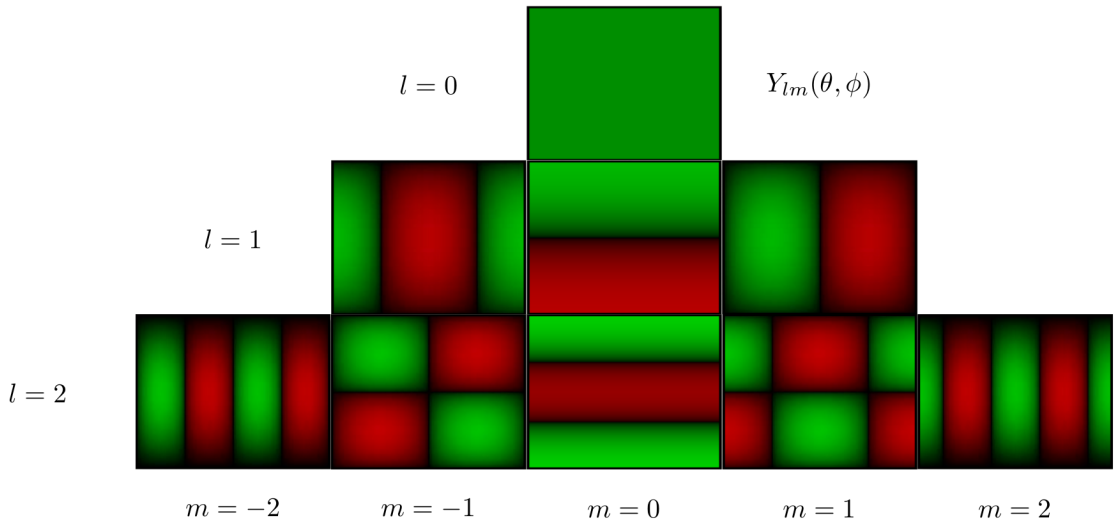


Figure 4.1. Nine spherical harmonic basis functions approximating Lambertian reflection.

Since the spherical harmonics form an orthonormal basis, any function f over the unit sphere can be expanded in terms of Y_{lm} [Green, 2003]. The following calculates a single coefficient for a spherical harmonic band:

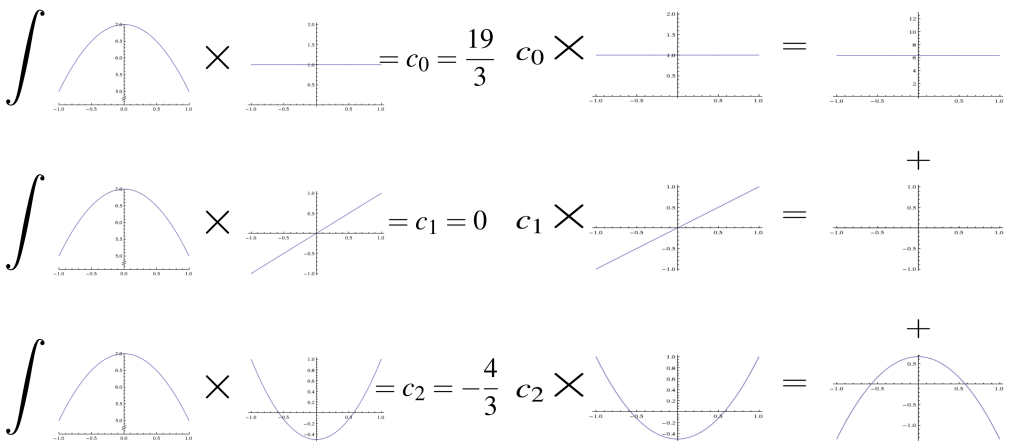
$$c_{lm} = \int_S f(s) Y_{lm} ds$$

where S is the set of samples taken over the domain of f. To reconstruct the approximated function $\tilde{f}$, sum the scaled copies of the corresponding spherical harmonic functions:

$$\tilde{f}(s) = \sum_{l=0}^{n-1} \sum_{m=-l}^{l} c_{lm} Y_{lm}(s) = \sum_{i=0}^{n^2} c_i Y_i(s).$$

Notice that an n^{th} order approximation would require n^2 coefficients; if the summation is over the infinite series of all spherical harmonic coefficients, the true function f can be reconstructed (see Figure 4.2).

Using spherical harmonics to compactly store the transport of radiance, the creation of a large number of irradiance volumes was possible. Unfortunately, the



(a) Projection **(b)** Reconstruction

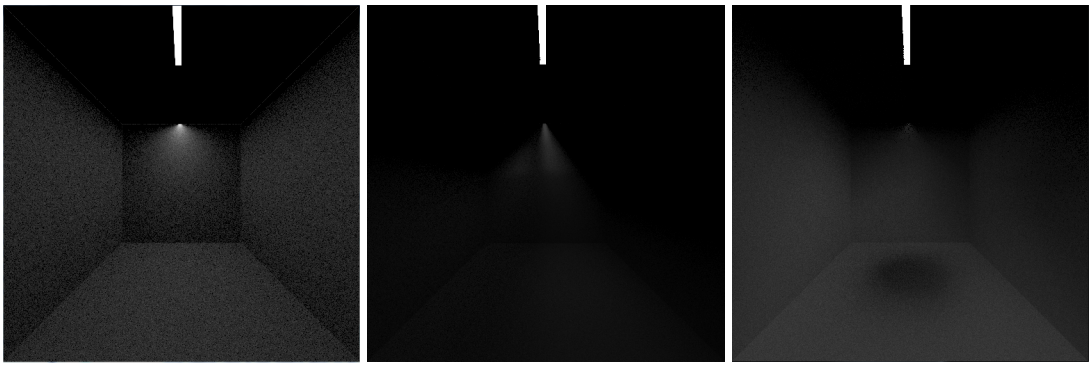
Figure 4.2. An example of projecting $f(x) = -2x^2 + 7$ using the first three Legendre polynomials $1, x$, and $\frac{1}{2}(3x^2 - 1)$.

approximation was far too noisy to be used in an importance sampling scheme; the synthesized image would contain so much white noise such that the scene is no longer recognizable. In an attempt to improve accuracy, spherical harmonics was dropped altogether.

4.1.2 Panel of Sensors

Instead, the entire scene was uniformly subdivided into voxels where a voxel is a volume element representing a value on a regular grid in 3D. As the computed radiance gets propagated back from the light towards the eye, radiance would be deposited at the intersected voxels along the entire path. Should the voxels be traversed by a path that does not hit the light, the radiance is reset to zero. Informal experiments show that the set of voxels containing radiance is sparse, so the memory footprint if considerably less than the irradiance volumes. The voxel traversal algorithm requires only two floating point comparisions and one floating point addition; in addition, multiple ray intersections when an object spans several voxels are eliminated [Amanatides et al., 1987].

Even though this approach was able to identify the voxels located at the light's



(a) Reference **(b)** Low Coverage **(c)** Sampling Bias

Figure 4.3. 4.3a is the scene's reference image consisting only of direct illumination. 4.3b was rendered using an insufficient number of light voxels that failed to cover the light's entrance while 4.3c was synthesized using a large number of tiny voxels.

entrance into the scene, importance sampling the voxels as a light source created some issues. As depicted in Figure 4.3, the scene lacks a lot of direct illumination; this is due to the set of voxels failing to cover the entire opening at the top. Adjusting the granularity of the voxel's size sidesteps this problem at the cost of increased memory consumption and traversal speed.

Even though reducing the size of each voxel such that it fitted the light's entrance perfectly avoided the previous issue, the synthesized image still came out wrong. To sample a voxel v as a light portal from a point x, the sampling needs to be over the solid angle v spans as seen by x. As it turns out, this is quite troublesome to compute without biasing the result. Alternately, each voxel was treated as a light source and the radiance was queried from the closest voxel intersected along the ray path. Even though the unit size of each voxel was several orders of magnitude smaller the entire scene, geometry could be intersecting the voxels that stored radiance. This translates into the slightly dark region at the middle of the floor in Figure 4.3c; it's not completely black because the solid angle v spans is covering a small portion of the light's passageway.

4.1.3 Ray Tracing for Light Portals

To avoid the previously discussed annoyances, voxels were abandoned completely. The takeaway was that the voxels that had radiance deposited into it were located at the light's entrance into the scene. Instead of voxelizing and storing radiance, the new idea is to estimate a best fitting flat geometry that would cover the entire opening where light penetrates into the scene; importance sampling that geometry (a.k.a light portal) would then become trivial as shown in equation (4.1). Using the lessons learned from the previous attempts, points on non-emitting surfaces around the light's passage are temporarily stored to estimate a best fitting light portal. As depicted in Figure 5.1, light is shining through the enclosed region. The points that are used to construct the light portals would ideally cover exactly the perimeter of the region.

Since casting rays is already done as part of the rendering pipeline, the following extra computations are quite efficient. From the origin of the ray that hits the light, a beam is traced along that direction and points are placed on the non-emitting surfaces that the beam intersects. The implicit assumption illustrated in Figure 4.4 is that the surfaces that the beam intersects belong to the actual geometry that is blocking the light out. Notice that three ray paths were traced at an arbitrary point in the scene, but only two beams were fired. The grayed out ray missed the light, so no point in wasting a beam for it. The beam for the ray colored in red did not intersect any geometry, so no points were collected. The beam for the green ray did intersect the tip of the ceiling's edge, so points were randomly placed on the region of intersection.

The current assumption works great when there are no other objects in the scene that would block those beams; this is breaks down immediately when a simple object like a sphere is introduced. Figure 4.5 shows that the beam would intersect with the sphere causing the algorithm to use points on that portion of the sphere as part of the light portal

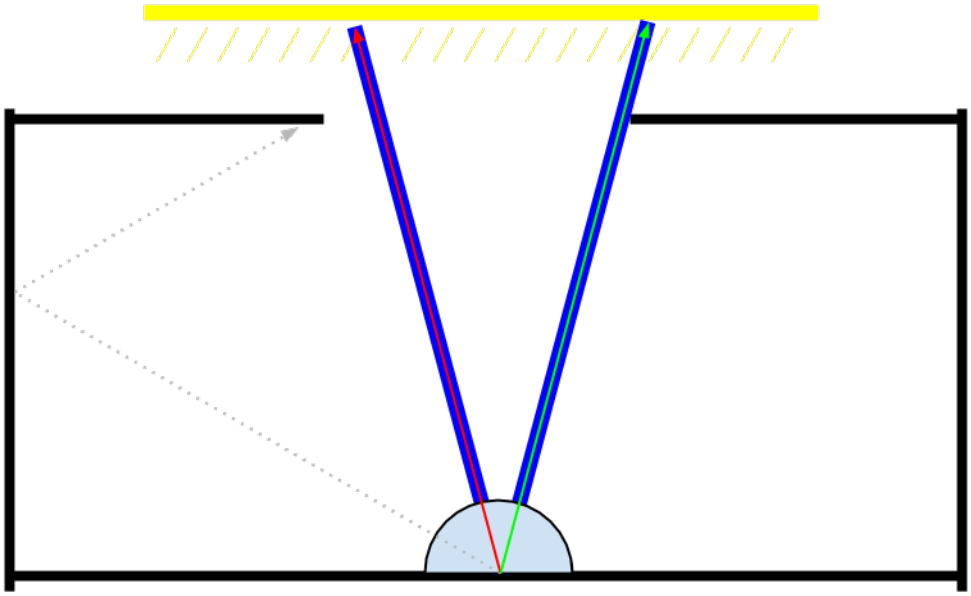


Figure 4.4. Tracing beams to figure out where to place the points.

estimation. The light portal would be positioned approximately somewhere in the red region. Even if all the points on the correct regions were given to the existing light portal estimation, to remove the incorrectly identified region on the sphere would impose more assumptions that varies with different scene configurations. Notice that thus far, only direct illumination is considered; if the light portals are known, indirect illumination is trivial to compute.

4.2 Photon Tracing

Upon reviewing the situations where light portals are used, there is a common theme: the scene is some enclosed space, the objects are located inside that space, and the light is outside of that space. Taking a page out of [Jensen, 2001], instead of sampling from the interior of a scene, photon tracing involves sampling from the light source exterior to the scene i.e. performing ray tracing starting at the lights in the scene. The photons that are deposited when a non-emitting surface intersects the ray paths are used

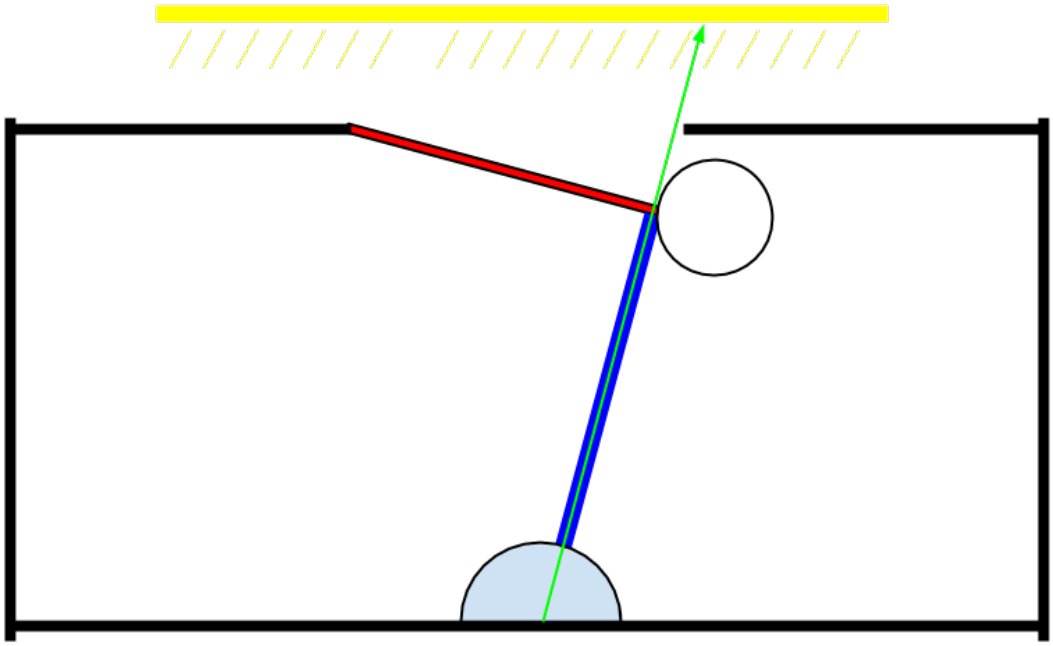


Figure 4.5. Poor estimation of light portal in a non-empty room.

to estimate the light portals. This means the photons are deposited on refractive surfaces (e.g. windows) as well, which intuitively makes sense because the light will refract through the surface thus making it a great candidate for a light portal. Obviously for reflective surfaces (e.g. mirrors), no photons are deposited.

Unlike the previous method of finding light portals, each ray now originates from the light, so a different set of criteria is needed. Note that the following is one approach; more complicated schemes can be used to determine where to place the photons. Recall in the old approach that the beams were traced and the intersected regions were populated with points, this procedure can be viewed as scanning the environment. In a similar manner, the photon tracing step involves sending out beams of light. Clearly, the simplest case is when the beam misses the geometry. When the beam's wavefront hits a surface in its entirety, that surface is ignored because placing photons there does not yield any new information about the location of the light portals.

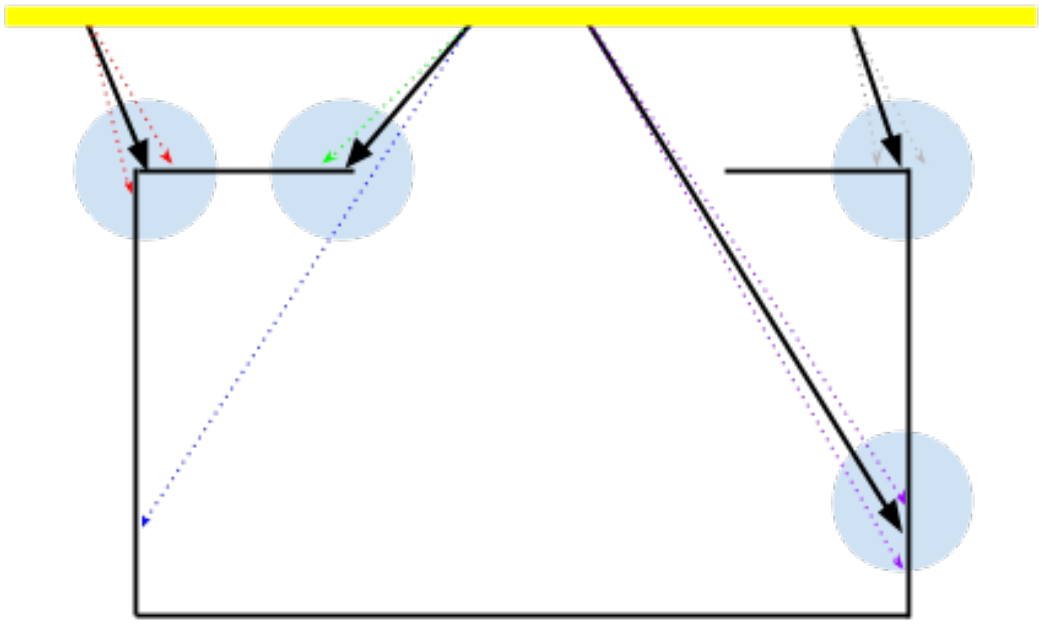


Figure 4.6. Side view of sampling from light. Only the photons deposited by the green rays are qualified to be used in the light portal estimation.

The case that is interesting is when the wavefront is partially occluded. Assuming no geometry is blocking the light from the scene's exterior, the algorithm must handle two scenarios. It must reject all the edges that do not border the light entrance because using those points to estimate the light portal would decrease the sampling efficiency; the light portals that the algorithm creates must span the entire opening otherwise the synthesized image is biased. The remaining situation is when a portion of the wavefront hits a surface and the other portion hits another surface that is further along the same beam. The region on the surface that is closer to the origin of the beam should be included in the estimation.

The implementation of the proposed algorithm requires existing renderers to support casting beams into the scene. In an attempt to reuse the already available ray casting and ray-triangle intersection operations, the following simplifications were made. A point y on the light source is randomly chosen; a random ray is then cast into the

scene from y. Should that yield an intersection with some surface x, a cone-beam, discretized into n rays, with a maximum opening angle of θ is traced in the direction of x. To generate a random direction on unit hemisphere proportional to solid angle with a maximum opening angle of θ along the Z-axis, [Dutré, 2001] derived

$$\phi = 2\pi\mu_1 \qquad \wedge \qquad \theta = \cos(1-\mu_2(1-\cos(\theta)))^{-1}.$$

As depicted in Figure 4.6, four random (black) rays were shot. Once a ray hits some geometry, the algorithm would proceed to cast cone-beams. The red rays at the left will not deposit any photons because the intersection of each of the n rays lie within a sphere whose radius is r. The sphere of rejection's purpose is to prevent deposition of photons in these false regions. The cone-beam consisting of purple rays do not deposit any photons since all of the ray intersections lie on the same plane. One of the gray rays did not intersect any geometry, so the cone-beam it belongs to does not deposit any photons. Only the green rays actually deposit photons because of the previously mentioned reasons and there are depth discontinuity in the cone-beam. Notice that the blue rays do not deposit any photons because those intersections are at a lower elevation with respect to the original ray.

The tuning of θ, n, r are dependent on the scene configuration. Clearly, θ controls how large the search area is; it should be large enough for the cone-beam to detect depth discontinuity. Once an area with depth discontinuity has been detected, it should be sampled as much as possible to increase the sample size of the light portal estimation. The photon tracing step can be stopped at any time to proceed to the next phase.

Algorithm 1. PhotonTrace(θ, n, r)

while not enough points **do**
  Generate a random point y on light sources
  Generate random ray $r(y, \Psi)$
  if $r(y, \Psi)$ intersects scene at x **then**
    for $i \ldots n$ **do**
      Generate cone-beam $r(y, \overrightarrow{yx})$ with a maximum opening angle of θ
      if $r(y, \overrightarrow{yx})$ intersects scene at x' **then**
        if distance$(x, x') > r$ and x' is above the plane$(-\overrightarrow{yx}, x)$ **then**
          Add x' to set of estimation points.
      else
        Stop tracing beam, and remove points added from beam.

4.3 Clustering Photons

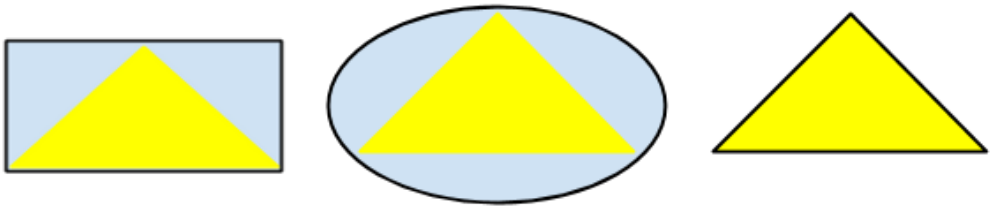
Remember that a bad light portal could make the final image biased or noisy. Along that train of thought, an inefficient light portal would still yield the same unbiased image, but it takes longer to synthesize than without the light portal. Efficiency is defined as

$$\varepsilon[\hat{F}] = (V[\hat{F}] \cdot T[\hat{F}])^{-1}$$

where $\hat{F}$ is the Monte Carlo estimate, V is the variance of the estimate, and T is the time required to estimate F [Veach and Guibas, 1995]. Since correctly fitted light portals yield unbiased estimations,

$$V(\hat{F}) = \mathrm{MSE}(\hat{F}) - \left(\mathrm{Bias}(\hat{F}, F)\right)^2 = \mathrm{MSE}(\hat{F})$$

where MSE is the mean squared error, and F is the true solution. As shown in Figure 4.7, the triangular yellow region represents the light while the blue region with the black solid border is the light portal. Fitting a triangular light portal avoids sampling regions that light is blocked out. Thus to generate efficient light portals, each spatially separated region of photons outlining the openings into the scene should be fitted its own light



(a) Quadrilateral-shaped **(b)** Disc-shaped **(c)** Triangular-shaped

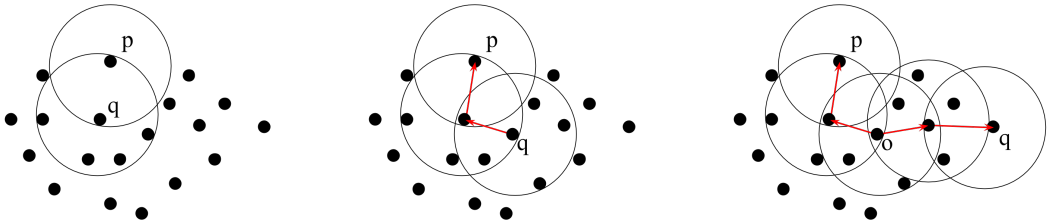
Figure 4.7. Illustrates efficiency of light portal.

portal. Even though the distribution of photons gathered during photon tracing is arbitrary, noisy, and dependent on the scene geometry, given enough samples, the photon clusters become spatially distinct. [Ester et al., 1996] proposed DBSCAN (density-based spatial clustering of applications with noise) to solve this kind of problem. DBSCAN takes two parameters ε and m. It can discover clusters of arbitrary shape. Last but not least, it scales with an overall time complexity of $\mathcal{O}(n \log n)$ where n is the number of point. Notice that this phase is independent of the renderer, and the photons can be stored on disk in some database for querying instead of in memory.

DBSCAN's main idea is that for each point of a cluster, the density in the neighborhood has to exceed some threshold. The shape of the neighborhood is determined by some distance function for two points p and q denoted as $d(p,q)$. The following will formalize these concepts.

Let D be a database of points. There are two kinds of points in a cluster: points on the border of the cluster (border points) and points inside the cluster (core points). The ε-neighborhood of a point p is defined as $N_\varepsilon(p) = \{q \in D \mid d(p,q) \leq \varepsilon\}$. A cluster $C(\varepsilon, m)$, where m is the minimum number of points, is a non-empty subset of D if it satisfies the maximality and connectivity conditions. Suppose $C_1, \ldots, C_k$ are clusters of D; noise is defined as the set of points in D not belonging to any clusters C_i.

The maximality condition states $\forall p,q$, if $p \in C$ and q is density-reachable from p parameterized with (ε, m), then $q \in C$. A point p is density-reachable from a point



(a) Directly Density-Reachable **(b)** Density-Reachable **(c)** Density-Connected

Figure 4.8. Illustration of DBSCAN definitions.

q parameterized with (ε, m) if there is a chain of points $p_1, \ldots, p_n$ with $p_1 = q$ and $p_n = p$ such that p_{i+1} is directly density-reachable from p_i. A point p is directly density-reachable from a point q parameterized with (ε, m) if $p \in N_\varepsilon(q)$ and $|N_\varepsilon(q)| \geq m$ (core point condition) holds. The core point condition is to account for the cardinality of a border point's N_ε being significantly less than the cardinality a core point's N_ε while requiring each cluster to have a minimum of m points.

The connectivity condition states $\forall p$ and q, p is density-connected to q parameterized with (ε, m). A point p is density-connected to a point q parameterized with (ε, m) if there is a point o such that both p and q are density-reachable from o parameterized with (ε, m). This condition covers the relation of border points when they are possibly not density-reachable from each other because the core point condition might not hold for both of them.

Figure 4.8 depicts the previously mentioned definitions. p is a border point while q is a core point in Figures 4.8a and 4.8b; for Figure 4.8c, p and q are border points while o is a core point. p is directly density-reachable from q, but the reverse is not true in Figure 4.8a. In Figure 4.8b, only p is density-reachable from q. Lastly, p and q are density-connected to each other through o.

With Algorithm 2 in hand, the distribution of photons can be clustered into its nearby region, and each cluster would have its own light portal. Notice that regionQuery

is frequently called; using the R*-tree proposed in [Beckmann et al., 1990] takes at most $\mathcal{O}(\log n)$ for a set of n points, which is why DBSCAN only takes $\mathcal{O}(n \log n)$.

Algorithm 2. DBSCAN(D, ε, m)

```
C ← 0
for each unvisited point p ∈ D do
    p.visited ← TRUE
    N_ε(p) ← regionQuery(p, ε)
    if |N_ε(p)| < m then
        p.class ← NOISE
    else
        C ← C + 1
        p.class ← C
        for each point q ∈ N_ε(p) do
            if not q.visited then
                q.visited ← TRUE
                N_ε(q) ← regionQuery(q, ε)
                if |N_ε(q)| ≥ m then
                    N_ε(p) ← N_ε(p) ∪ N_ε(q)
            if q.class ∈ {UNCLASSIFIED, NOISE} then
                q.class ← C
```

4.4 Fitting Light Portals

After the photons have been classified into different clusters, the set of points in each cluster can be fitted to some planar geometry model, which will serve as a light portal. Upon examining the distribution of photons in each cluster, even though DBSCAN tries to remove noisy data points, there are still outliers that degrade the planar geometry estimation.

4.4.1 RANSAC (RANdom SAmple Consensus)

RANSAC is a well-known general parameter estimation approach designed to cope with large proportion of outliers in the input data [Fischler and Bolles, 1981]. It has been studied over the past thirty years and still is a very active area of research.

For the purpose of this thesis, the classic version of RANSAC is satisfactory; exploring other variations of RANSAC may prove to be unfruitful [Choi et al., 2009]. Note that Algorithm 3 has heuristics on picking τ and N, but performance via trial and error works just as good.

Algorithm 3. RANSAC$(\mathscr{D}, \mathscr{M}, \varepsilon, \tau, N)$

1: Select randomly from the entire dataset $\mathscr{D}$ the minimum number of points required to determine model $\mathscr{M}$'s parameters.
2: Solve for the parameters of $\mathscr{M}$.
3: $n \leftarrow$ number of inliers i.e. data points that fit $\mathscr{M}$ with an error tolerance of ε.
4: If $\frac{n}{|D|}$ exceeds some threshold τ, re-estimate the model parameters using all the identified inliers and terminate.
5: Otherwise, repeat RANSAC (maximum of N times).

4.4.2 Minimum Volume Covering Ellipsoid (MVCE)

Using RANSAC to estimate the plane of the light portal is one approach to handling outliers in the data. Should the clustering phase improve in removing outliers, running RANSAC as a pre-processing step becomes moot. The outliers that remain after the previous step will have less impact on the light portal estimation. Notice that RANSAC could be used to compute the plane of the light portal; the only missing pieces of information are the extents of that plane. The naive solution is simply to take the average position and extents over the plane. As demonstrated later in Section 5, this leads to suboptimal performance. Many complex schemes can be used to fit a model to the leftover photons; the one presented in this section was chosen due to its simplicity and computation.

The chosen scheme is to compute the minimum volume covering ellipsoid over the cluster of photons [Kumar and Yildirim, 2005]. Let the photons be denoted as a set of m points in n dimensional space: $\mathscr{S} = \{x_1, x_2, \ldots, x_m\} \in \mathbb{R}^n$ where the affine hull of

the set $\mathscr{S}$ spans $\mathbb{R}^n$. An ellipsoid is defined as

$$\mathscr{E} = \left\{ x \in \mathbb{R}^n | (x-c)^\top E(x-c) \leq 1 \right\}$$

where $c \in \mathbb{R}^n$ is the center of the ellipsoid and $E \in \mathbb{S}^n_{++}$ is the positive definite matrix describing the ellipsoid's eigenvalues and eigenvectors [Boyd and Vandenberghe, 2004]. The volume of $\mathscr{E}$ is defined as

$$\text{Volume}(\mathscr{E}) = V_n \sqrt{\det(E^{-1})}$$

where

$$V_n = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)}$$

is the volume of the unit n-ball with Γ being the gamma function [Hannah, 1996, Abou-Moustafa and Ferrie, 2007]. Since the square root function is monotonic and V_n evaluates to some positive scalar, a natural formulation of MVCE is

$$\begin{aligned} \underset{E,c}{\text{minimize}} \quad & \det(E^{-1}) \\ \text{subject to} \quad & (x_i - c)^\top E(x_i - c) \leq 1, \qquad i = 1, \ldots, m \\ & E = E^\top \succ 0 \end{aligned}$$

where $\succ$ represents generalized strict inequality for matrices and vectors. In order to use this formulation in a convex optimization framework, the representation of the ellipsoid needs to be modified. Since $E \in \mathbb{S}^n_{++}$, there exists a Cholesky decomposition such that

$E = LL^* = LL^\top$ where L^* is the conjugate transpose of L. Let $A = L^\top$ and $b = L^\top c$:

$$\begin{aligned}(x-c)^\top E(x-c) &= x^\top Ex - x^\top Ec - c^\top Ex + c^\top Ec \\ &= x^\top LL^\top x - x^\top LL^\top c - c^\top LL^\top x + c^\top LL^\top c \\ &= x^\top A^\top Ax - x^\top A^\top b - b^\top Ax + b^\top b \\ &= (Ax-b)^\top (Ax-b) \\ &= \|Ax-b\|_2\end{aligned}$$

This change of variable allows the ellipsoid to be defined as

$$\mathscr{E} = \{x \in \mathbb{R}^n | \|Ax-b\|_2 \leq 1\}.$$

Recall that the square root function was removed earlier since it was a monotonic function. Since [Boyd and Vandenberghe, 2004] has shown that the $\det(A^{-1})$ is a log-convex function on $\mathbb{S}^n_{++}$, a similar trick can be performed to turn the optimization problem into

$$\begin{aligned}\underset{A,b}{\text{minimize}} \quad & \log\det(A^{-1}) \\ \text{subject to} \quad & \|Ax_i - b\|_2 \leq 1, \qquad i = 1,\ldots,m \\ & A = A^\top \succ 0.\end{aligned}$$

Although the last formulation fits into a convex optimization framework, it (the primal problem) is difficult to solve efficiently [Boyd and Vandenberghe, 2004]. Forming the Lagrangian (dual problem) using non-negative Lagrange multipliers to relax the primal problem turns out to be much easier to solve for. Let $\mathscr{S}' = \{\pm q_1, \ldots, \pm q_m\}$ where $q_i^\top = \begin{bmatrix} x_i^\top & 1 \end{bmatrix}$ for $i = 1,\ldots,m$. This technique is known as lifting each x_i to the hyperplane $\mathscr{H} = \{(x, x_{n+1}) \in \mathbb{R}^{n+1} | x_{n+1} = 1\}$. Notice that $\mathscr{S}'$ is centered at the origin. In order to recover the solution to the original problem, $\text{MVCE}(\mathscr{S}) = \text{MVCE}(\mathscr{S}') \cap \mathscr{H}$ [Kumar

and Yildirim, 2005]. To formulate the lifted primal problem requires the following manipulations of the original formulation of MVCE:

$$\begin{aligned}(x-c)^\top E(x-c) &= x^\top Ex - x^\top Ec - c^\top Ex + c^\top Ec \\ &= x^\top Ex + x^\top b + b^\top x + d \\ &= \left[\begin{array}{c|c} x^\top E + b^\top & x^\top b + d \end{array}\right] q \\ &= q^\top \left[\begin{array}{cc} E & b \\ b^\top & d \end{array}\right] q \\ &= q^\top M q\end{aligned}$$

where $M \in \mathbb{S}^{n+1}_{++}$ according to Schur's Complement lemma. The lifted primal problem formulation is then

$$\begin{aligned}\underset{M}{\text{minimize}} \quad & \log\det(M^{-1}) \\ \text{subject to} \quad & q_i^\top M q_i \leq 1, \qquad i = 1,\ldots,m \\ & M = M^\top \succ 0.\end{aligned}$$

The Lagrangian of this lifted primal problem is

$$L(M,\lambda) = \log\det(M^{-1}) + \sum_{i=1}^{m} \lambda_i (q_i^\top M q_i - 1)$$

where $\lambda_i \geq 0$ are the Lagrange multipliers or dual variables. The Lagrange dual function

is then

$$\begin{aligned} g(\lambda) &= \inf_M(L(M,\lambda)) \\ &= -\mathbf{1}^\top\lambda + \inf_M(\log\det(M^{-1}) + \sum_{i=1}^{m}\lambda_i q_i^\top M q_i) \\ &= -\mathbf{1}^\top\lambda - f^*(-M) \\ &= -\mathbf{1}^\top\lambda + \log\det(M) + (n+1) \end{aligned}$$

where inf (infimum) can be thought of as the minimum such that λ could result in $-\infty$, and f^* is the conjugate of the objective function.

Let $f : \mathbb{R}^n \to \mathbb{R}$. The conjugate $f^* = \mathbb{R}^n \to \mathbb{R}$ is defined as

$$f^*(y) = \sup_{x\in\mathbf{dom}f} y^\top x - f(x)$$

where $y \in \mathbb{R}^n$ for which the sup (supremum), which can be thought of as the maximum, is finite. For the problem at hand, $f(X) = \log\det X^{-1}$ on $\mathbb{S}^{n+1}_{++}$. The conjugate function is defined as

$$f^*(Y) = \sup_{X\succ 0}(\mathrm{tr}(YX) + \log\det X)$$

since $\mathrm{tr}(YX)$ is the equivalent inner product on $\mathbb{S}^{n+1}_{++}$ and $\det(X^{-1}) = \det(X)^{-1}$ [Petersen and Pedersen, 2006]. Notice that $Y \prec 0$ must be true; otherwise, Y can have an eigenvector v with $\|v\|_2 = 1$ and eigenvalue $\lambda \geq 0$ such that $X = I + tvv^\top$, which is known as the elementary reflector or Householder transformation. Using the definition of trace and Schur's Complement lemma:

$$\begin{aligned} \mathrm{tr}(YX) + \log\det X &= \mathrm{tr}(Y + tYvv^\top) + \log\det(I + tvv^\top) \\ &= \mathrm{tr}\,Y + t(v^\top Y v) + \log(1 + tv^\top I^{-1} v)\det(I) \\ &= \mathrm{tr}\,Y + t\lambda + \log(1+t) \end{aligned}$$

which is unbounded above as $t \to \infty$. Since

$$\nabla_X \mathrm{tr}(YX) + \log\det X = Y + X^{-1} = 0,$$

the maximal value of $X = -Y^{-1}$. Thus

$$\begin{aligned} f^*(Y) &= \sup_{X \succ 0} (\mathrm{tr}(YX) + \log\det X) \\ &= -\mathrm{tr}(YY^{-1}) + \log\det(-Y^{-1}) \\ &= -\log\det(-Y) - (n+1) \end{aligned}$$

with $\mathbf{dom} f^* = -\mathbb{S}^{n+1}_{++}$.

The most important property of the dual function is that it is concave even if the objective function is not convex given Slater's condition holds. [Boyd and Vandenberghe, 2004] has shown that it does hold so the duality gap is zero i.e. solving for the optimal solution to the dual problem is the same as solving for the optimal solution to the primal problem. The Karush-Kuhn-Tucker (KKT) conditions state that the following must hold in order for the solution to be optimal

$$\begin{aligned} \frac{\partial L}{\partial M} = 0 &= \frac{-\det(M^{-1}){M^{-1}}^\top}{\det M^{-1}} + \sum_{i=1}^{m} \lambda_i q_i q_i^\top \\ 0 &= -M^{-1} + Q\Lambda Q^\top \\ M &= (Q\Lambda Q^\top)^{-1} \end{aligned}$$

where $\Lambda = \mathrm{diag}(\lambda)$ and $Q = [q_1, \ldots, q_m] = \left[P^\top \mathbf{1}\right]^\top \in \mathbb{R}^{(n+1)\times m}$ with $P \in \mathbb{R}^{n\times m}$. The

dual problem can be expressed as

$$\begin{aligned} \underset{\lambda}{\text{maximize}} \quad & \log\det((Q\Lambda Q^\top)^{-1}) - \mathbf{1}^\top\lambda + n + 1 \\ \text{subject to} \quad & Q\Lambda Q^\top \succ 0 \\ & \lambda \succeq 0. \end{aligned}$$

Notice that this can be equivalently written as

$$\begin{aligned} \underset{u}{\text{maximize}} \quad & \log\det\Pi(u) \\ \text{subject to} \quad & \mathbf{1}^\top u = 1 \\ & u \succeq 0. \end{aligned}$$

where

$$\Pi(u) = (n+1)\sum_{i=1}^{m} u_i q_i q_i^\top = \sum_{i=1}^{m} \lambda_i q_i q_i^\top$$

with

$$\lambda_i = (n+1)u_i.$$

The only reason to rewrite the dual problem into a neater form is to satisfy the input problem specification of generic convex optimization solvers. Using Schur's Complement

lemma to take the inverse of M yields

$$
\begin{aligned}
M = \Pi(u)^{-1} &= \left((n+1) \sum_{i=1}^{m} u_i q_i q_i^\top \right)^{-1} \\
&= \frac{1}{n+1} \left(\begin{bmatrix} P \\ \mathbf{1}^\top \end{bmatrix} U \begin{bmatrix} P^\top & \mathbf{1} \end{bmatrix} \right)^{-1} \\
&= \frac{1}{n+1} \left(\begin{bmatrix} PUP^\top & Pu \\ (Pu)^\top & \mathbf{1}^\top u \end{bmatrix} \right)^{-1} \\
&= \frac{1}{n+1} \left(\begin{bmatrix} I & Pu \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} PUP^\top - Pu(Pu)^\top & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} I & \mathbf{0} \\ (Pu)^\top & 1 \end{bmatrix} \right)^{-1} \\
&= \frac{1}{n+1} \begin{bmatrix} I & \mathbf{0} \\ -(Pu)^\top & 1 \end{bmatrix} \begin{bmatrix} (PUP^\top - Pu(Pu)^\top)^{-1} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} I & -Pu \\ \mathbf{0}^\top & 1 \end{bmatrix} \\
&= \frac{1}{n+1} \begin{bmatrix} I & \mathbf{0} \\ -(Pu)^\top & 1 \end{bmatrix} \begin{bmatrix} Z & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} I & -Pu \\ \mathbf{0}^\top & 1 \end{bmatrix}
\end{aligned}
$$

where $U = \mathrm{diag}(u)$ and $Z = (PUP^\top - Pu(Pu)^\top)^{-1}$. The quadratic form of M can now be written as

$$
\begin{aligned}
q^\top M q &= \begin{bmatrix} x^\top & 1 \end{bmatrix} \Pi(u)^{-1} \begin{bmatrix} x \\ 1 \end{bmatrix} \\
&= \frac{1}{n+1} \begin{bmatrix} x^\top & 1 \end{bmatrix} \begin{bmatrix} I & \mathbf{0} \\ -(Pu)^\top & 1 \end{bmatrix} \begin{bmatrix} Z & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} I & -Pu \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} \\
&= \frac{1}{n+1} \begin{bmatrix} x^\top - (Pu)^\top & 1 \end{bmatrix} \begin{bmatrix} Z & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} x - Pu \\ 1 \end{bmatrix} \\
&= \frac{1}{n+1} (x^\top Z x - x^\top Z P u - (Pu)^\top Z x + (Pu)^\top Z P u + 1).
\end{aligned}
$$

Consequently, the ellipsoid defined in terms of the dual variables is

$$\mathscr{E} = \left\{ x \in \mathbb{R}^n | (x-c)^\top E(x-c) \leq 1 \right\}$$

where

$$c = Pu \quad \wedge \quad E = \frac{Z}{n} = \frac{1}{n}(PUP^\top - cc^\top)^{-1}$$

since

$$\begin{aligned} 1 &\geq \frac{1}{n+1}(x^\top Zx - x^\top ZPu - (Pu)^\top Zx + (Pu)^\top ZPu + 1) \\ n+1 &\geq x^\top Zx - x^\top ZPu - (Pu)^\top Zx + (Pu)^\top ZPu + 1 \\ 1 &\geq \frac{1}{n}(x^\top Zx - x^\top ZPu - (Pu)^\top Zx + (Pu)^\top ZPu) \\ &\Rightarrow x^\top Ex - x^\top Ec - c^\top Ex + c^\top Ec. \end{aligned}$$

Since E is a symmetric matrix, its singular values equal its eigenvalues. To compute the semi-axes, simply perform singular value decomposition. The magnitude of the semi-axes are the inverse square roots of the singular values while left singular vectors correspond to the extents of the ellipsoid.

Khachiyan First-Order Algorithm

Now that the dual problem have been formulized, it can be fed into a generic quadratic programming solver. This section will present a simple solution that makes use of the convexity of the objective function. The key idea is to iteratively compute the gradient of the function and perform line search along the optimal direction [Sun and Freund, 2004].

[Frank and Wolfe, 1956] introduced the Conditional-Gradient method for solving non-linear optimization problems. Recall that $f(u) = \log\det\Pi(u)$ is the objective

function to optimize where $\mathbf{1}^\top u = 1$ and $u \succeq 0$; the linearization of $f(u)$ at $u = \overline{u}$ is

$$f(\overline{u}) + \nabla f(\overline{u})(u - \overline{u}),$$

which is the first order Taylor expansion of f at $\overline{u}$ with

$$\begin{aligned}
\nabla f(u)_i &= \frac{\partial \log\det \Pi(u)}{\partial u_i} \\
&= \operatorname{tr}\left(\Pi(u)^{-1}\frac{\partial \Pi(u)}{\partial u_i}\right) \\
&= \operatorname{tr}\left(\frac{(\sum_{j=1}^m u_j q_j q_j^\top)^{-1}}{n+1}(n+1)q_i q_i^\top\right) \\
&= q_i^\top (QUQ^\top)^{-1} q_i
\end{aligned}$$

for $i = 1, \ldots, m$. The linear programming problem can be formulated as

$$\begin{aligned}
\underset{u}{\text{maximize}} \quad & \nabla f(\overline{u})^\top u \\
\text{subject to} \quad & \mathbf{1}^\top u = 1 \\
& u \succeq 0.
\end{aligned}$$

Since the feasible region of this linear programming problem is the unit simplex in $\mathbb{R}^m$, the optimal solution u^* is the unit vector $e_j \in \mathbb{R}^m$ where

$$j = \underset{i}{\operatorname{argmax}} \nabla f(\overline{u})_i.$$

Since $\mathbb{R}^m$ is a convex set, the line segment joining $\overline{u}$ and u^* is also in the set; hence, performing a line search over f can be formulated as

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \log\det\Pi((1-\alpha)\overline{u}+\alpha e_j) \\ \text{subject to} \quad & 0 \leq \alpha \leq 1 \end{aligned}$$

where [Khachiyan and Todd, 1993] have brilliantly observed that the closed-form solution to this line search problem is

$$\alpha^* = \frac{\nabla f(\overline{u})_j - (n+1)}{(n+1)(\nabla f(\overline{u})_j - 1)}.$$

It should be noted that computing $\nabla f(\overline{u})$ is sometimes not possible using the inverse matrix operation due to floating point issues. In that case, the Moore-Penrose pseudoinverse can be used. This check can be done efficiently by computing the condition number of the matrix to be inverted.

Algorithm 4. ConditionalGradient(P, ε)

$Q \leftarrow \left[P^\top \mathbf{1}\right]^\top$
$\overline{u} \leftarrow \frac{1}{m}\mathbf{1}$
$UB = \mathbf{1}$
$LB = \mathbf{0}$
while $\|UB - LB\|_2 > \varepsilon$ **do**
  $LB = \overline{u}$
  Compute $\nabla f(\overline{u})_i = q_i^\top (QUQ^\top)^{-1} q_i$ for $i = 1, \ldots, m$
  $j \leftarrow \underset{i}{\text{argmax}} \nabla f(\overline{u})_i$
  $\alpha \leftarrow \frac{\nabla f(\overline{u})_j - (n+1)}{(n+1)(\nabla f(\overline{u})_j - 1)}$
  $\overline{u} \leftarrow (1-\alpha)\overline{u} + \alpha e_j$
  $UB = \overline{u}$
$c \leftarrow Pu$
$E \leftarrow (PUP^\top - cc^\top)^{-1}/n$

Chapter 5

Results

The experiments carried out in this section define surface material to have either perfectly diffuse or specular materials. The single light source in the scene is an area diffuse emitter. Triangular and elliptical light portals were sampled uniformly using techniques described in [Dutré, 2001, Wang et al., 2004].

This framework's set of input parameters are denoted as

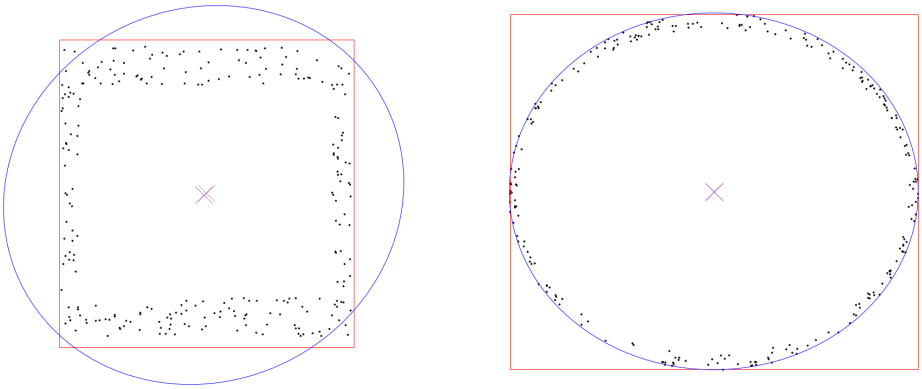
$$\text{PT}(\theta, n, r), \text{DBSCAN}(\varepsilon, m), \text{RANSAC}(\mathscr{M}, \varepsilon, \tau, N), \text{MVCE}(\varepsilon).$$

Depending on the scene geometry and materials, the input parameters may require tuning to improve the estimation of the light portals.

5.1 Estimation of Light Portals

The blatantly obvious question to ask is why MVCE was selected? As mentioned earlier, a finite plane, a triangle mesh, an ellipse, or any 2D polygon can be used to represent the light portals. The following will demonstrate that MVCE is a fast robust method compared to the one of the alternative algorithms.

As illustrated in Figure 5.1, when the data points lie exactly on a plane, using PCA to estimate the square light portal bests MVCE [Shlens, 2005, Gottschalk, 2000].

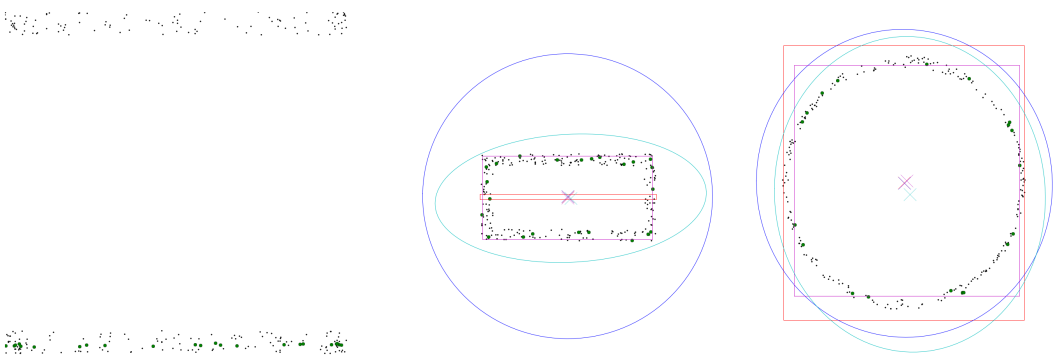


(a) Points on the edge of a plane. **(b)** Points on the edge of an ellipse.

Figure 5.1. Synthetic data points generated to lie exactly on a plane. PCA yielded the square while MVCE produced the ellipse.

Likewise, MVCE is the most appropriate for estimating a light portal in the shape of an ellipse. The PCA method computed the orientation of the light portal, took the average of all the points to be the center, and assigned the largest extents possible along its orientation.

Unfortunately, the cleaned up data after running DBSCAN is not as ideal. DBSCAN's results would contain a mixture of points lying on slightly different planes (see Figure 5.2). Without RANSAC, the PCA method completely failed in estimating the points outlining the shape of a rectangle. With RANSAC, the PCA method succeeded for that case, but its estimation of the points outlining the shape of an ellipse became too small. MVCE on the other hand succeeded in covering the entire set of points regardless of whether RANSAC was used; in the case where RANSAC was used, the estimation improved. Thus the following experiments will use MVCE with RANSAC as the preferred method to estimate the light portals.



(a) XZ-View. **(b)** XY-View. **(c)** XY-View.

Figure 5.2. Noisy data points that lies on two planes. Half of the data points lie on a plane below the other half. The green dots are the inliers selected by RANSAC; the cyan ellipses and magentas square used the inliers for the estimation.

5.2 Efficiency of Light Portals

At the end of the simulation, the only issues that matter are how many samples were used, how long the simulation took, and how good was the estimator. The following experiments will evaluate the efficiency of the estimated light portals with respect to manually placed light portals using two simple scenes.

The first scene is a 1.0m^3 box whose center is positioned at $(0,0,0)$ with a $(0.25^2)\text{m}^2$ square hole in the center of the ceiling. The walls of the cube are as thin as a plane and their surface is perfectly diffuse with $p_d = 0.75$. A $(0.95^2)\text{m}^2$ square diffuse emitter is positioned 0.05m centered above the box radiating with a power of 25W. The camera is positioned at $(0,0,0.5)$ in a right-handed coordinate system whose aspect ratio is 1. Each pixel in the reference 512×512 image was randomly sampled 2^{17} times.

The other scene is essentially the same; instead of a single hole in the center of the ceiling, there are four $(0.125^2)\text{m}^2$ square holes each with one of its corners positioned 0.125m horizontally and vertically away from a corner of the ceiling.

The configurations of the framework for both scenes are as follow:

$$\text{PT}(\theta = 3^\circ, n = 32, r = 1.2\text{m})$$

$$\text{DBSCAN}(\varepsilon = 0.125\text{m}, m = 16)$$

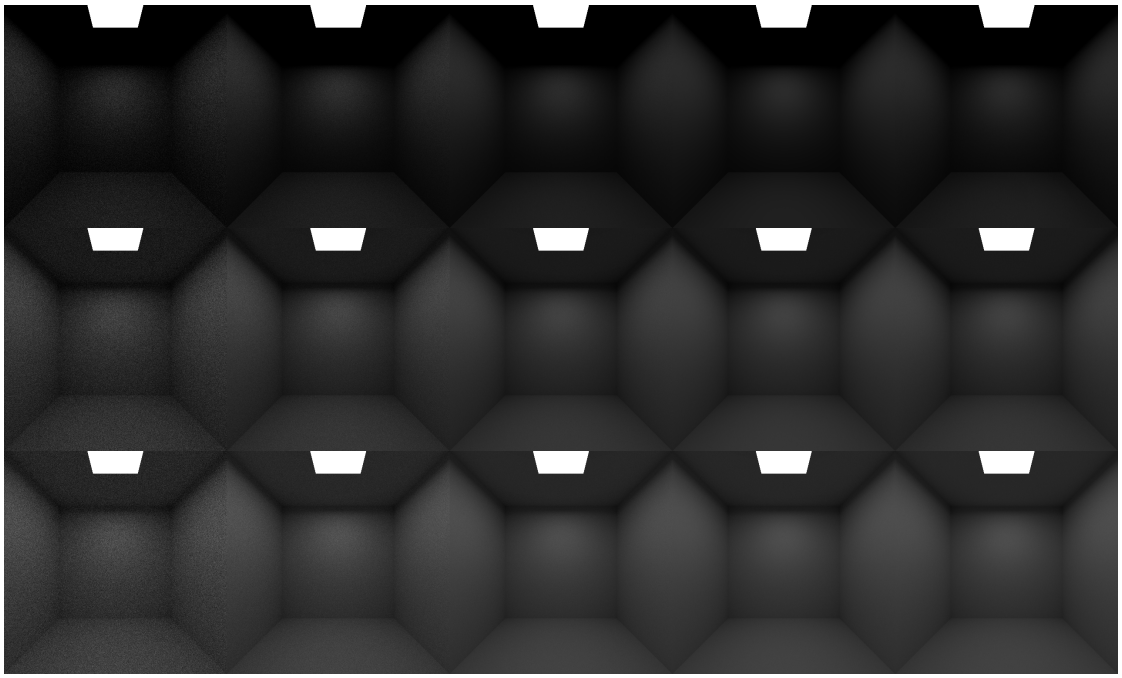
$$\text{RANSAC}(\mathscr{M} = \text{Plane}, \varepsilon = 0.07, \tau = 0.07, N = 1024)$$

$$\text{MVCE}(\varepsilon = 0.001)$$

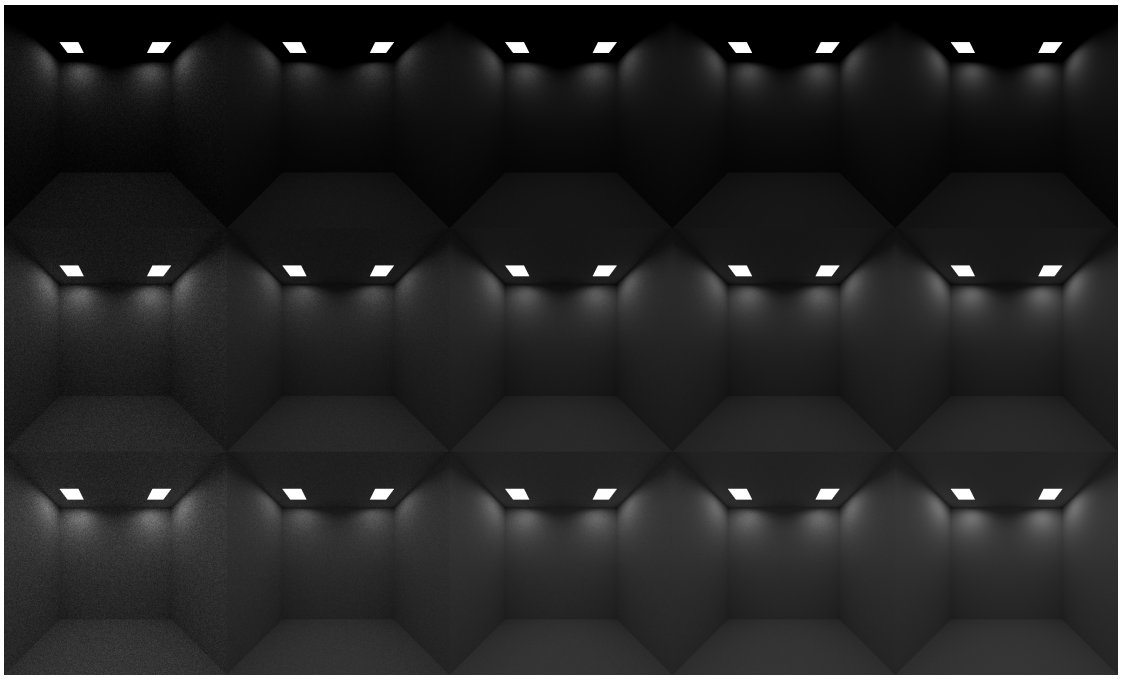
Since the multiple holes in the second scene are a lot smaller, 8192 photons were gathered for the light portal estimation. The first scene's hole is a lot bigger, so only 2048 photons were needed to create an efficient light portal. The methods used in the following comparisons are based off of Path Tracing specifically only hemispherical formulation (HF), MIS between hemispherical formulation and area formulation (AF_HF), MIS between hemispherical formulation and estimated light portal (eLP_HF), and MIS between hemispherical formulation and manually placed light portals (mLP_HF).

Figure 5.4 visually highlights the regions that deviate from the reference image using the root mean square error (RMSE). Notice how the noise becomes more apparent as the path length increases. This means all the techniques become less effective as more indirect illumination is added i.e. they all require more samples. Figure 5.5 and Table 5.1 quantitatively depicts this behavior.

There are two reasons why the estimated light portals are a little more than half the efficiency of manually placed light portals. The manually placed light portal consists of two triangles whereas the estimated light portal consists of a single ellipse. Randomly sampling the surface of an ellipse is more costly than the combined efforts of randomly selecting a triangle and sampling its surface. The major detrimental contributor is the overfitting of the light portal as shown in Figure 5.6. The light portals' regions where the light is blocked waste valuable samples and computation time.

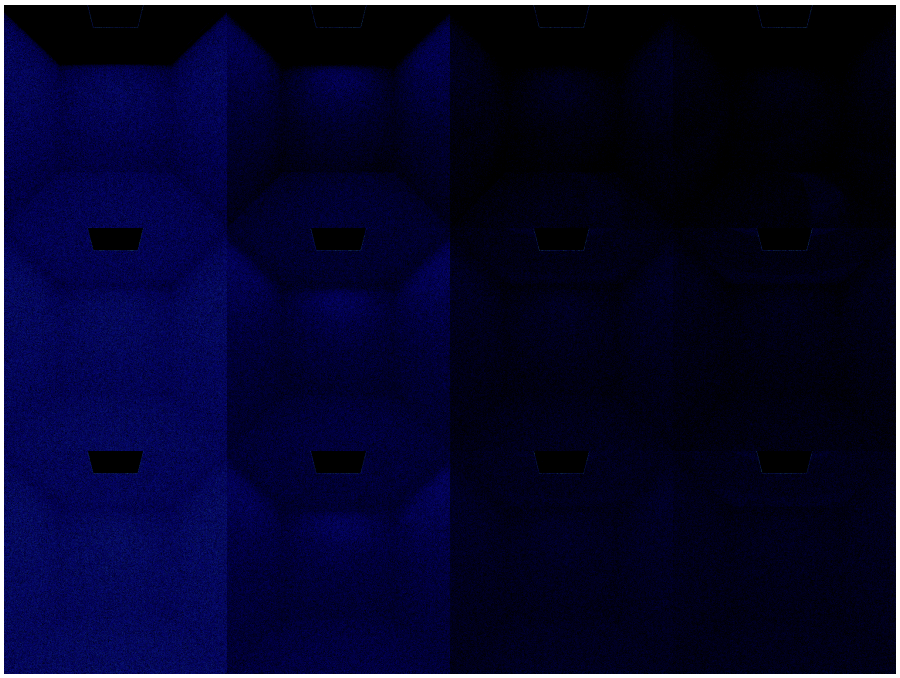


(a) Single Hole

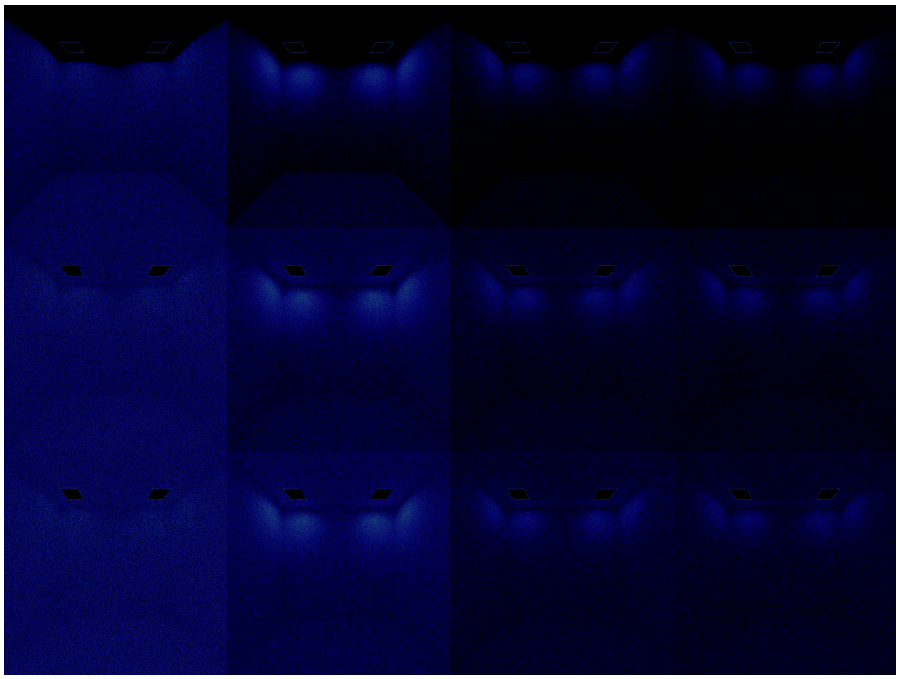


(b) Multiple Holes

Figure 5.3. Comparison of noise between the different techniques. The images in each row were rendered using a maximum path length of 2, 4, and 6 respectively. The columns representing the different methods are ordered as follows: HF, AF_HF, eLP_HF, mLP_HF, and reference image.

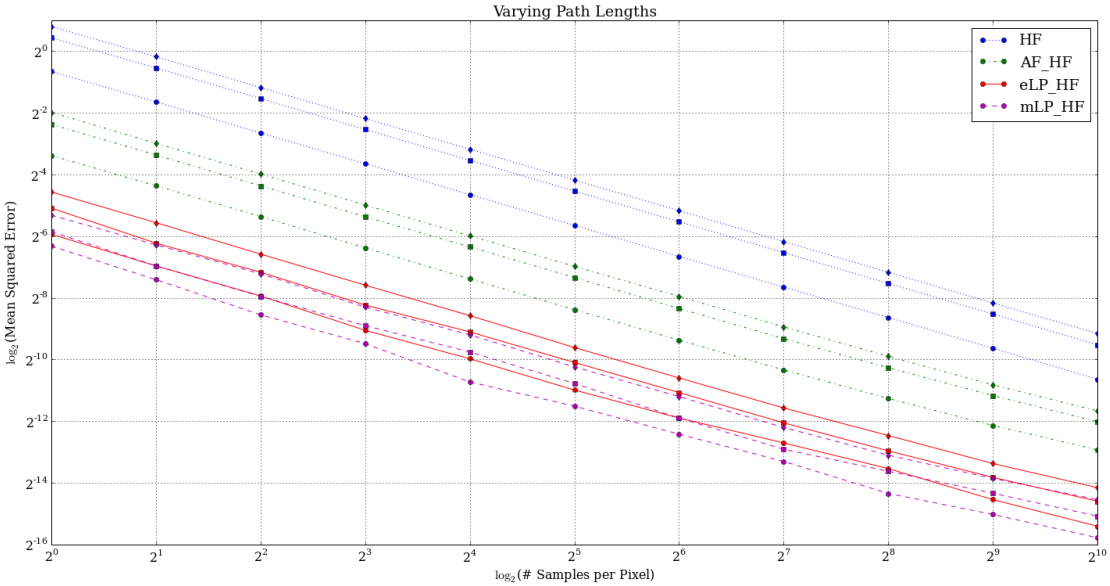


(a) Single Hole

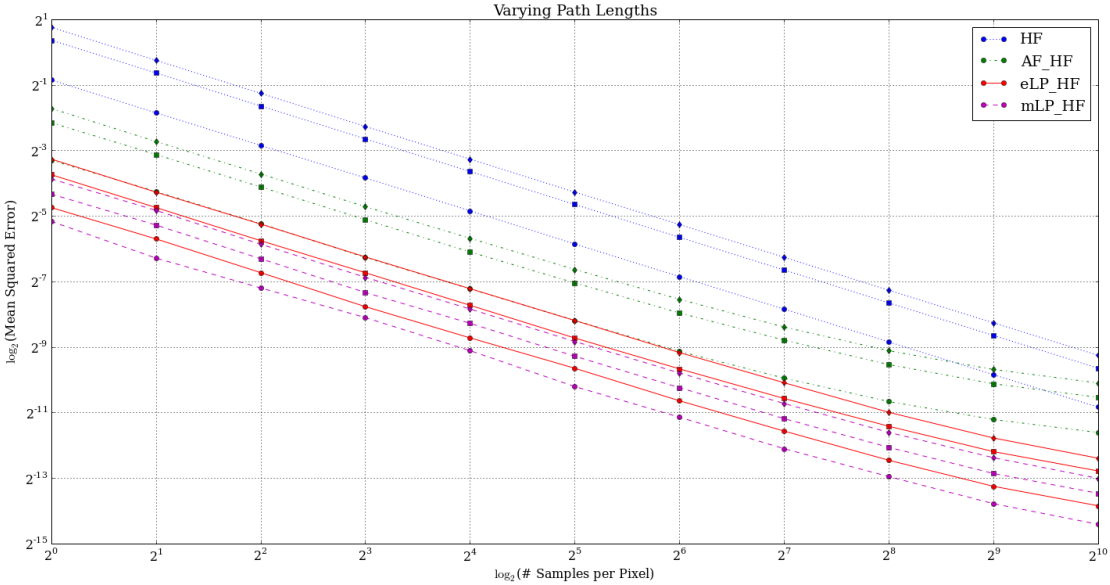


(b) Multiple Holes

Figure 5.4. RMSE of the synthetic images and the reference image ordered in the same manner as Figure 5.3. The dark regions indicate accurate estimations. Visually, the difference in quality between estimated light portals and manual light portals are barely noticeable.

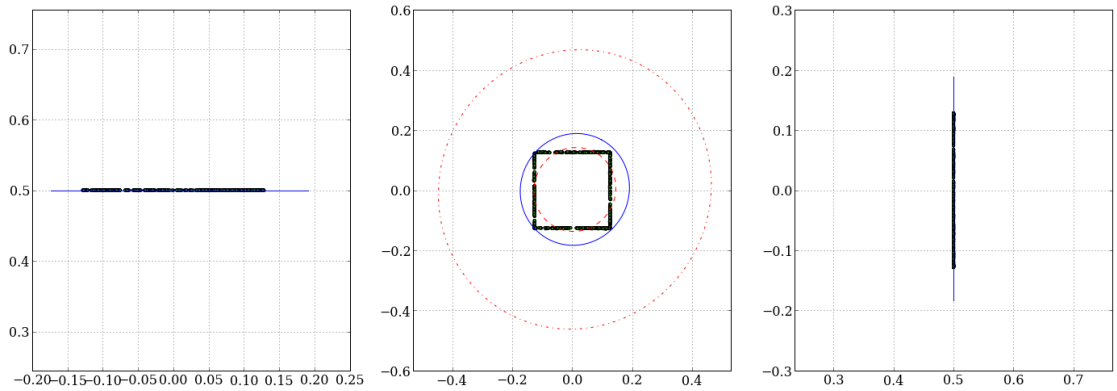


(a) Single Hole

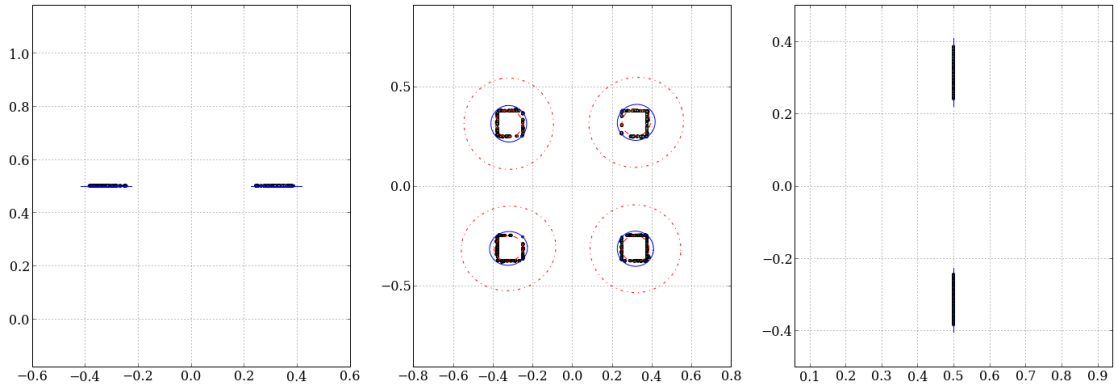


(b) Multiple Holes

Figure 5.5. Accuracy of each method in different scenes. The lines populated with dots, squares, and diamonds respectively represents path lengths of 2, 4, and 6.



(a) Single Hole



(b) Multiple Holes

Figure 5.6. XY, XZ, and YZ view of the light portal and the photons used for the light portal estimation. The blue ellipses are the estimated light portals used in the simulations; they are approximately 1.7 times the area of the corresponding hole in each scene. The inner red ellipse denotes the smallest light portal possible before LP_HF's efficiency is degraded to AF_HF's; likewise, the outer red ellipse indicates the largest light portal possible. The inner and outer red ellipses are respectively 0.9 and 10.5 times the area of the corresponding hole in each scene. These bounds are specific to the scenes and vary depending on AF_HF's performance.

Table 5.1. Efficiency of each method for different scenes. The time (seconds) represents how long the method took to render a 512×512 image using one sample per pixel. Notice how each technique's efficiency degrades exponentially as path length increases.

	Single Hole		Multiple Holes	
Path Length	Time	Efficiency	Time	Efficiency
HF (2)	0.258	6.205	0.253	7.092
AF_HF (2)	0.402	24.955	0.415	19.799
eLP_HF (2)	0.406	140.508	0.428	56.04
mLP_HF (2)	0.389	219.44	0.378	88.194
HF (4)	0.356	2.024	0.365	2.157
AF_HF (4)	0.686	7.248	0.697	5.152
eLP_HF (4)	0.713	46.928	0.746	16.024
mLP_HF (4)	0.647	85.455	0.65	27.773
HF (6)	0.41	1.363	0.421	1.429
AF_HF (6)	0.845	4.526	0.862	3.152
eLP_HF (6)	0.879	25.988	0.935	9.335
mLP_HF (6)	0.756	46.946	0.786	16.717

The two scenes essentially exhibit the same behavior. The significant differences are that each method's efficiency (excluding HF's) is lower in the second scene compared to the first scene despite similar computation times, and each method's MSE curve (excluding HF's) is not as straight as HF's curve. HF's slight improvement in efficiency is not statistically significant and tends to fluctuate as different pseudo-random number generator seeds are used.

The MSE curve (especially AF_HF's) is plateauing because MIS' balance heuristic is not an optimal sampling strategy when one of the sampling techniques (e.g. HF, AF, LP) is an extremely good match for the integrand. [Veach and Guibas, 1995] discussed this issue as a set of low-variance problems and proposed several variations of the balance heuristic one of which is the power heuristic. The idea is to make large weights closer to one and small weights closer to zero via

$$w_j = \frac{p_j^\beta}{\sum_{k=1}^{n} p_k^\beta}$$

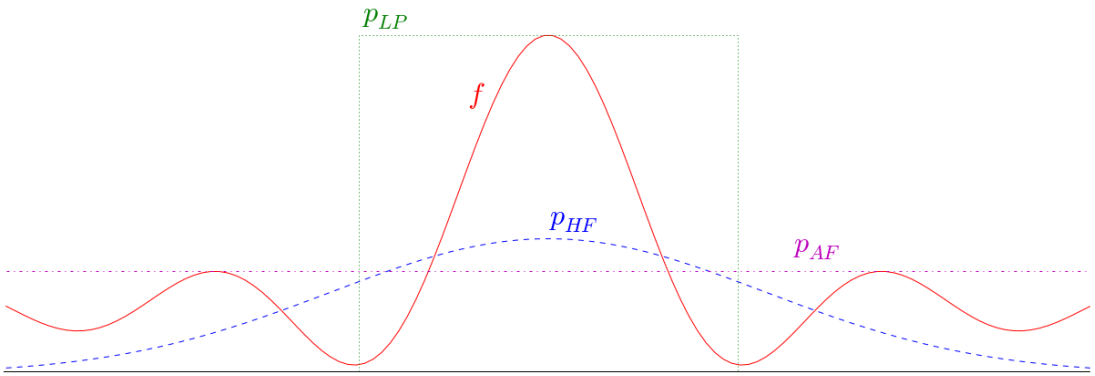


Figure 5.7. Intuition behind using MIS power heuristic to handle low-variance problems.

where $\beta = 2$ have been found to be a reasonable value.

The intuition behind this is illustrated in Figure 5.7. Imagine f as the rendering equation and p denotes the PDF of the technique used. The single large peak can be considered as direct lighting and the rest are indirect lighting, which is analogous to the first scene; the second scene could be thought of as having four thin peaks instead of just one wide one. Notice how p_{HF} and p_{AF} cover the entire function f, this illustrates the idea that either formulation will solve the rendering equation without adding bias.

Combining HF and AF using MIS power heuristic with $\beta \leq 1$ yielded an improvement over just using HF because p_{HF} is putting less samples in the indirect lighting regions whereas AF is covering them uniformly. Recall that p_{AF} must be converted into the same space as p_{HF}, this conversion will evaluate p_{AF} to have a larger value than p_{HF}. Making $\beta > 1$ means putting more emphasis on p_{AF} in turn reducing the contribution from p_{HF} at the large peak thus decreasing the overall accuracy.

The previously discussed behavior is the opposite when combining HF and LP using MIS power heuristic. p_{LP} uniformly samples f in the direct lighting regions and is zero everywhere else. Since p_{LP} matches f in scale compared to p_{HF}, clearly emphasizing p_{LP} (after converting into the same space as p_{HF}) will yield a better estimation.

Unfortunately, making $\beta \gg 2$ yielded a negative effect on the total accuracy because p_{LP} matches f around the peak while p_{HF} matches f around the base of the peak. This reasoning also explains why the power heuristic with respect to light portals is more effective for the second scene (four small holes) compared to the first scene (one large hole) as illustrated in Figure 5.8.

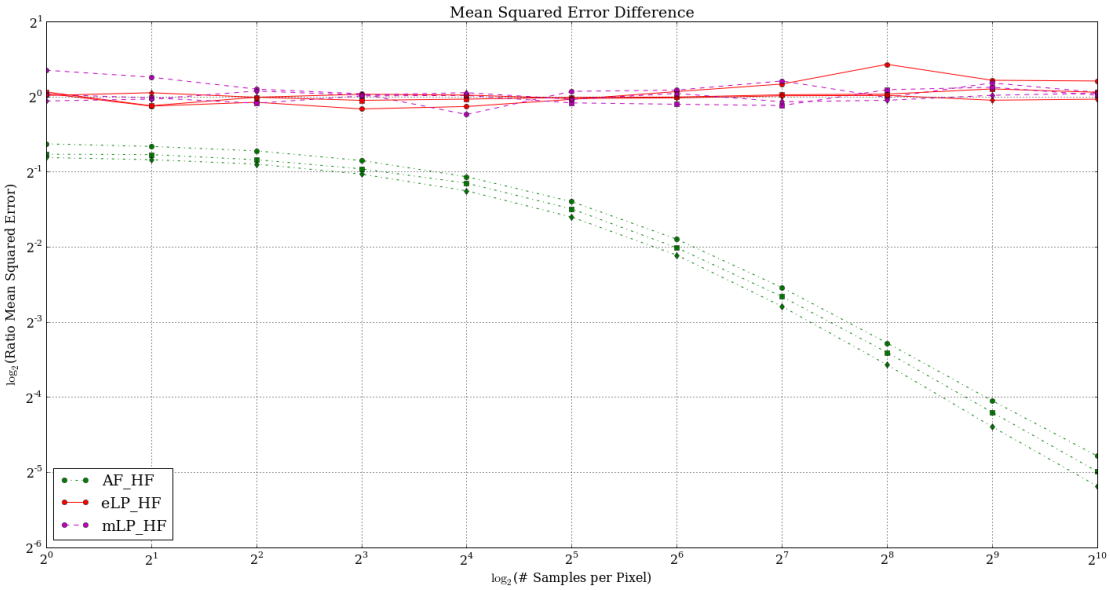
5.3 Limitations of Light Portals

The following experiment will elaborate on the current limitations of this automated approach. The scene consists of a 3D model of the Sibenik Cathedral and an area light source shining only on one side of the cathedral. The Sibenik model has only two materials: diffuse for all surfaces except windows, and the windows behave like glass. RANSAC and MVCE parameters remain the same as the previous experiments.

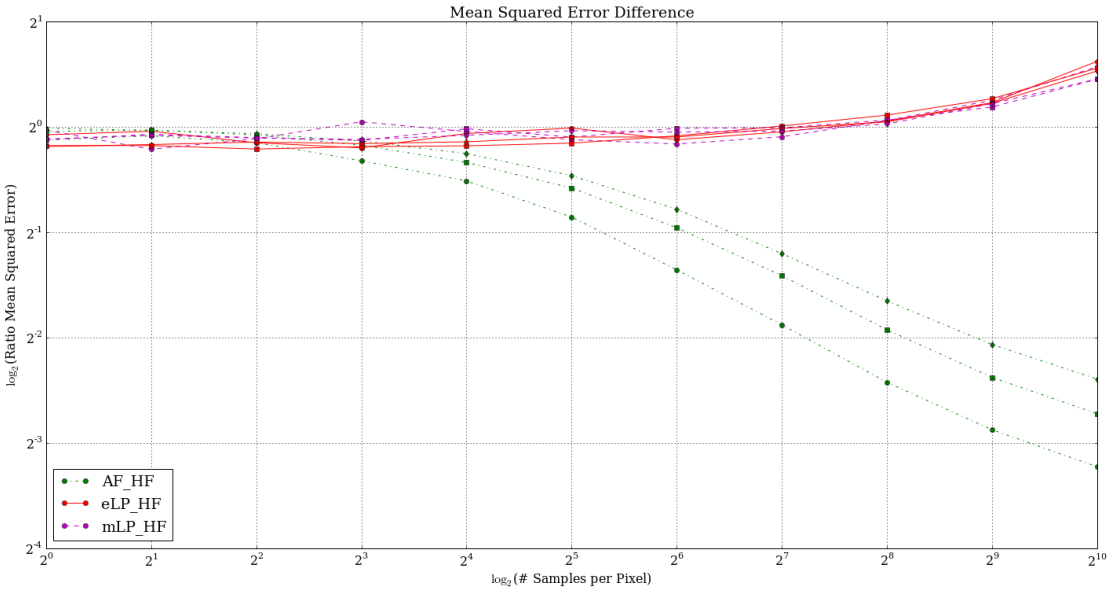
Given an arbitrary model such as the Sibenik Cathedral, the current solution requires the user to tune PT's parameters. Luckily, since the photon tracing step performs ray tracing with a fixed path length of one, this can be done in an interactive manner. The current unoptimized implementation performs this operation on the order of seconds.

There are three scenarios to consider for PT: Sibenik with holes (i.e. windows removed), Sibenik using only depth discontinuity, and Sibenik using both depth discontinuity and material properties. Figure 5.9 demonstrates that feeding PT bad parameters will result in a generic point cloud of the scene regardless of the scenario.

Figure 5.10 tries to show how the user would go about tuning PT. Starting from the bad parameters in Figure 5.9, θ is reduced until there is apparent structure in the scene preferably the outline of the holes. Recall that increasing or decreasing θ is dependent on the scene; θ should be scaled such that the cone-beam it simulates is limited to the perimeter of the hole as much as possible. Notice that there are unwanted edges in the point cloud due to false detection of holes. The next step is to scale r to remove these



(a) Single Hole



(b) Multiple Holes

Figure 5.8. Illustrates the accuracy differences using the ratio of MIS' balance heuristic to power heuristic. The legend is the same as in Figure 5.5 with HF excluded since HF does not use MIS. The technique using light portals received a nice enough improvement to straighten out the MSE curve. Unfortunately, AF_HF's accuracy worsened, which means the power heuristic with $\beta = 2$ is not the optimal strategy for combining AF and HF in these two scenes; AF_HF did exhibit a similar improvement when $\beta < 1$.

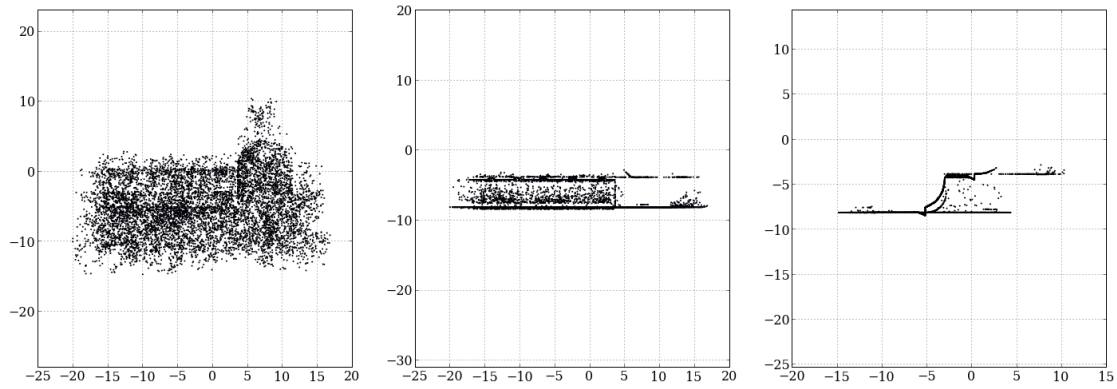
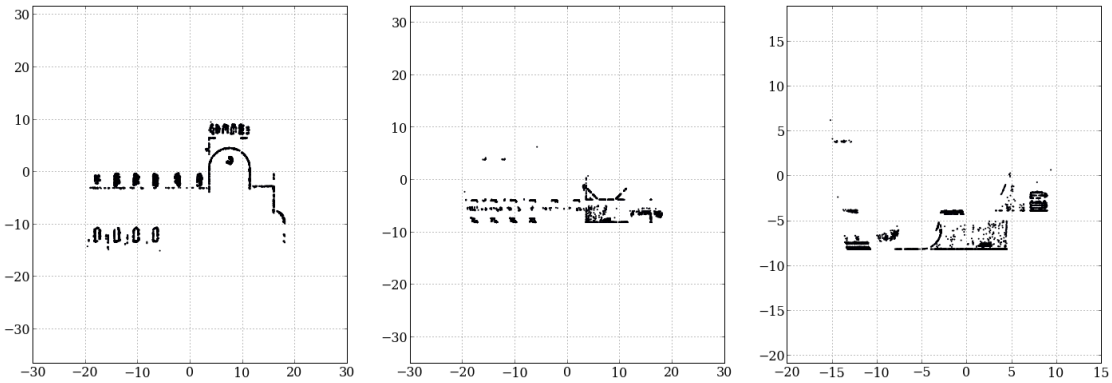


Figure 5.9. Tuning photon tracing for Sibenik using arbitrary initial parameters such as $(\theta = 3^\circ, n = 32, r = 1.2\text{m})$.

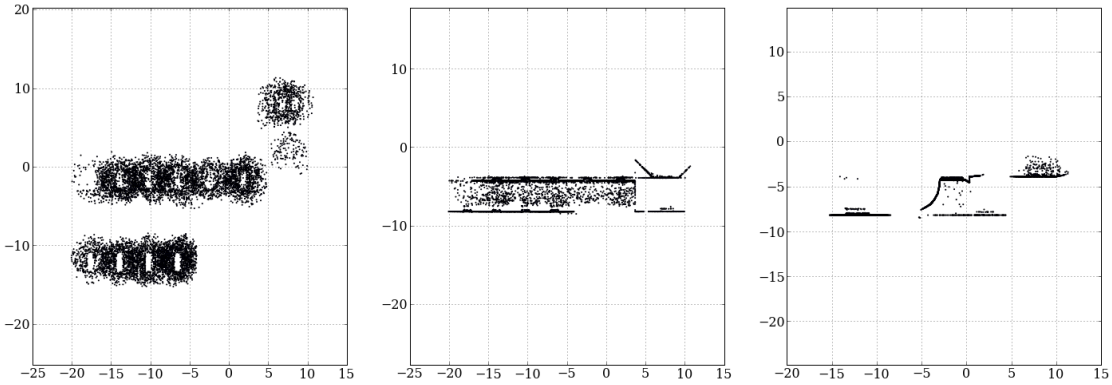
false regions. Since the depth discontinuity between the holes and its surrounding is large, increasing r should detect only the regions of interest. Unsuccessfully, if the holes in the scene lack strong depth discontinuity, the result of PT will be noise as portrayed in Figure 5.11 regardless of what parameters are tried.

One enhancement to PT mentioned earlier is to use the material properties of each surface. Since the only way light can enter the Sibenik's interior is through the glass windows, PT should only deposit photons on specular surfaces and non-specular regions around the window (see Figure 5.12). This is much more efficient than sampling all windows because only half of the windows are in the light path. The points are then clustered via DBSCAN with $\varepsilon = 0.5$m and nineteen light portals are estimated using RANSAC MVCE. $\varepsilon = 0.5$ happens to work great because the windows are roughly 1m in width; making it any larger would cause proximate light portals to be merged together.

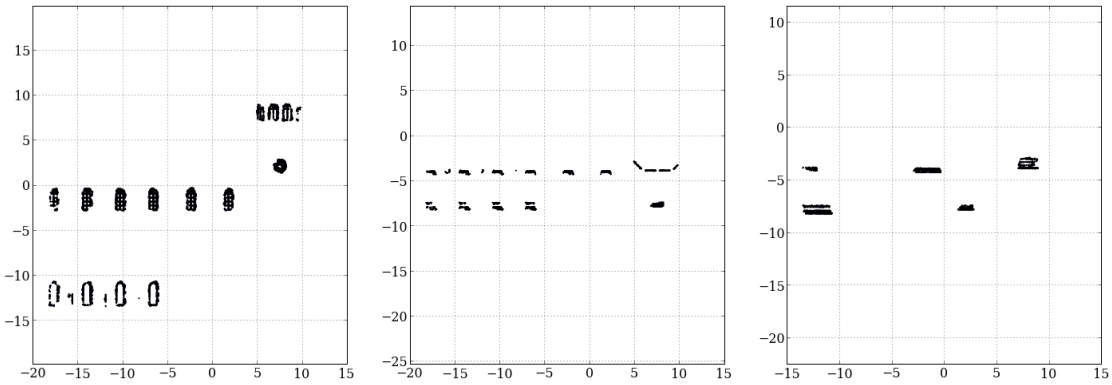
The scenes chosen thus far have been relatively orthodox. For unorthodox architectures such as Figure 5.13, this method breaks down. The current framework would only create a light portal at the top of the tunnel when it should have created it towards the bottom. The sphere that is blocking the light source will cause the algorithm to create two light portals at the outermost opening. Even ignoring that issue, it is not clear which



(a) $(\theta = 0.25^\circ, r = 1.2\text{m})$

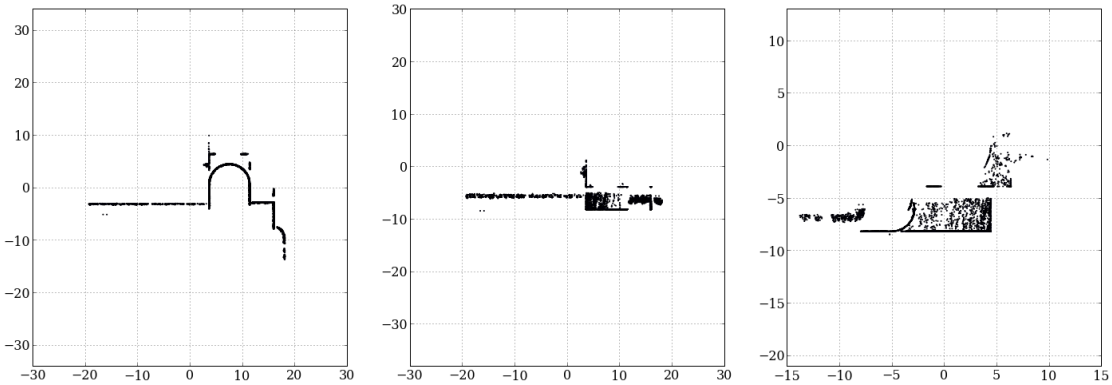


(b) $(\theta = 3^\circ, r = 6.5\text{m})$

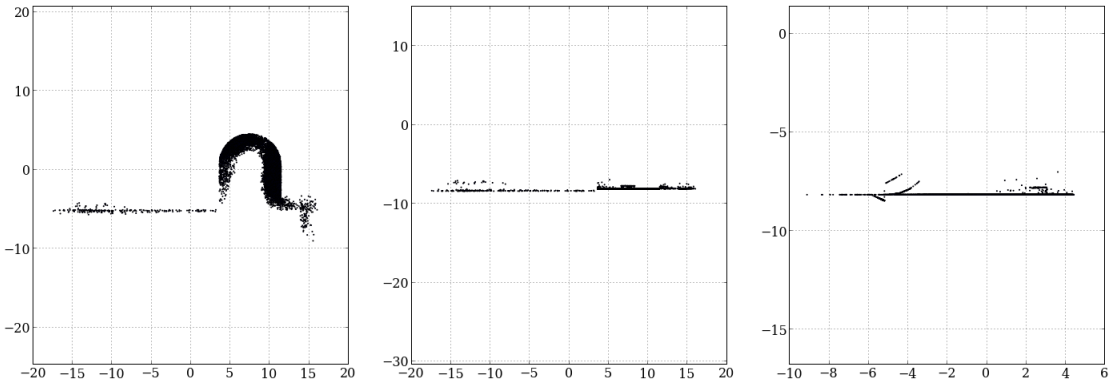


(c) $(\theta = 0.25^\circ, r = 6.5\text{m})$

Figure 5.10. Photon tracing Sibenik with holes. Imagine θ and r as sliders that the user would move back and forth to segment out the holes. θ and r can be adjusted independently and then combined to yield an overall better result. With some more tweaks, the noise in between the lowest set of windows goes away.

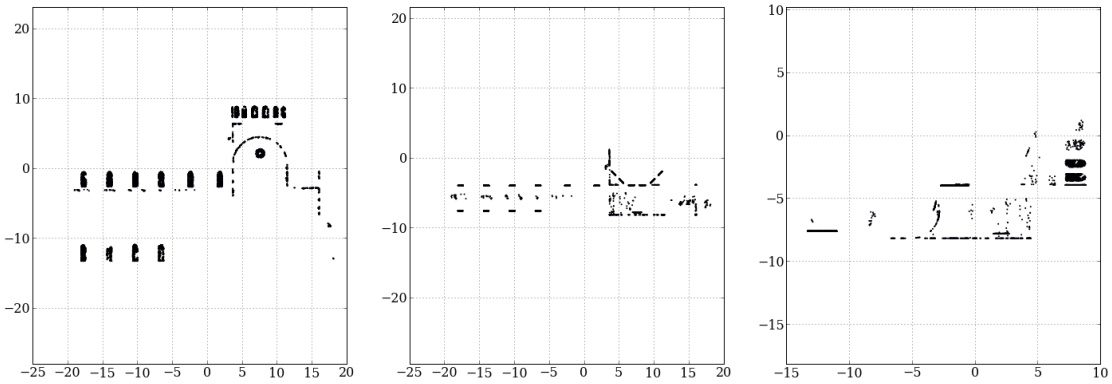


(a) ($\theta = 0.25°, r = 1.2$m)

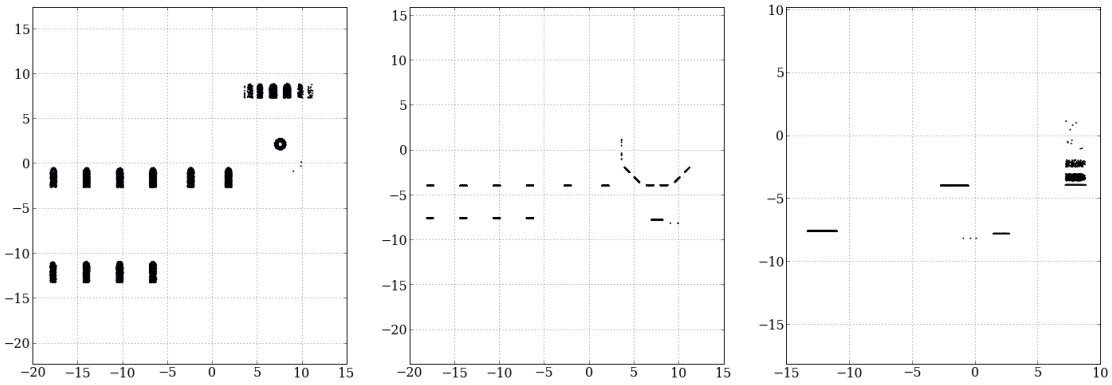


(b) ($\theta = 3°, r = 5.5$m)

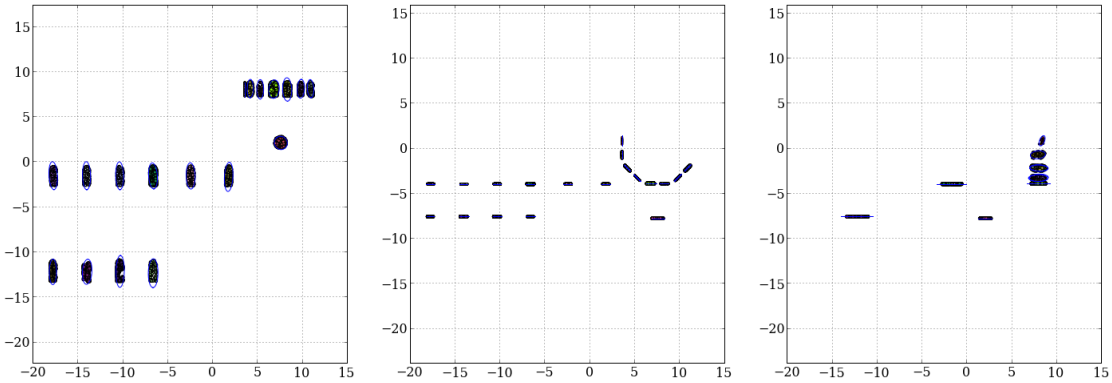
Figure 5.11. Failures of photon tracing Sibenik using only depth discontinuity. The Sibenik's dome is protruding out much more than the thin window frames, so using only depth to segment the windows failed.



(a) $(\theta = 0.25°, r = 1.2\text{m})$



(b) $(\theta = 3°, r = 6.5\text{m})$



(c) $(\theta = 0.25°, r = 6.5\text{m})$

Figure 5.12. Effectiveness of incorporating specular information into PT's detection. Photon tracing using material properties on the previously failed scene detects all the desired Sibenik windows. The good parameters found earlier in Figure 5.10 are reused here since the specular information only restricts the detection to around the windows.

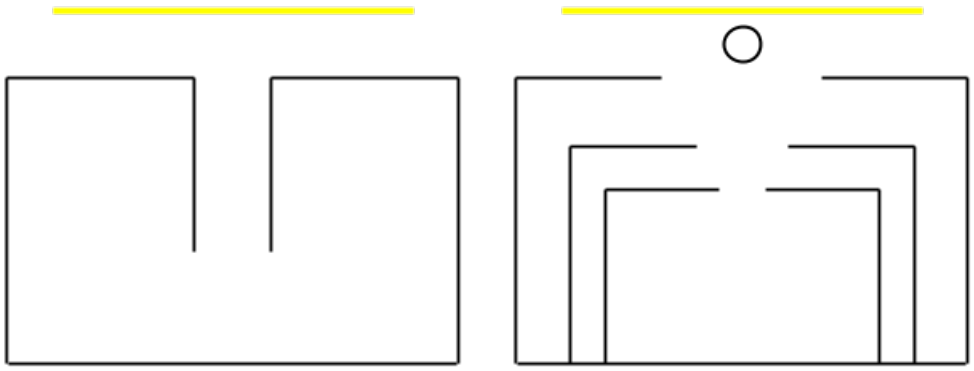


Figure 5.13. Unorthodox scenes that yield ambiguous placement of light portals.

entrance the light portal should cover.

Chapter 6

Conclusions

The contribution of this thesis is an automated method of estimating the light portals in orthodox architectural scenes. The resulting light portals can be integrated into existing renderers.

The process to produce said light portals can be summarized as find the openings in the scene that light penetrates, and estimate a 2D polygon that covers those openings. Photon tracing and DBSCAN takes care of the former while RANSAC MVCE handles the latter. However, RANSAC is merely used to make MVCE less susceptible to noise. As long as the former phase is accurate, the latter is good to go without RANSAC.

The results demonstrate that light portals reduce the required number of samples for a smooth rendered image, and the amount of time needed is on the order of solving the area formulation of the rendering equation. The unoptimized implementation of the above algorithms run in less than a minute; all of them were implemented in Python except photon tracing, which was done in C++. Note that light portals must cover all openings in the scene; otherwise, the final image will be biased unless multiple importance sampling is used. MIS's power heuristic is the established weighing scheme to combine light portals with other techniques.

At the moment, the user must tune the framework's inputs per scene to yield efficient light portals. To alleviate this burden, photon tracing and DBSCAN should

incorporate more surface information in its detection. Adding specular information to the detection have resulted in more robust parameter values. In addition, DBSCAN treats all points equally in its clustering. The Machine Learning community have proposed (and is still actively exploring) extensions to allow each point to limit the influence of its neighbors.

The gain in accuracy from using a perfect light portal versus one that is the output of MVCE is minor. This suggests that trying to replace the ellipse with a more complicated mesh may prove unfruitful. What is worthwhile has been hinted at in Figure 5.13 i.e. figuring out the placement of light portals for indirect lighting. As the path length grows, the effectiveness of light portals (and other techniques) diminishes exponentially. In the scene with the tunnel, it would be much more beneficial to direct rays toward the bottom floor right below the tunnel. Informal experiments have shown that without light portals, the fraction of paths that hit the light over the total number of paths traced is 0.006. With light portals, that fraction increased to 0.30, which means there is still a lot of improvements indirect light portals can provide.

As illustrated in the results, given a set of perfect light portals, sampling them efficiently is not well understood. Another worthwhile direction is to explore scene understanding, which happens to be a growing area of interest in Computer Vision. Light portals at the moment are uniformly selected and sampled at random; the sampling strategy should be directing samples toward regions that contributes the most overall. In exotic unorthodox scenes, there may be many overlapping light portals. Although MIS solves the problem of overcounting, sampling the overlapped areas is wasteful.

Bibliography

Karim T Abou-Moustafa and Frank P Ferrie. The minimum volume ellipsoid metric. In *Pattern Recognition*, pages 335–344. Springer, 2007.

John Amanatides, Andrew Woo, et al. A fast voxel traversal algorithm for ray tracing. In *Proceedings of EUROGRAPHICS*, volume 87, pages 3–10, 1987.

James Arvo and David Kirk. Particle transport and image synthesis. *ACM SIGGRAPH Computer Graphics*, 24(4):63–66, 1990.

Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. *The R*-tree: an efficient and robust access method for points and rectangles*, volume 19. ACM, 1990.

Stephen Poythress Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Sunglok Choi, Taemin Kim, and Wonpil Yu. Performance evaluation of ransac family. In *BMVC*, pages 1–12, 2009.

Philip Dutré. Global illumination compendium. *Cornell University*, 2001.

Philip Dutre, Kavita Bala, Philippe Bekaert, and Peter Shirley. *Advanced global illumination*, volume 11. AK Peters, 2003.

Albert Einstein. On the influence of gravitation on the propagation of light. *Annalen der Physik*, 35(10):898–908, 1911.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. Kdd, 1996.

Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

Stefan Gottschalk. *Collision queries using oriented bounding boxes*. PhD thesis, The University of North Carolina, 2000.

Jeremy J Gray. Olinde rodrigues' paper of 1840 on transformation groups. *Archive for History of Exact Sciences*, 21(4):375–385, 1980.

Robin Green. Spherical harmonic lighting: The gritty details. In *Archives of the Game Developers Conference*, volume 2, pages 2–3, 2003.

Donald P Greenberg, Kenneth E Torrance, Peter Shirley, James Arvo, Eric Lafortune, James A Ferwerda, Bruce Walter, Ben Trumbore, Sumanta Pattanaik, and Sing-Choong Foo. A framework for realistic image synthesis. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 477–494. ACM Press/Addison-Wesley Publishing Co., 1997.

Gene Greger, Peter Shirley, Philip M Hubbard, and Donald P Greenberg. The irradiance volume. *Computer Graphics and Applications, IEEE*, 18(2):32–43, 1998.

Toshiya Hachisuka, Wojciech Jarosz, Richard Peter Weistroffer, Kevin Dale, Greg Humphreys, Matthias Zwicker, and Henrik Wann Jensen. Multidimensional adaptive sampling and reconstruction for ray tracing. In *ACM SIGGRAPH 2008 papers*, SIGGRAPH '08, pages 33:1–33:10, New York, NY, USA, 2008. ACM. ISBN 978-1-4503-0112-1. doi: 10.1145/1399504.1360632. URL http://doi.acm.org/10.1145/1399504.1360632.

John Hannah. A geometric approach to determinants. *The American mathematical monthly*, 103(5):401–409, 1996.

Henrik Wann Jensen. Importance driven path tracing using the photon map. In *Rendering Techniques 95*, pages 326–335. Springer, 1995.

Henrik Wann Jensen. *Realistic image synthesis using photon mapping*. AK Peters, Ltd., 2001.

James T Kajiya. The rendering equation. In *ACM SIGGRAPH Computer Graphics*, volume 20, pages 143–150. ACM, 1986.

Malvin H Kalos and Paula A Whitlock. *Monte carlo methods*. John Wiley & Sons, 2008.

Anton Kaplanyan and Carsten Dachsbacher. Cascaded light propagation volumes for real-time indirect illumination. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, pages 99–107. ACM, 2010.

Alexander Keller and Ingo Wolf. *Efficient importance sampling techniques for the photon map*. Fachbereich Informatik, Univ., 2000.

Leonid G Khachiyan and Michael J Todd. On the complexity of approximating the maximal inscribed ellipsoid for a polytope. *Mathematical Programming*, 61(1):137–159, 1993.

Piyush Kumar and E Alper Yildirim. Minimum-volume enclosing ellipsoids and core sets. *Journal of Optimization Theory and Applications*, 126(1):1–21, 2005.

Eric P Lafortune and Yves D Willems. A 5d tree to reduce the variance of monte carlo ray tracing. In *Rendering Techniques 95*, pages 11–20. Springer, 1995.

Fred Edwin Nicodemus. *Geometrical considerations and nomenclature for reflectance*, volume 160. US Department of Commerce, National Bureau of Standards Washington, D. C, 1977.

Sumanta N Pattanaik, James A Ferwerda, Kenneth E Torrance, and Donald Greenberg. Validation of global illumination simulations through ccd camera measurements. In *Color and Imaging Conference*, volume 1997, pages 250–253. Society for Imaging Science and Technology, 1997.

Kaare Brandt Petersen and Michael Syskind Pedersen. The matrix cookbook, 2006.

Ravi Ramamoorthi. Precomputation-based rendering. *Foundations and Trends® in Computer Graphics and Vision*, 3(4):281–369, 2009.

Iman Sadeghi, Heather Pritchett, Henrik Wann Jensen, and Rasmus Tamstorf. An artist friendly hair shading system. *ACM Transactions on Graphics (TOG)*, 29(4):56, 2010.

Jürgen P Schulze, Daniel Acevedo, John Mangan, Andrew Prudhomme, Phi Nguyen, and Philip Weber. Democratizing rendering for multiple viewers in surround vr systems. In *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, pages 77–80. IEEE, 2012.

Peter Shirley and Changyaw Wang. Direct lighting calculation by monte carlo integration. In *Photorealistic Rendering in Computer Graphics*, pages 52–59. Springer, 1994.

Jonathon Shlens. A tutorial on principal component analysis. *Systems Neurobiology*

Laboratory, University of California at San Diego, 2005.

Peter-Pike Sloan. Stupid spherical harmonics (sh) tricks. In *Game Developers Conference*, volume 2008, 2008.

Peng Sun and Robert M Freund. Computation of minimum-volume covering ellipsoids. *Operations Research*, 52(5):690–706, 2004.

Eric Veach and Leonidas J Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 419–428. ACM, 1995.

Eric Veach and Leonidas J Guibas. Metropolis light transport. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 65–76. ACM Press/Addison-Wesley Publishing Co., 1997.

CM Wang, NC Hwang, YY Tsai, and CH Chang. Ellipse sampling for monte carlo applications. *Electronics Letters*, 40(1):21–22, 2004.

Gregory J Ward. The radiance lighting simulation and rendering system. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 459–472. ACM, 1994.