國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

以密度估計技術實現強韌光傳輸模擬
Density Estimation Techniques for Robust Light Transport Simulation

張子捷

Tzu-Chieh Chang

指導教授：歐陽明博士

Advisor: Ming Ouhyoung, Ph.D.

中華民國 108 年 6 月

June, 2019

doi:10.6342/NTU201901709

國立臺灣大學碩士學位論文
口試委員會審定書

以密度估計技術實現強韌光傳輸模擬

Density Estimation Techniques for Robust Light Transport Simulation

本論文係張子捷君（學號 R05922176）在國立臺灣大學資訊工程學系完成之碩士學位論文，於民國 108 年 6 月 11 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

歐陽明

（指導教授）

莊永裕

葉正聖

系 主 任 莊永裕

doi:10.6342/NTU201901709

誌謝

感謝家人、朋友及師長於在學習期間給予我的幫助以及教導，您們對於維持我在電腦圖學領域的興趣及研究動力功不可沒。也感謝所有曾評論過我各式渲染結果的人們，你們極大程度的維持了我開發的動力。

doi:10.6342/NTU201901709

doi:10.6342/NTU201901709

Acknowledgements

I would like to thank my family, friends, and teachers for providing consistent help and guidance over the years. My constant interest in computer graphics is only possible because of you. I am grateful to all the people that spent their precious time commenting on my renderings as you keep me motivated for developing renderers.

doi:10.6342/NTU201901709

doi:10.6342/NTU201901709

摘要

本研究提出了一套基於密度估計技術的全局光照演算法。此方法奠基於現有之漸進式光子映射法，在疊代的過程中逐步加入更高頻率的光影細節。我們的方法藉由移除所有在場景模型內的奇異點，以強韌地捕捉所有可能的光傳輸路徑，且不必像一般光子映射法需在能量估計階段時進行材質相關參數的計算。我們所提出的光子錐收縮形式能確保在使用有限記憶體的前提下達成一致的估計，並有正式之數學證明與漸進誤差分析實驗以佐證其收斂性。最後，我們將含有複雜光傳輸路徑之場景其渲染結果與現今一致及無偏方法進行了質與量的分析和比較。

關鍵字： 光照傳輸，光子映射，密度估計，全局光照，算繪，渲染，彩現，光線追蹤，光子錐

doi:10.6342/NTU201901709

doi:10.6342/NTU201901709

Abstract

We propose a robust light transport algorithm that is able to capture all possible transport paths where the vast majority of the techniques involved are based on density estimation. It builds on top of existing progressive photon mapping methods where higher frequency lighting effects are added in subsequent iterations. Our approach is capable of eliminating all singularities encountered in a scene representation, and does not require evaluating material properties during the energy estimation process unlike the rest of photon mapping methods. The proposed photon cone focusing scheme is a consistent method that requires only finite memory to converge. A formal proof of convergence is given along with asymptotic analysis of estimation variance and bias. We compare our results to recent consistent and unbiased methods with scenes that contain difficult light paths in both qualitative and quantitative ways.

Keywords: light transport, photon mapping, density estimation, global illumination, rendering, ray tracing, photon cone

doi:10.6342/NTU201901709

doi:10.6342/NTU201901709

Contents

doi:10.6342/NTU201901709

doi:10.6342/NTU201901709

doi:10.6342/NTU201901709

List of Figures

doi:10.6342/NTU201901709

doi:10.6342/NTU201901709

doi:10.6342/NTU201901709

doi:10.6342/NTU201901709

List of Tables

doi:10.6342/NTU201901709

doi:10.6342/NTU201901709

List of Symbols and Abbreviations

Mathematical Symbols

x	position in space
ω	local direction vector around surface normal
ψ	joint position-direction coordinates
θ	deviated angle from surface normal
ϕ	deviated angle from surface tangent
ω_i	incident direction
ω_o	outgoing direction
L_i	incident radiance
L_o	outging/scattered radiance
L_e	emitted radiance
W_e	emitted importance
W_t	sensor response function
f	surface BSDF
$\mathcal{K}$	kernel function of density estimators
$\mathcal{M}$	manifold formed by scene geometries
Ω	domain of scattering directions
ξ	uniform random number in $[0, 1)$
X	sample of position
Θ	sample of direction
Ψ	sample of joint position-direction coordinates

doi:10.6342/NTU201901709

$\mathcal{L}$	sample of radiance
u	first dimention of local parameterization of scene manifold
v	second dimention of local parameterization of scene manifold
σ	spatial bandwidth of photon cones
$\mathcal{P}$	photon map (set of photons)
$\mathcal{P}_\Lambda$	photon cone map (set of photon cones)

Common Abbreviations

GI	global illumination
IBL	image-based lighting
BRDF	bi-directional reflectance distribution function
BSDF	bi-directional scattering distribution function
PDF	probability density function
LTE	light transport equation
PT	path tracing
UPT	unidirectional path tracing
BDPT	bi-directional path tracing
ERPT	energy redistribution path tracing
PM	photon mapping
PPM	progressive photon mapping
SPPM	stochastic progressive photon mapping
MLT	metropolis light transport
PSSMLT	primary sample space metropolis light transport
VCM	vertex connection and merging
NEE	next-event estimation
MIS	multiple importance sampling
PCF	photon cone focusing

Chapter 1

Introduction

Curiosity is a strong driver of human civilization. Being able to simulate phenomena in the observable world is beneficial in many ways. The continuation of Moore's law makes it possible to perform physics simulation with more precision over the years, pushing our understanding of nature to a state never seen before. Light transport, in computer graphics, is among the genre of simulating macroscopic behavior of electromagnetic radiation in the visible spectrum. It is often seen in the design of architecture, visual effects, video games, and much more. Accurately synthesize imageries that are indistinguishable from actual photos and/or measured data in real-time is long being the holy grail of computer graphics. The advent of physically based rendering can be seen as an ongoing attempt to pursue such goal, with the help of more efficient algorithms and cutting edge hardware in each year.

However, with increased computational power, our demand for modeling details has grown even more. Scenes with light paths that are unable to sample using traditional methods in a reasonable amount of time, or complex layered materials with ultra high resolution textures, are no longer being rare. Notably, render time today is not dominated by ray tracing in production environments. Instead, scene assets such as materials now take a significant portion of the render time to fetch and evaluate [7][5]. Hardware accelerations such as nVidia's RTX cards can further increase the traversal performance of acceleration structures, posing other components in a more performance-critical part.

As the field advances, we often see light transport theories that require special care to singularities within a scene, or even unable to render them. Such singularities can

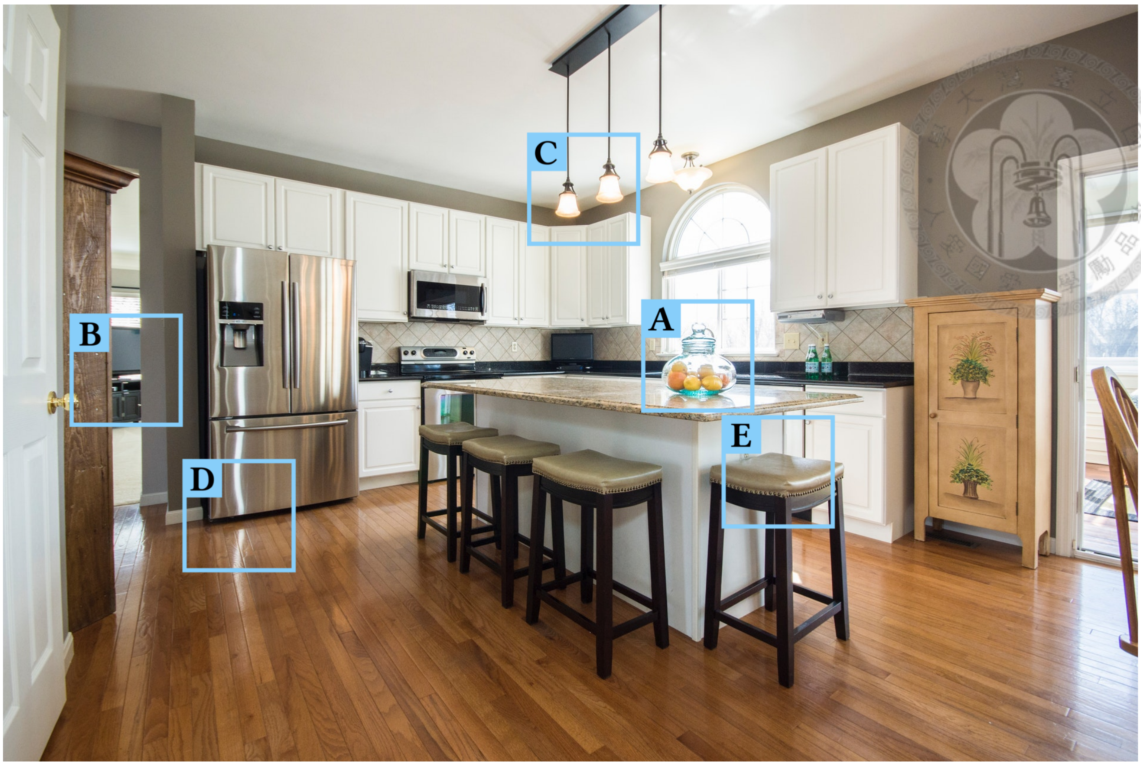


Figure 1.1: A photograph of some usual interior scene with difficult-to-handle lighting phenomena marked. Caustics and its reflections from lighting through glass panels (A). Indirect illumination from adjacent room (B). Small liminares with translucent fixture (C). Glossy interreflections (D). Complex layered material (E). Image courtesy of Milly Eaton from Pexels.

be delta responses in bi-directional scattering distribution function (BSDF) of a material, or energy/importance emitting sources such as point light, directional light, and pinhole camera. The family of path tracing methods is unable to handle caustics generated by point lights and received by pinhole camera under certain configurations, for example. Even with photon mapping methods which excelled at rendering caustics, they rely on finding non-delta BSDFs to merge vertices and will fail if no such material is found. See Section 5.1.1 for discussions regarding this issue.

The main objective of this thesis is to develop a light transport algorithm that is robust in terms of the number of captured light paths, eliminating any singularities encountered while maintaining the nice property of path reuse to gain efficiency. This can also relieve the burden of handling special cases in light transport from programmers. Also, the method should converge to the correct energy estimate of a scene. Moreover, as material evaluation (shader execution) has become an expensive operation nowadays, we

doi:10.6342/NTU201901709

would like to reuse shading results as much as possible. To satisfy our goals, we focus on Monte-Carlo methods that are capable of progressively refine the result with cached values. This will allow us to reuse and refine calculated results iteratively, and terminate the render process if sufficient quality is obtained.

1.1 Related Work

Photon mapping algorithms are a family of global illumination methods first introduced by Jensen and Christensen [19]. It is a two-pass rendering algorithm that traces photons (particles) in the first pass, then visualizes scene illumination in the second pass using the constructed photon map. The photon map is essentially an approximation of the illumination information of a scene, which can be cached in subsequent visualization passes. However, the original algorithm is limited by the amount of memory the computing device has, and suffers from low-frequency biases that can only be cleared by increasing photon density (which demands more memory).

Veach [31] introduced several algorithms that are essential to later studies. Important to our work is the path integral formulation and a formal description of particle tracing methods under a unified mathematical framework. Furthermore, his variance reduction techniques such as multiple importance sampling (MIS) is among the core of our algorithm. He also identified that there exist certain arrangements of emitters (lights and sensors) and materials that are unable to handle by any unbiased methods.

Later, Hachisuka et al. proposed the progressive photon mapping (PPM) method [11] and an improved version called stochastic PPM (SPPM) [10]. These methods greatly increase the usability of photon mapping as they can utilize an effectively unlimited number of photons for rendering, and the resulting image converges to the ground truth with enough number of iterations. The fact that view paths (paths starting from the camera) are still handled by canonical path tracing (PT) makes PPM and SPPM inherit the properties of PT hence the inefficiencies in handling certain light paths.

Knaus and Zwicker generalized Hachisuka's work using a probabilistic point of view [24], removing the need for storing view path buffers. Kaplanyan and Dachsbacher further in-

doi:10.6342/NTU201901709

vestigated its statistical properties and proposed an adaptive PPM (APPM) algorithm [22] that optimally balances the noise and bias trade-off in photon mapping methods. Following APPM is the path space regularization framework again introduced by Kaplanyan and Dachsbacher [23]. The path space regularization framework can include the aforementioned photon mapping methods as a special case to their mathematical formulation. Their method mollifies delta distributions in BSDFs and is, in theory, able to handle all possible light transport paths. Our photon cone radiance estimator is similar to this work on specular surfaces and SPPM on the iteration process, except that we store and reuse particle records on all types of materials including delta distributions. Bouchard et al. [2] included specular mollification (angular regularization) in the path space regularization work as an additional sampling strategy much like the way spatial regularization is combined (vertex connection and merging, the VCM method, by Georgiev et al. [8]; and unified path sampling, the UPS method, by Hachisuka et al. [12]). Our method also puts the spatial and angular domain in a unified mathematical framework, with differences as we directly extend probabilistic PPM into a joint position-directional space, and does not require material evaluation during the radiance estimating process.

Complex luminaires which are composed of hundreds of individual emitting, reflecting, and transmitting parts are a typical example of a *hard* scene. Several works are dedicated to modeling complex luminaires with caches. Edgar et al. [33] models light sources such as chandeliers with anisotropic point lights and radiance volumes. And Kniep et al. [25] extends Jensen's photon mapping algorithm to incorporate angular domain in order to model the emission profile of automotive tail lamps and other complex light sources. Inspired by their 4-D kernel, we design our kernel with a parameter controlling the relative trade-off of bias and error between spatial and angular domain. Our approach can also apply to their work with an additional benefit that a consistent estimate can be obtained in the limit with finite memory.

These works laid the foundation for developing our photon cone focusing (PCF) algorithm.

doi:10.6342/NTU201901709

1.2 Summary of Original Contributions

We list major contributions and key insights developed in this work as follows:

A Robust Light Transport Formulation

Our formulation of photon cone radiance estimate can handle all types of transport paths, and it is robust in terms of the number of captured lighting phenomena. It is a two-pass method that does not require the accessibility of material properties in the radiance estimation stage (second pass). The cached radiance map is in itself a standalone light field representation independent of material descriptions.

Convergence Analysis

Mathematical proofs, as well as experimental results, are given assuring the image rendered by our method converges to the ground truth.

Multiple Importance Sampling for Photon Cones

We developed several multiple importance sampling schemes for generating particle records, improving statistical behaviors of the proposed light transport algorithm.

A Formal Description of Photon Mapping

We identify the fact that photon mapping has two common interpretations and most works do not clearly distinguish one from the other, which can lead to confusion in understanding the details of a proposed method. A concise description is given to clarify the issue.

New 4-D Kernel for Estimating Surface Energy

A joint position-directional kernel is designed to estimate and control the relative trade-off of bias and error between spatial and angular domain. We analyzed bounds for the required parameter to ensure consistent renderings.

doi:10.6342/NTU201901709

1.3 Thesis Organization

The thesis is divided into 6 chapters, providing background and in-depth descriptions regarding light transport using density estimation techniques. In the following parts, we provide general knowledge for understanding the photon mapping method and its advanced variants in Chapter 2. In Chapter 3, we introduce the photon cone radiance estimator and the photon cone focusing (PCF) method, with mathematical proofs for its convergence. Chapter 4 is dedicated to implementation details of PCF such as the rendering pipeline and several multiple importance sampling techniques, along with guidance on controlling several input parameters. Then, in Chapter 5, we identify characteristics of the proposed method and compare our results to existing global illumination methods. Finally, conclusions and potential future works are given in Chapter 6.

We provide appendices which cover more details in the context of the referencing chapters.

doi:10.6342/NTU201901709

Chapter 2

Understand Photon Mapping

Particle[1] tracing methods are widely used in rendering lighting setups where path sampling algorithms have low probability in producing a non-zero path contribution, e.g., *SDS* paths. Notably, photon mapping as originally proposed by Jensen [18] was effective in rendering caustics and has received much developments and extensions over the years. It is often being described as shooting energy packets from light sources and record their contribution on the surfaces in the scene (the *photon map*), which can later be utilized to actually render the scene (*visualization*). Despite being intuitive, its relation to the rendering equation [21] and how surfaces interact with the energy packets are rarely described formally in papers and online literatures[2]. Moreover, photons usually carry radiant flux which implies the total flux of the lights in a scene must be known. Such information is often not readily available or difficult to obtain accurately; for instance, one might resort to Monte-Carlo methods to obtain radiant flux for procedural lights. It is also not straightforward to extend the energy packet interpretation in case other quantities are being transported.

This chapter is intended to glue the concepts in photon mapping altogether in a single place, with the assumption that readers are familiar with unbiased light transport algorithms (especially unidirectional path tracing). In particular, solid theoretical foundations 2.2 are provided for understanding its advanced variants.

[1] In computer graphics, the word *photon* is often used in place of particle. It is not the same as photon in the field of physics.

[2] Note that Hachisuka made a similar observation in his dissertation [9, Section 4.4.3].

2.1 Energy Packet Interpretation

This section serves as a quick review of the traditional interpretation of photon mapping. We do not go into details as good references exist [20], and the following derivation is included for completeness.

2.1.1 Photon Shooting Pass

As mentioned earlier, photon mapping methods was first formulated as discretizing light energy into small packets and trace them using standard Monte-Carlo techniques [28] [18]. Often, the energy packets carry *radiant flux* which is also known as the *watt* (abbreviated as W; or J/s, for joule per second) in SI unit, denoted as

$$\Phi = \frac{\partial Q}{\partial t}, \tag{2.1}$$

where Q is the emitted radiant energy from lights and t is time. It is advised to shoot photons according to the distribution formed by emitted energy of each light, such that each photon does not differ too much in carried energy Φ_j. Every time a photon hit a non-specular surface, its incoming direction $\omega_{i,j}$ and carried energy Φ_j are recorded. Note that for conservation of energy, photons should further satisfy the normalization condition

$$\Phi_0 = E\left[\sum_{j=1}^{M} \Phi_j\right], \tag{2.2}$$

where Φ_0 is the total energy emitted in the scene and M is the amount of photons.

2.1.2 Visualization Pass

After the photon shooting pass, the solution of the rendering equation [21]

$$L_o\left(x, w_o\right) = L_e\left(x, w_o\right) + \int_{\Omega} L_i\left(x, w_i\right) f\left(x, w_i, w_o\right) \cos\theta_i dw_i, \tag{2.3}$$

can be approximated using the information stored previously (the *photon map*). Recall that irradiance, the radiant flux received by a surface per unit area, has the form

$$E = \frac{\partial \Phi}{\partial A}, \tag{2.4}$$

where A is the surface area; and radiance, the radiant flux received by a surface per unit area per unit projected solid angle

$$L = \frac{\partial^2 \Phi}{cos\theta_\omega \partial A \partial \omega} = \frac{\partial E}{cos\theta_\omega \partial \omega}, \tag{2.5}$$

where θ_ω is the angle between surface normal and ω. Assuming E depends only on the variables mentioned, after applying a change of variable $dE_i = cos\theta_i L_i d\omega_i$ with the incoming irradiance E_i and substituting into the rendering equation, we have

$$L_o(x, w_o) = L_e(x, w_o) + \int_\Omega f(x, w_i, w_o)\, dE_i. \tag{2.6}$$

Since each photon carries a differential amount of energy $\Phi_j = \Delta\Phi_0$ from the total emitted energy Φ_0, the irradiance arriving at some surface can be approximated easily

$$\Delta E_i = E_j = \frac{\Delta\Phi_0}{\Delta A} = \frac{\Phi_j}{\pi r^2}, \tag{2.7}$$

where ΔA is often replaced with the 2-D area of a circle under the assumption that the surface geometry is locally flat around the position of a photon. It is now clear that equation (2.6) can be solved using the photon map

$$L_o(x, w_o) \approx L_e(x, w_o) + \frac{1}{\pi r^2} \sum_{j=1}^{M} f(x, w_i, w_o)\, \Phi_j, \tag{2.8}$$

where the summation selects only photons around x with a radius r.

With equation (2.8) at hand, a lookup table for excitant radiance $L_o(x, w_o)$ of the scene (the photon map) can be constructed. Finally, image synthesis techniques, e.g., path tracing (see appendix A), can utilize the table to render images.

doi:10.6342/NTU201901709

2.2 Measurement Interpretation

Lights in modern scenes are typically modeled with spatially varying emission profiles such as textured area lights, IBL techniques, or even procedurally generated sources. Energy packet interpretation 2.1 does not fit well in current situations since other than diffusive area lights, the calculation of radiant flux of a light source is quite involved. For example, an environment map that models light sources infinitely far away with filters applied; or an arbitrary emission profile applied on an implicit surface. In such cases, irradiance E or radiance L are more natural quantities to trace and store. This section is intended to give a generalized interpretation of photon mapping that is able to work with arbitrary transport quantities, which is also crucial to our *photon cone focusing* method.

2.2.1 Mathematical Formulation

Our derivation is based on the particle tracing framework by Veach [31, Appendix 4.A]. For rendering an image, the goal is to obtain pixel radiance L_{pixel} by solving the measurement equation

$$L_{\text{pixel}} = \int_{A_{\text{sensor}}} \int_{\Omega} W_e \left(x_1 \to x_0\right) L_i \left(x_1 \to x_0\right) d\omega dA, \tag{2.9}$$

where W_e is the sensor response function and L_i is the incoming radiance. Here, we write the function arguments $(x_k, \omega_{i,k})$ as $(x_{k+1} \to x_k)$ for clarity reasons[3]. A more general description of the measurement equation can be found in A.2. Also, the radiance that comes from x_{k+1} to x_k can be written as[4]

$$\begin{aligned} L_o \left(x_{k+1} \to x_k\right) &= L_e \left(x_{k+1} \to x_k\right) \\ &+ \sum_{n=1}^{\infty} \left\{ \int_{\Omega^n} L_e \left(x_{k+n+1} \to x_{k+n}\right) \cdot \prod_{i=1}^{n} \Big[f \left(x_{k+i+1} \to x_{k+i} \to x_{k+i-1}\right) \cos\left(\theta_{k+i}\right) \Big] d\omega^n \right\}. \end{aligned} \tag{2.10}$$

[3] Not to confuse our notation with the *three-point* form rendering equation. The derivation here does not perform the necessary change of integration domain yet.

[4] See A.1 for the origination of the formula.

As an example, to solve L_{pixel}, we can first (somehow) obtain a solution for (2.10) with $k = 0$ under steady-state. After that, we have

$$L_o(x_{k+1} \to x_k)|_{k=0} = L_o(x_1 \to x_0) = L_i(x_1 \to x_0). \tag{2.11}$$

We can later substitute this result into the measurement equation, then L_{pixel} can be obtained using any integration technique.

Photon mapping is a two-pass method in the sense (2.10) and (2.9) can be solved separately, i.e., the former can be approximated by the *photon mapping pass* and the later is solved during the *visualization pass*. One of the reasons that make photon mapping efficient is that the solution for L_i generated in the photon mapping pass is cached and reused in the visualization pass.

Unbiased Particle Tracing

We now use the term *particle* in place of photon to describe the records generated for the visualization pass. The choice is made due to the fact that the measurement interpretation is capable to work with any emitted quantities. The following is a generalized version of the measurement equation

$$\mathcal{Q} = \int_A \int_\Omega W_t(x_{\ell+1} \to x_\ell)\, L_i(x_{\ell+1} \to x_\ell)\, d\omega dA, \tag{2.12}$$

where $\mathcal{Q}$ is an arbitrary quantity measured by the sensor response function W_t. Note that the sensor domain A should not be restricted on a camera sensor. Any surface that is a subregion of the 2-D manifold $\mathcal{M}$ formed by scene geometries can be a *hypothetical sensor*.

Now we limit ourselves to the case of synthesising an image with length $k+1$ paths only, and we want to construct a lookup table for $L_o(x_k \to x_{k-1})$ to facilitate the process. Denoting the manifold formed by the reachability of length $n = 1$ paths starting from lights as $\mathcal{M}_k$. Samples of $L_i(x_{k+1} \to x_k)$ can be obtained by performing light tracing from x_{k+1} and the sampled value $\mathcal{L}_i(X_{k+1} \to X_k)$ is directly stored on $\mathcal{M}_k$. Suppose

M samples are stored, and the j-th one is written as $\mathcal{L}_i\left(X_{k+1,j} \to X_{k,j}\right)$. As each stored sample is an unbiased estimate of the incoming radiance for length $n = 1$ paths from light sources, the measurement $\mathcal{Q}_k$ on $\mathcal{M}_k$ can be estimated, also unbiasedly, as

$$\mathcal{Q}_k \approx \frac{1}{M} \sum_{j=1}^{M} W_t\left(X_{k+1,j} \to X_{k,j}\right) \mathcal{L}_i\left(X_{k+1,j} \to X_{k,j}\right). \tag{2.13}$$

This summation indicates that any quantity that is linearly accumulable can be estimated with a suitable choice of W_t.

2.2.2 The Zero Contribution Problem

Our goal is to compute $L_o\left(x_k \to x_{k-1}\right)$. From estimator (2.13), the quantity $\mathcal{Q}_k$ should be measured as radiance value. To find W_t for a radiance measure, the integral on k-th vertex of the rendering equation is extracted as

$$L_o\left(x_k \to x_{k-1}\right) = \int_{\Omega_k} L_i\left(x_{k+1} \to x_k\right) f\left(x_{k+1} \to x_k \to x_{k-1}\right) \cos\left(\theta_k\right) dw_k. \tag{2.14}$$

Being a point measure of outgoing radiance on x_k, it is equivalent to measure radiance on $\mathcal{M}_k$ with a delta distribution[5]

$$\int_{\mathcal{M}_k} \delta\left(x'_k - x_k\right) \int_{\Omega_k} L_i \cdot f\left(x_{k+1} \to x'_k \to x_{k-1}\right) \cos\left(\theta_k\right) dw_k dA', \tag{2.15}$$

where x'_k are positions on $\mathcal{M}_k$ and some arguments are omitted in the integrand for clarity. Observing equation (2.15), it can be concluded that in order to obtain an unbiased measurement of L_o, the hypothetical sensor response function should be defined as

$$W_t\left(x_{k+1} \to x'_k\right) = \delta\left(x'_k - x_k\right) f\left(x_{k+1} \to x'_k \to x_{k-1}\right) \cos\left(\theta_k\right), \tag{2.16}$$

[5] A more rigorous form would be writing the equation as a Lebesgue integral.

and a corresponding estimator for (2.15) can be derived from (2.13)

$$L_o\left(x_k \to x_{k-1}\right) \approx \\ \frac{1}{M}\sum_{j=1}^{M} \delta\left(X_{k,j} - x_k\right) f\left(X_{k+1,j} \to X_{k,j} \to x_{k-1}\right) \mathcal{L}_i\left(X_{k+1,j} \to X_{k,j}\right) \cos\left(\Theta_{k,j}\right). \tag{2.17}$$

Although M samples of L_i on the k-th vertex were generated and stored as $\mathcal{L}_{i,j}$ in the unbiased particle tracing step 2.2.1 and those records can be utilized in the estimator (2.17), this approach is of little to no benefit. To see this, suppose unidirectional path tracing (UPT) is used to solve the pixel measurement equation (2.9), and let X'_ℓ be the ℓ-th randomly sampled vertex. The estimator is thus responsible for providing information about $L_o\left(X'_k \to X'_{k-1}\right)$. It is unfortuante, by the way W_t is structured, estimator (2.17) now contains a term

$$\delta\left(X_{k,j} - X'_k\right). \tag{2.18}$$

UPT has zero probability to randomly sample an exact location $X'_k = X_{k,j}$ that is associated with a stored sample $\mathcal{L}_{i,j}$ for it to utilize. As a result, length $k+1$ path will always have zero contribution to the final image.

To formally state the problem, $\delta\left(x'_k - x_k\right)$'s infinitesimal bandwidth is the cause of the zero contribution issue. As a side note, it is possible to use stored values $\mathcal{L}_{i,j}$ by directly connect sampled path vertex X'_{k-1} to $X_{k,j}$. This technique is known as bi-directional path tracing (BDPT) which is outside the scope of this chapter.

2.2.3 A Biased Solution

To solve the problem of zero contribution, photon mapping replaces the delta distribution $\delta\left(x'_k - x_k\right)$ with a canonical kernel $\mathcal{K}\left(x'_k - x_k\right)$ that integrates to 1 over its support

$$\int_{\mathcal{M}_k} \mathcal{K}\left(x'_k - x_k\right) dA = 1. \tag{2.19}$$

The normalization condition is of the essense since it ensures correct weighting of the measured quantity. The kernel $\mathcal{K}$, however, introduces biases to the integral (2.15) as well as the estimator (2.17).

Proximity Bias

Proximity bias is commonly known as an intrinsic source of bias in density estimation methods (we will see that both energy packet and measurement interpretations of photon mapping are a form of density estimation in Section 2.3). Formally, it is present due to the fact that a blurred equation

$$\int_{\mathcal{M}_k} \mathcal{K}\left(x'_k - x_k\right) \int_{\Omega_k} L_i \cdot f\left(x_{k+1} \to x'_k \to x_{k-1}\right) \cos\left(\theta_k\right) dw_k dA, \tag{2.20}$$

is solved instead of the original one (2.14). Implementation-wise, the estimator for (2.20) would be

$$\frac{1}{M} \sum_{j=1}^{M} \mathcal{K}\left(X_{k,j} - x_k\right) f\left(X_{k+1,j} \to X_{k,j} \to x_{k-1}\right) \mathcal{L}_i\left(X_{k+1,j} \to X_{k,j}\right), \tag{2.21}$$

which requires the ability to gather particle records $\mathcal{L}_{k,j}$ where kernel $\mathcal{K}$ has non-zero value. Such reconstruction scheme will inevitably induce aliases if the variation of the lighting phenomena is above the Nyquist frequency. Several techniques have been proposed to better reconstruct lighting information, including photon differentials [27] and anisotropic kernels [6]. Progressive algorithms solve this issue by gradually reduce the bandwidth of $\mathcal{K}$ and eventually achieve infinite amount of sample points in an infinitesimal region, eliminating all biases (in the limit, see 2.5).

Topological Biases

We see topological biases as a by-product of proximity bias. Topological biases arise due to the assumption that the surface geometry, i.e., $\mathcal{M}_k$, is locally flat around the position of a particle record, which is what the kernel normalization condition (2.19) usually depends on. Such an assumption can easily break down in several scenarios

Several methods exist to directly tackle topological biases, including ray maps [13] and photon ray splatting [16].

Generalization

So far, the mathematical framework presented earlier was limited to the case of length $k+1$ paths only, and particle records were generated for length 1 paths from lights. Note, however, the derivation can be easily extended to arbitrary path lengths by substituting different values of k. As long as particle records $\mathcal{L}_{i,j}$ are generated unbiasedly, the estimator (2.21) holds for any traced length[6]

2.3 Kernel Density Estimation

The estimator in energy packet (2.8) and measurement intepretations (2.21) both share the same form

$$\frac{1}{M}\sum_{j=1}^{M}\mathcal{K}\left(X_j^n - x^n\right) w_j, \tag{2.22}$$

where w_j is some sample weight generated by a random process. In situations where w_j is almost a constant or its local variation is low, the estimator is effectively the same as *kernel density estimation* [29], which implies that they can share similar properties. Kernel density estimation provides a non-parametric way to esimate a probability density $p\left(x\right)$ with only observable samples being drawn from it. We denote the estimated probability density as $\hat{p}\left(x\right)$. The method states that to obtain an approximation of $p\left(x\right)$, the following estimator can be used

$$p\left(x\right) \approx \frac{1}{Mh}\sum_{i=1}^{M}\mathcal{K}\left(\frac{x - x_i}{h}\right) = \hat{p}\left(x\right), \tag{2.23}$$

and h is a variable bandwidth[7].

It is worth noting that in photon mapping methods, the target function is usually the ra-

[6] The correctness of this result is straightforward to see in a bi-directional radiance estimation framework. For a more rigorous description, see [31, Appendix 4.A].

[7] It is possible to generalize the kernel density estimator to n dimensions, see [26, Section 16.2.2]. We kept the univariate form here for simplicity.

diance field L_o, which is neither normalized nor estimated with unweighted samples. From the weighted estimator (2.22), it would be more suitably called *kernel smoother*, which is a statistical technique to obtain function value by a weighted average of neighboring samples. Nevertheless, such observation is useful as it enabled us to analyze induced errors, e.g., *mean integrated square error* (MISE) or more commonly used *asymptotic integrated mean square error* (AMISE).

2.4 The Complete Photon Map

Normally, photon maps contain particle records $\mathcal{L}_j$ that will be utilized to construct samples for estimating outgoing radiance. Denoting a photon map that stores M length[8] n particles as $\mathcal{P}_{n,M}$. A photon map can be the combination of N maps, each with a different stored length. We call the one that combines all path lengths and phonemona as a *complete* photon map $\mathcal{P}_c$ defined as

$$\mathcal{P}_c = \mathcal{P}_{\mathbb{Z}^+,NM} = \bigcup_{n=1}^{N} \mathcal{P}_{n,M},\ N \to \infty. \tag{2.24}$$

With finite memory, such photon map is possible using *russian roulette* and omitting zero-weight particles during the tracing stage as they will not contribute to the summation (2.25).

In order to solve the measurement equation and obtain our goal, L_{pixel}, some integration technique is chosen and we use the photon map as a lookup table for $L_i\left(x_{k+1} \to x_k\right)$ with arbitrary k's. Equivalently, since $\mathcal{P}_c$ stores estimations for all kinds of path length, it is the photon map to use if all light paths are required. The *complete photon radiance estimator* can now be defined as

$$L_o\left(x \to x'\right) \approx \frac{1}{M} \sum_{j=1}^{|\mathcal{P}_c|} W_t\left(x, X_j\right) \mathcal{P}_{c,j}. \tag{2.25}$$

The sensor response function W_t has finite support in most if not all algorithms, and non-

[8] Path length starting from lights.

doi:10.6342/NTU201901709

contributing particles are often culled by spatial acceleration structures. As a result, a single range query to $\mathcal{P}_c$ will suffice for calculating the summations in the estimator. Also note the $1/M$ factor in the estimator, as we are effectively approximating N integrals at the same time, and energy from all path lengths sould accumulate to $L_o\left(x \rightarrow x'\right)$ (as far as geometric optics is concerned, energy from different sources can be linearly accumulated).

To simplify the expression and for future generalizations, we denote the arbitrary measurement Q taken on some location $\boldsymbol{\xi}$ (not necessarily 2-D) which is approximated using a photon map as

$$\mathcal{Q}\left(\boldsymbol{\xi}, h, \mathcal{P}_{n,M}\right), \tag{2.26}$$

with h being the bandwidth used for the approximation. As an example, equation (2.25) can now be written as

$$L_o\left(x \rightarrow x'\right) \approx \mathcal{L}\left(x \rightarrow x', r, \mathcal{P}_c\right). \tag{2.27}$$

2.5 Progressive Extensions

It is possible to eliminate energy estimation bias using photon maps. Generally speaking, a consistent[9] result is achieved via progressively shrinking the bandwidth of the kernel in use along with other techniques in order to arrive at the state of infinite photon density.

2.5.1 The XPPM Family

Progressive photon mapping (PPM) [11] and stochastic progressive photon mapping (SPPM) [10] are the classic progressive extensions of photon mapping. They are capable of achieving a zero-bias result in the limit, i.e., equal to the ground truth. However, they require a large amount of memory for storing additional information on path vertices starting from the camera, no matter the implementation is a canonical or reversed one [14].

To further reduce the memory usage of SPPM, a probabilistic based derivation of SPPM was proposed [24] (we call it probabilistic progressive photon mapping, PPPM).

[9] Hachisuka's *Five Common Misconceptions about Bias in Light Transport Simulation* is a good read to understand unbiasedness and consistency in light transport algorithms.

PPPM does not require the storing of any additional information on path vertices to consistently estimate light energy using photon maps. This is possible as the convergence of SPPM is, in fact, not dependent on the local kernel bandwidth and can be replaced by a global bandwidth that is shared by all path vertices. The introduction of adaptive progressive photon mapping (APPM) [22] not only significantly reduce the variance in energy estimation, they also showed that the global bandwidth does not need to be calculated progressively.

We call these methods collectively as the *XPPM family*, since they are all progressive variants of the photon mapping algorithm.

2.5.2 Bi-directional Unification

There are certain types of paths that can be handled better with unbiased methods such as path tracing, and some are better with density estimation based methods like photon mapping. By formulating density estimation and bi-directional path tracing (BDPT) under a unified mathematical framework, it is possible to combine the XPPM family with BDPT to achieve better results. Notable unifications are vertex connection and merging (VCM) [8] and unified path sampling (UPS) [12].

In a unified framework, density estimation is being treated as a new sampling technique in addition to all camera and light subpath combinations. Connecting sampled subpaths originated from camera and light directly is called *vertex connection*, while density estimation connects nearby subpaths and is called *vertex merging*. These operations can be further combined (weighted) using *multiple importance sampling* [32].

These techniques are however often being considered *heavyweight* implementation-wise. Nonetheless, several production systems implemented VCM/UPS including Pixar's RenderMan (exposed as PxrVCM integrator) and Weta Digital's Manuka [5].

2.5.3 Mollifier, Relaxation, and Beyond

Even with a unified framework for XPPM and BDPT, there exist paths that are unsamplable with both methods, meaning that scenes dominated by those light transport paths

doi:10.6342/NTU201901709

may be partially or completely black if rendered using VCM/UPS (see Section 5.1.1 for more information). One of the first light transport algorithms to address such issue is the path space regularization (PSR) framework [23], which is possible to apply to all path sampling methods. PSR is based on selectively mollify the singularities encountered in a scene description, such that they are no longer unsamplable. In theory, their mathematical framework can describe all possible consistent methods.

A unique variant of progressive photon mapping is photon relaxation. Progressively relaxed photon map [30] can lead to low variance energy estimation by virtually balancing the stored photons, and feature detection can be added for preserving fine details such as complex caustics.

Finally, extending photon mapping to volumetric light transport is also an ongoing area of research. In addition to Jensen's work [18], it is worth noting that by increasing the dimensionality of photon samples (from photon points, beams to photon planes, volumes) it is possible to achieve unbiased energy estimation using photon maps [1] [4].

doi:10.6342/NTU201901709

doi:10.6342/NTU201901709

Chapter 3

Derivations of Photon Cone Focusing

In this chapter, we extend canonical density estimation techniques for image synthesis to higher dimensions. In particular, our formulation progressively refines the radiance estimation on a point in space and converges to the true radiance function in the limit. The derivation focuses on estimating L_o on the scene manifold $\mathcal{M}$, and the mathematical framework makes extensions to volume rendering straightforward. Also note that it is possible to estimate arbitrary energy function provided the particle records are correctly traced 2.2.

The following sections focus on mathematical analyses of transient and statistical behavior of the *photon cones*, which is based on a 4-D kernel (3.1). In Section 3.2, we modified canonical radiance estimate into one that directly measure the outgoing radiance without the need of evaluating material properties during the visualization pass. Later from Section 3.3 to 3.5, formal proof is given which shows that the proposed method converges to the ground truth. Finally, the pseudocode of the algorithm is given in Section 3.6, and generalizations that make the method applicable in more scenarios are shown in the last section (3.7).

3.1 Choosing a 4-D Kernel

As our method is intended to directly reconstruct outgoing radiance L_o from particle records stored on the scene manifold $\mathcal{M}$, a 4-D kernel is required due to the fact that

L_o is a 4-D function (possible parameterizations include (x, ω) or (x_0, x_1)). Denoting the kernel as $\mathcal{K}(\psi' - \psi)$, with ψ being a joint representation of position and direction, we first decompose it into a product of positional $\mathcal{K}_p$ and directional $\mathcal{K}_d$ parts

$$\mathcal{K}(\psi' - \psi) = \mathcal{K}_p(x' - x)\,\mathcal{K}_d(\omega' - \omega)\,. \tag{3.1}$$

The normalization condition from (2.19) should also be obeyed for correct radiance measure

$$\int_\Omega \int_\mathcal{M} \mathcal{K}_p(x' - x)\,\mathcal{K}_d(\omega' - \omega)\,dA d\omega = 1. \tag{3.2}$$

For positional part, all the variants from density estimation literatures can be used. In case of minimizing the *asymptotic integrated mean square error* (AMISE), studies have shown [29, Chapter 3] [15, Section 3] that the performance of a variety of kernels do not differ much, including a theoretically optimal one[1]. Thus, our choice of $\mathcal{K}_p$ is a box kernel in 2-D due to its computational efficiency (requiring only a scalar multiplication)

$$\mathcal{K}_p(x' - x) = \begin{cases} \dfrac{1}{\pi r^2}, & \text{if } \|x' - x\| < r \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

For directional part, a kernel that is easier to perform normalization and change of variables is preferred since this would facilitate the analysis of convergence. Our assumption is that the choice of directional kernel does not affect performance similar to the case in positional kernels, as well as their product. Several kernels were investigated, including von Mises-Fisher distribution, GGX [34], and directional variants of positional kernels. We pick a distribution that has a shape of cone (essentially a directional extension of box kernel) as it meets our requirements decently

$$\mathcal{K}_d(\omega' - \omega) = \begin{cases} \dfrac{1}{2\pi\left[1 - cos\left(\theta_\omega\right)\right]}, & \text{if } \theta(\omega' - \omega) < \theta_\omega \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

[1] They all have the efficiency metric $C(K)$ close to 1, while the Epanechnikov kernel has $C(K) = 1$.

To normalize the kernel, integrating over its support and scale the result to 1 is necessary. Let $\mathcal{S} = \Omega \times \mathcal{M}$ as $\mathcal{K}$'s support, it can be shown that a properly normalized kernel is

$$\begin{aligned}
&\int_{\mathcal{S}} \mathcal{K}\left(\psi' - \psi\right) d\psi \\
&\quad = \int_{\Omega}\int_{\mathcal{M}} \mathcal{K}_p\left(x' - x\right)\mathcal{K}_d\left(\omega' - \omega\right) dA d\omega \\
&\quad = \int_{\Omega}\int_{\mathcal{M}} \frac{1}{2\pi^2 r^2\left[1 - cos\left(\theta_\omega\right)\right]} dA d\omega \\
&\quad = 1.
\end{aligned} \tag{3.5}$$

Finally, we write our kernel $\mathcal{K}\left(\psi' - \psi\right)$ in its local space as

$$\mathcal{K}_\sigma = \frac{1}{2\pi^2\sigma^2\left[1 - \cos\left(t\left(\sigma\right)\right)\right]} = \frac{1}{C_\sigma}, \tag{3.6}$$

where σ controls the kernel's bandwidth, and $t\left(\sigma\right)$ is a function that maps input bandwidth to θ_ω. Generally, in order to achieve convergence with the kernel, $t\left(\sigma\right)$ should satisfy the basic condition that if

$$\sigma \to 0, t\left(\sigma\right) \to 0. \tag{3.7}$$

3.2 Photon Cone Measurement

In Section 2.2.1, we show that typical particle tracing process can be thought of as integrating incoming radiance L_i over a hypothetical sensor on scene manifold $\mathcal{M}$ with subtended incoming solid angle Ω_i

$$\mathcal{Q} = \int_{\mathcal{M}}\int_{\Omega_i} W_t\left(x_{\ell+1} \to x_\ell\right) L_i\left(x_{\ell+1} \to x_\ell\right) d\omega_i dA. \tag{3.8}$$

Under the assumption that L_o around x_ℓ and ω_o gives a good approximation of the actual radiance (or equivalently, nearby BSDFs to BSDF at x_ℓ), the sensor is restructured into one that measures outgoing quantity directly

$$\begin{aligned}\mathcal{Q} &= \int_{\Omega_o}\int_{\mathcal{M}}\int_{\Omega_i} W_t\left(x_{\ell+1}\to\psi'\right)L_i\left(x_{\ell+1}\to x'_\ell\right)d\omega_i dA d\omega_o \\ &= \int_{\Omega_o}\int_{\mathcal{M}} W'_t\left(\psi'\right)L_o\left(\psi'\right)dAd\omega_o.\end{aligned} \tag{3.9}$$

If radiance estimate on ψ is desired, W'_t can be determined in accordance with equation (2.16) and our kernel $\mathcal{K}_\sigma$ (3.6), which is simply

$$W'_t\left(\psi'\right) = \mathcal{K}_\sigma\left(\psi'-\psi\right). \tag{3.10}$$

An estimator for the measurement would then be

$$\begin{aligned}L_o\left(\psi\right) &\approx \frac{1}{M}\sum_{j=1}^{|\mathcal{P}_c|}\mathcal{K}_\sigma\left(\Psi_j-\psi\right)\frac{f\left(X_{\ell+1,j}\to\Psi_j\right)\mathcal{P}_{c,j}}{p_{\Omega_o}\left(\Psi_j\right)} \\ &= \frac{1}{M}\sum_{j=1}^{|\mathcal{P}_\Lambda|}\mathcal{K}_\sigma\left(\Psi_j-\psi\right)\mathcal{P}_{\Lambda,j} \\ &= \mathcal{L}\left(\psi,\sigma,\mathcal{P}_\Lambda\right),\end{aligned} \tag{3.11}$$

where $\mathcal{P}_\Lambda$ is what we refer to as the complete *photon cone map* here and $\mathcal{L}$ being a *photon cone radiance estimate*.

Note that to render a raster image, the ultimate goal is to compute pixel values $I_{u,v}$. This would correspond to another measurement equation located on the image plane, i.e., camera sensor. It is defined as (omitting the subscripts of pixel values as each of them has the same form)

$$I = \int_{A_{\text{sensor}}}\int_{\Omega} W_e\left(x_1\to x_0\right)L_i\left(x_1\to x_0\right)d\omega dA, \tag{3.12}$$

which can be solved using standard Monte-Carlo techniques. A corresponding estimator is written as[2]

$$I \approx \hat{I} = \frac{1}{N}\sum_{i=1}^{N}\sum_{n=1}^{\infty}\frac{F_n\left(\boldsymbol{X}_i^n\right)}{P_n\left(\boldsymbol{X}_i^n\right)}, \tag{3.13}$$

[2] For a concrete example, see appendix A.

where F_n represents the *measurement contribution function* [31, Chapter 8] for length n path and P_n is the probability density for generating the sample $\boldsymbol{X}_i^n$.

Suppose $\mathcal{P}_\Lambda$ is available, estimator (3.13) can be computed more efficiently by lookup radiance values with $\mathcal{L}\left(\psi, \sigma, \mathcal{P}_\Lambda\right)$. If the photon map is involved on the k-th vertex starting from camera, the pixel estimator can be written as

$$\hat{I} \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{n=1}^{k-1} \frac{F_n\left(\boldsymbol{X}_i^n\right)}{P_n\left(\boldsymbol{X}_i^n\right)} + \frac{1}{N} \sum_{i=1}^{N} \frac{F_k'\left(\boldsymbol{X}_i^k\right) \mathcal{L}\left(\Psi_k, \sigma, \mathcal{P}_\Lambda\right)}{P_k\left(\boldsymbol{X}_i^k\right)}, \tag{3.14}$$

where F_k' is F_k without the direct radiance sample $L_e\left(X_k \to X_{k-1}\right)$. The former part of the estimator will have convergence properties inherit from the sampling method of choice; while the later part, the *photon cone pixel estimator*, will be thoroughly analyzed in the following sections.

3.3 Properties of Photon Cone Radiance Estimate

In this section, the variance and expected error of the photon cone radiance estimate is being analyzed. Our approach is similar to the one made by Knaus and Zwicker [24], with several differences that are directly related to our formulation. The derivations here are essential for later decisions and proofs.

3.3.1 Variance of the Estimation Error

Lemma 3.1. *Let ε_σ be the estimation error produced by using the photon cone radiance estimate with a sufficiently small bandwidth σ for length k paths. Then*

$$\text{Var}[\varepsilon_\sigma] \propto \frac{1}{\sigma^2\left[1 - cos\left(t\left(\sigma\right)\right)\right]}.$$

Proof. Let $L_o\left(\psi\right)$ be the true radiance function for the estimator. Extracting particles with path length k, ε_σ can be written as

$$\begin{aligned}\varepsilon_\sigma &= \mathcal{L}\left(\psi, \sigma, \mathcal{P}_{\Lambda,k,M}\right) - L_o\left(\psi\right)\\ &= \frac{1}{M}\sum_{j=1}^{M}\mathcal{K}_\sigma\left(\Psi_j - \psi\right)\mathcal{L}_j - L_o\left(\psi\right).\end{aligned} \tag{3.15}$$

It has been shown in appendix B.1 that Var$[\varepsilon_\sigma]$ can be expressed as

$$\frac{1}{M}\left\{\mathrm{E}[\mathcal{K}_\sigma]^2\mathrm{Var}[\mathcal{L}] + \mathrm{Var}[\mathcal{K}_\sigma]\left(\mathrm{E}[\mathcal{L}]^2 + \mathrm{Var}[\mathcal{L}]^2\right)\right\}. \tag{3.16}$$

The equation can be further simplified by analyzing statistical properties related to $\mathcal{K}_\sigma\left(\Psi - \psi\right)$:

$$\begin{aligned}\mathrm{E}[\mathcal{K}_\sigma\left(\Psi - \psi\right)] &= \int_{\mathcal{S}}\mathcal{K}_\sigma\left(\psi' - \psi\right)p\left(\psi'\right)d\psi'\\ &\approx p\left(\psi\right)\int_{\mathcal{S}}\mathcal{K}_\sigma\left(\psi' - \psi\right)d\psi'\\ &= p\left(\psi\right),\end{aligned} \tag{3.17}$$

and

$$\begin{aligned}\mathrm{Var}[\mathcal{K}_\sigma\left(\Psi - \psi\right)] &= \mathrm{E}[\mathcal{K}_\sigma^2\left(\Psi - \psi\right)] - \mathrm{E}[\mathcal{K}_\sigma\left(\Psi - \psi\right)]^2\\ &\approx \int_{\mathcal{S}}\mathcal{K}_\sigma^2\left(\psi' - \psi\right)p\left(\psi'\right)d\psi' - p^2\left(\psi\right)\\ &\approx p\left(\psi\right)\int_{\mathcal{S}}\mathcal{K}_\sigma^2\left(\psi' - \psi\right)d\psi' - p^2\left(\psi\right)\\ &= p\left(\psi\right)\int_{\mathcal{S}}\left[\frac{1}{C_\sigma}\right]^2 d\psi' - p^2\left(\psi\right)\\ &= \frac{p\left(\psi\right)}{C_\sigma} - p^2\left(\psi\right).\end{aligned} \tag{3.18}$$

The analyzation is valid under the assumption that provided kernel bandwidth is sufficiently small, $p\left(\psi'\right)$ is mostly constant within the kernel support $\mathcal{S}$, i.e., $p\left(\psi'\right) \approx p\left(\psi\right)$. This is especially true since our method will reduce σ after every iteration 3.4.

Substituting (3.17) and (3.18) back into equation (3.16), we have

$$\begin{aligned}\mathrm{Var}[\varepsilon_\sigma] &\approx \frac{1}{M}\left\{p^2(\psi)\,\mathrm{Var}[\mathcal{L}] + \left[\frac{p(\psi)}{C_\sigma} - p^2(\psi)\right]\left(\mathrm{E}[\mathcal{L}]^2 + \mathrm{Var}[\mathcal{L}]\right)\right\} \\ &= \frac{1}{M}\left\{\frac{p(\psi)}{C_\sigma}\mathrm{E}[\mathcal{L}]^2 + \frac{p(\psi)}{C_\sigma}\mathrm{Var}[\mathcal{L}] - p^2(\psi)\,\mathrm{E}[\mathcal{L}]^2\right\}.\end{aligned} \tag{3.19}$$

Finally, it is possible to approximate $p(\psi)$ with kernel density estimation

$$p(\psi) \approx \hat{p}(\psi) = \frac{1}{M}\sum_{j=1}^{M}\mathcal{K}_\sigma(\psi - \psi').$$

It follows directly that as M is usually on the order of tens of millions, the density $p(\psi)$ would be much less than the kernel it builds upon. As a result, we can safely use the approximation $1/C_\sigma \gg p(\psi)$ as long as particles do not form a dense cluster, then obtain the expression

$$\begin{aligned}\mathrm{Var}[\varepsilon_\sigma] &\approx \frac{1}{M}\left\{\frac{p(\psi)}{C_\sigma}\mathrm{E}[\mathcal{L}]^2 + \frac{p(\psi)}{C_\sigma}\mathrm{Var}[\mathcal{L}]\right\} \\ &= \frac{p(\psi)}{MC_\sigma}\left[\mathrm{E}[\mathcal{L}]^2 + \mathrm{Var}[\mathcal{L}]\right] \\ &\propto \frac{1}{C_\sigma} = \frac{1}{\sigma^2\left[1 - cos\left(t(\sigma)\right)\right]}.\end{aligned} \tag{3.20}$$

■

The result can also be justified by the fact that the estimated radiance function will inherit the characteristics of the shape the kernel function has (it is essentially a weighted sum 2.3), and large σ will smooth out irregular sample distributions hence lower the variance.

3.3.2 Expected Estimation Error

Similar to 3.3.1, with ε_σ being the photon cone radiance estimation error, its expected value will be useful for later derivations.

Lemma 3.2. *The expected estimation error ε_σ has the property*

$$\mathrm{E}[\varepsilon_\sigma] \propto \sigma^2.$$

Proof. It has been shown in appendix B.2 that $\mathrm{E}[\varepsilon_\sigma]$ can be written as

$$\mathrm{E}[\mathcal{K}_\sigma\left(\Psi - \psi\right)]\mathrm{E}[\mathcal{L}] - L_o\left(\psi\right). \tag{3.21}$$

The term that is related to the bandwidth can be expanded in the way similar to equation (3.17), i.e.,

$$\mathrm{E}[\mathcal{K}_\sigma\left(\Psi - \psi\right)] = \int_{\mathcal{S}} \mathcal{K}_\sigma\left(\psi' - \psi\right) p\left(\psi'\right) d\psi', \tag{3.22}$$

except that we will instead perform Taylor expansions on the integrand in order to obtain meaningful results for $\mathrm{E}[\varepsilon_\sigma]$. Treating position and direction samples in $\mathcal{S}$ as locally independent variables[3], $p\left(\psi'\right)$ can be decomposed as a product $p_m\left(x'\right) p_d\left(\omega'\right)$ which is only valid in the vicinity of ψ. Expanding the density functions separately at (x, ω) then convolve them, we write the intermediate form as

$$\begin{aligned} &p_m\left(x'\right) p_d\left(\omega'\right) = \\ &\quad\left[p_m\left(x\right) + \left(x' - x\right)^T \cdot \nabla p_m\left(x\right) + \frac{1}{2!}\left(x' - x\right)^T H_m\left(x\right)\left(x' - x\right) + \cdots\right] \times \\ &\quad\left[p_d\left(\omega\right) + \left(\omega' - \omega\right)^T \cdot \nabla p_d\left(\omega\right) + \frac{1}{2!}\left(\omega' - \omega\right)^T H_d\left(\omega\right)\left(\omega' - \omega\right) + \cdots\right], \end{aligned} \tag{3.23}$$

where H_m and H_d are the Hessian matrices in Cartesian and spherical coordinates, respectively. After substituting the convolved density series back into equation (3.22), most terms will integrate to zero as they are odd moments of a distribution, e.g., $\int_D p_i^n \mathcal{K}\left(p\right) dp = 0$ for odd n's. A series of algebraic operation (see appendix B.3) then shows that we are left with the form $p_m\left(0\right) p_d\left(0\right) + C \cdot \sigma^2$, with C being a constant. We can now write the

[3] This is similar to the Ψ and $\mathcal{L}$ case; since $\mathcal{S}$ is comparatively small to the scale of the scene.

doi:10.6342/NTU201901709

expected estimation error as

$$\left[p_m\left(0\right)p_d\left(0\right)+C\cdot\sigma^2\right]\cdot\mathrm{E}[\mathcal{L}]-L_o\left(\psi\right). \tag{3.24}$$

Exploiting the fact that using a kernel with infinitesimal bandwidth as W_t will measure the true outgoing radiance (3.9), $L_o\left(\psi\right)$ can be similarly expressed as $\mathrm{E}[\delta\left(\Psi-\psi\right)]\mathrm{E}[\mathcal{L}]$. The expected value of a delta kernel is simply the density function sampled at the local origin, leading to the following result

$$L_o\left(\psi\right)=p_m\left(0\right)p_d\left(0\right)\cdot\mathrm{E}[\mathcal{L}]. \tag{3.25}$$

Substituting into equation (3.24), we then obtain the relation

$$\begin{aligned}\mathrm{Var}[\varepsilon_\sigma]&=\left[p_m\left(0\right)p_d\left(0\right)+C\cdot\sigma^2\right]\cdot\mathrm{E}[\mathcal{L}]-p_m\left(0\right)p_d\left(0\right)\cdot\mathrm{E}[\mathcal{L}]\\&=C\cdot\mathrm{E}[\mathcal{L}]\cdot\sigma^2\\&\propto\sigma^2.\end{aligned} \tag{3.26}$$

■

This result naturally fits our expectation since larger kernel bandwidths smooth out finer details, including sample noise caused by the stochastic process of tracing particles; and expected estimation error decreases with smaller bandwidths as it measures the radiance function in a way closer to what the original rendering equation gives.

3.4 Progressive Bandwidth Shrinkage

As most progressive density estimation rendering algorithms, our method rely on the same principle to achieve a consistent result (converges to the true energy function on the image plane); the idea is to allow the estimation variance grow by a certain amount in each iteration, while making the averaged variance and expected error over N iterations vanish. Hachisuka et al. [11] first propose the technique that satisfy these criteria on the 2-D do-

doi:10.6342/NTU201901709

main $\mathcal{M}$, which is later formally proved in [24]. In the following sections, we will show that our photon cone radiance estimate can also achieve convergence in the 4-D space $\mathcal{S} = \Omega \times \mathcal{M}$.

In the i-th iteration on the joint location ψ, the variance of the estimation error $\text{Var}[\varepsilon_{\sigma,i}]$ is allowed to grow in the next iteration in the following manner

$$\frac{\text{Var}[\varepsilon_{\sigma,i+1}]}{\text{Var}[\varepsilon_{\sigma,i}]} = \frac{\sigma_i^2\left[1 - cos\left(t\left(\sigma_i\right)\right)\right]}{\sigma_{i+1}^2\left[1 - cos\left(t\left(\sigma_{i+1}\right)\right)\right]} = \frac{i+1}{i+\alpha} \tag{3.27}$$

with Lemma 3.1 and the condition $0 < \alpha < 1$.

3.5 Proof of Convergence

In this section, we prove that the formulation of Photon Cone Focusing converges to the solution for the rendering equation. Specifically, consider the mean squared error (MSE) of the pixel estimate $\hat{I}$ (3.14)

$$MSE\left(\hat{I}\right) = \text{E}[\left(\hat{I} - I\right)^2] = \text{Var}[\hat{I}] + \left(\text{E}[\hat{I}] - I\right)^2, \tag{3.28}$$

the decomposition into variance and bias makes it clear that both terms must approach zero for a consistent result, i.e., $\text{Var}[\hat{I}] = 0$ and $\text{E}[\hat{I}] = I$ providing the number of iterations $N \to \infty$.

It has been shown in [24, Appendix C] that $\text{Var}[\hat{I}] = 0$ is true if

$$\frac{1}{N^2}\sum_{i=1}^{N} \text{Var}[\varepsilon_{\sigma,i}] = \text{Var}[\overline{\varepsilon}_{\sigma,N}] \to 0, \tag{3.29}$$

which requires the variance of the average estimation error $\text{Var}[\overline{\varepsilon}_{\sigma,N}]$ converge to zero (proved in 3.5.1). Also, the condition for $\text{E}[\hat{I}] = I$ is similarly derived in [24, Appendix D], which requires the following condition to hold, writing in our notation as

$$\frac{1}{N}\sum_{i=1}^{N} \text{E}[\varepsilon_{\sigma,i}] = \text{E}[\overline{\varepsilon}_{\sigma,N}] \to 0. \tag{3.30}$$

This condition states the prerequisites for the expected average estimation error $\mathrm{E}[\overline{\varepsilon}_{\sigma,N}]$ to converge to zero (proved in 3.5.2).

Finally, the convergence of the photon cone radiance estimate (3.14) is summarized in Section 3.5.3.

3.5.1 Variance of the Average Estimation Error

As we follow the same variance increasing scheme in most progressive density estimation literatures, the derivation is only dependent on the ratio of $\mathrm{Var}[\overline{\varepsilon}_{\sigma,i}]$ between successive iterations. It has been proved in [24, Appendix E] that $\mathrm{Var}[\overline{\varepsilon}_{\sigma,N}]$ is of the order $O\left(N^{-\alpha}\right)$, satisfying (3.29) with sufficiently large N.

3.5.2 Expected Average Estimation Error

We show that the following condition on $\mathrm{E}[\overline{\varepsilon}_{\sigma,N}]$ is true, confirming the required condition of convergence (3.30).

Lemma 3.3. *The expected average estimation error* $\mathrm{E}[\overline{\varepsilon}_{\sigma,N}]$ *converges to the order*

$$\mathrm{E}[\overline{\varepsilon}_{\sigma,N}] = O\left(N^{\frac{\alpha-1}{p+1}}\right)$$

with $p \geq 1$.

Proof. The implicit bandwidth reduction rule (3.27) has an asymptotic form (see equation B.13) under *small angle approximation*:

$$\frac{\mathrm{Var}[\varepsilon_{\sigma,i+1}]}{\mathrm{Var}[\varepsilon_{\sigma,i}]} = \frac{i+1}{i+\alpha} \approx \frac{\sigma_i^2 t^2\left(\sigma_i\right)}{\sigma_{i+1}^2 t^2\left(\sigma_{i+1}\right)}. \tag{3.31}$$

Let $t\left(\sigma\right)$ being a polynomial of order $p \geq 1$ (the lowest exponent a monomial has)[4], i.e., satisfying $t\left(\sigma\right) = O\left(\sigma\right)$ and fulfilling the other requirement[5]. The reduction rule can be

[4] Note that the highest exponent a monomial has is what we called *degree* throughout the thesis, which is the opposite of *order*

[5] The additional constraint to apply is (3.7)

expressed (asymptotically) as

$$\frac{\sigma_i^2 \sigma_i^{2p}}{\sigma_{i+1}^2 \sigma_{i+1}^{2p}} = \left(\frac{\sigma_i}{\sigma_{i+1}}\right)^{2p+2} . \tag{3.32}$$

With Lemma 3.2, it turns out that the expected average estimation error in the i-th iteration is related to the bandwidth reduction rule after some manipulations

$$\frac{\mathrm{E}[\varepsilon_{\sigma,i+1}]}{\mathrm{E}[\varepsilon_{\sigma,i}]} = \frac{\sigma_{i+1}^2}{\sigma_i^2} = \left(\frac{i+\alpha}{i+1}\right)^{\frac{1}{p+1}} . \tag{3.33}$$

We can now write the expected average estimation error in terms of the error on first iteration

$$\begin{aligned} \mathrm{E}[\varepsilon_{\sigma,i}] &= \mathrm{E}[\varepsilon_{\sigma,1}] \cdot \prod_{k=1}^{i-1} \left(\frac{k+\alpha}{k+1}\right)^{\frac{1}{p+1}} \\ &= \mathrm{E}[\varepsilon_{\sigma,1}] \left(\frac{1}{i}\right)^{\frac{1}{p+1}} \cdot \prod_{k=1}^{i-1} \left(\frac{k+\alpha}{k}\right)^{\frac{1}{p+1}} . \end{aligned} \tag{3.34}$$

Recall one of the identities of the beta function $B\left(a,b\right) = B\left(a,b\right) \cdot \left(b/\left(a+b\right)\right)$, $\mathrm{E}[\varepsilon_{\sigma,N}]$ can be expressed and rearranged as follows:

$$\begin{aligned} \mathrm{E}[\overline{\varepsilon}_{\sigma,N}] &= \frac{1}{N} \sum_{i=1}^{N} \mathrm{E}[\varepsilon_{\sigma,i}] \\ &= \frac{\mathrm{E}[\varepsilon_{\sigma,1}]}{N} \left[1 + \sum_{i=2}^{N} \left(\prod_{k=1}^{i-1} \left(\frac{k+\alpha}{k}\right)^{\frac{1}{p+1}} \right) \cdot \left(\frac{1}{i}\right)^{\frac{1}{p+1}} \right] \\ &= \frac{\mathrm{E}[\varepsilon_{\sigma,1}]}{N} \left[1 + \sum_{i=2}^{N} \left(\frac{1}{B\left(\alpha,i\right) \alpha \cdot i} \right)^{\frac{1}{p+1}} \right] \\ &= \frac{\mathrm{E}[\varepsilon_{\sigma,1}]}{N} \left[\sum_{i=1}^{N} \left(\frac{1}{B\left(\alpha,i\right) \alpha \cdot i} \right)^{\frac{1}{p+1}} \right] \\ &= \frac{\mathrm{E}[\varepsilon_{\sigma,1}]}{N \cdot \alpha^{1/(p+1)}} \left[\sum_{i=1}^{N} \left(\frac{1}{B\left(\alpha,i\right) \cdot i} \right)^{\frac{1}{p+1}} \right], \end{aligned} \tag{3.35}$$

while the last steps use the fact that $B\left(\alpha,1\right) = 1/\alpha$. Borrowing the result from [24, Appendix F], we see that

$$\sum_{i=1}^{N}\frac{1}{B\left(\alpha,i\right)\cdot i}=\frac{1}{B\left(\alpha,N\right)\cdot\alpha}+\frac{1}{B\left(\alpha,N\right)\cdot N}-1=\Theta\left(N^{\alpha}\right). \tag{3.36}$$

Here we form an inequality to bound the series in (3.35). Hölder's inequality for counting measure states that for a and b in the open interval $(1,\infty)\, with\ 1/a+1/b=1$, we have

$$\sum_{i=1}^{N}|x_iy_i|\leq\left(\sum_{i=1}^{N}|x_i|^{a}\right)^{\frac{1}{a}}\left(\sum_{i=1}^{N}|y_i|^{b}\right)^{\frac{1}{b}}. \tag{3.37}$$

Substituting the series in (3.35) as y_i, and let $x_i=1$, $a=\left(p+1\right)/p$, $b=p+1$, we have

$$\sum_{i=1}^{N}\left(\frac{1}{B\left(\alpha,i\right)\cdot i}\right)^{\frac{1}{p+1}}\leq N^{\frac{p}{p+1}}\cdot\left(\sum_{i=1}^{N}\frac{1}{B\left(\alpha,i\right)\cdot i}\right)^{\frac{1}{p+1}}. \tag{3.38}$$

Using the bound provided by (3.36), the RHS of the inequality is bounded as $N^{\frac{p}{p+1}}\cdot\Theta\left(N^{\frac{\alpha}{p+1}}\right)=\Theta\left(N^{\frac{p+\alpha}{p+1}}\right)$. This corresponds to the fact that

$$\mathrm{E}[\overline{\varepsilon}_{\sigma,N}]=\frac{\mathrm{E}[\varepsilon_{\sigma,1}]}{N\cdot\alpha^{1/(p+1)}}\cdot O\left(N^{\frac{p+\alpha}{p+1}}\right)=O\left(N^{\frac{\alpha-1}{p+1}}\right). \tag{3.39}$$

■

3.5.3 Convergence of the Pixel Estimator

We have shown that in our formulation of photon cones, Var$[\hat{I}]=0$ and E$[\hat{I}]=I$, which means the estimated pixel value $\hat{I}$ is an exact match of the true value I. Recall that Var$[\overline{\varepsilon}_{\sigma,N}]=O\left(N^{-\alpha}\right)$ and E$[\overline{\varepsilon}_{\sigma,N}]=O\left(N^{(\alpha-1)/(p+1)}\right)$, implying a larger value of α sacrifices estimation error for smaller variance, and vice versa. As specified earlier, the value of α should be limited in the open interval $(0,1)$. This is easy to see now: for $\alpha=0$, $\hat{I}$ will always have non-zero amount of variance, and for $\alpha=1$, the estimation error of $\hat{I}$ is never zero.

3.6 The PCF Method

We now give a basic pseudocode describing the PCF method (algorithm 1). As our method is based on the probabilistic derivation of the progressive photon mapping algorithm [24], the main loop can be trivially parallelized, and no statistical records for photon cones need to be stored. The method requires a *photon cone map* which is a data structure for finding nearby particle records. i.e., photon cones in the joint position-directional space ψ.

Algorithm 1: Basic Implementation of PCF

```
Input: Scene, $\alpha$, $p$, $\sigma_1$
Output: HDR Radiance Buffer ($\hat{I}$)
$\hat{I} := 0$                                        // zero the buffer
$\sigma := \sigma_1$
perform any Scene pre-processing, e.g., Section 4.2.3
while less than required number of iterations do
    /* Photon Cone Shooting Pass                                        */
    while photonConeBuffer is not full do
        hitPoints := unbaised particle tracing        // all surface types are valid
        for each hitPoint do
            photonConeBuffer += sample photon cones
    build photonConeMap given photonConeBuffer        // e.g., a kD-tree
    /* Visualization Pass                                               */
    while camera samples not exhausted do
        $L := 0$
        hitPoints := unbaised ray tracing
        index := find a suitable point in hitPoints  // e.g., delta/glossy/diffuse surface
        $L$ += estimate radiance for path lengths $\leq$ index
        $L$ += estimate radiance using photonConeMap for path lengths $>$ index
        add $L$ as a sample to $\hat{I}$
    update $\sigma$ according to equation (3.27)
```

For implementation, the angular bandwidth θ_ω can be inferred by the mapping function $(t(\sigma))$ of choice, which is assumed to be a polynomial of order $p \geq 1$. Each iteration will refine the image with more details as the kernel bandwidth shrinks. In the visualization pass, any unbiased method can be used to estimate radiance for path lengths smaller or equal to the index found. The photon cone map estimate in the next line will include all missing path lengths. The two-part estimation process corresponds to the two summations in equation (3.14). Note that it is also possible to perform nested PCF, i.e., substituting the unbiased radiance estimation stage with another photon cone radiance estimation. This will have the extra benefit of *dissolving* (eliminating) any singularities encountered, making PCF robust in terms of the number of captured light paths.

Generating particle records (photon cone samples) can be done via standard Monte-Carlo sampling. In the example shown in fig. 3.1, a corresponding photon cone sample $L_o(\Psi)$ would be calculated as

$$L_o(\Psi) = \frac{L_e(X_0 \to X_1)\cos(\Theta_0) f(X_0 \to X_1 \to X_2)\cos(\Theta_1) f(X_1 \to X_2 \to X_3)}{pdf_A(X_0)\, pdf_\omega(X_0 \to X_1)\, pdf_\omega(X_1 \to X_2)\, pdf_\omega(X_2 \to X_3)}. \tag{3.40}$$

Other path lengths follow accordingly.

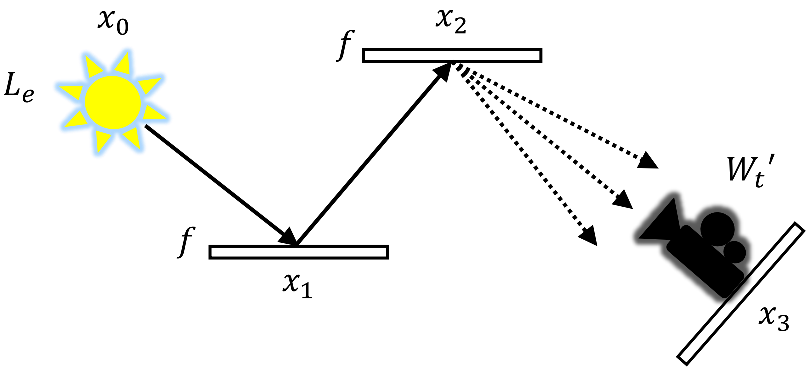


Figure 3.1: The generation of photon cone samples in the case of path length = 2. We store photon cones on x_2 (directions indicated with dotted arrows).

3.7 Generalizations

So far, our derivations are based on the conical kernel we chose in Section 3.1. In the following sections, we discuss possible generalizations of photon cone focusing.

doi:10.6342/NTU201901709

3.7.1 Arbitrary 4-D Kernels

As the kernel bandwidth shrinks, any kernel will approach a Dirac delta in $\mathbb{R}^4$, and it is possible to perform a similar derivation (described in Section B.3) to obtain the same expression for $\mathrm{E}[\mathcal{K}_\sigma(\Psi - \psi)]$. We do not explore in such direction much as we believe that kernel shape should not affect the performance of energy estimation (see how we choose the kernel in Section 3.1), and the additional cost of evaluating more complex kernel is likely not worth the small gain in terms of AMISE.

3.7.2 Participating Media

Our derivation is readily applicable to the situation where participating media is present. The kernel function would remain the same in the angular domain, while the spatial domain should be changed into a sphere, making it a 5-D function. The rest of the derivation should be the same (up to changing the spatial domain into a 3-D space), making photon cone focusing a consistent method in volumetric light transport, which is similar to what Knaus and Zwicker discussed[6] in their paper [24, Section 3].

[6] However, we suspect that Knaus and Zwicker may have made a mistake on generalizing their results to volume rendering, as we do not agree $\mathrm{E}[\epsilon(x, r)] \propto r^3$, it should remain r^2.

Chapter 4

The PCF Rendering Algorithm

We introduce a robust global illumination (GI) algorithm that eliminates all singularities in a scene, while maintaining the path reuse property of the XPPM family (Section 2.5.1). The core of the algorithm is based on photon cone focusing (Chapter 3, hence the name *PCF*) with several enhancements and specializations. Our formulation of the *photon cone map* can be seen as a standalone light field representation, which does not require fetching and evaluating material properties during radiance queries. In this chapter, we will explore the details of the PCF rendering algorithm.

This chapter bridges between theories from Chapter 3 and practical implementations. First, we introduced several guiding techniques for photon cones in Section 4.2; which includes BSDF importance sampling, direct camera connections, and a conical importon map. Later, with all these techniques, we show that it is possible to leverage all their benefits by *multiple importance sampling* in Section 4.3. Finally, possible extensions for future work are explored in the last Section 4.4. This chapter provides essential knowledge to understand high-level concepts of the algorithm.

4.1 A High-Level Overview

Our rendering pipeline is an iterative one that refines the rendered image with higher frequency details in each iteration. Inside each iteration, we use photon cone maps to record an approximation of the outgoing energy from the underlying surface. The type of light scattering event can be diffusive, glossy and specular, and can all be recorded by photon cones. Later a path sampling process starting from the camera will utilize the energy cached by photon cones to form an image (assembling different component of the lighting phenomena into a final accumulation buffer) on the camera sensor. A written pseudocode of the algorithm can be found in Section 3.6.

4.1.1 Storing and Querying Photon Cones

As the case of photon mapping where an acceleration structure is constructed over particle records (photons) for faster lookup, we need a similar data structure for photon cones. In our implementation, we use point kD-tree for storing photon cone's spatial position (3-D coordinates) and use simple linear search for the angular domain for nearest neighbor queries. This is what we called the photon cone map. The reason why we did not include the angular domain and construct a 5-D acceleration structure is that cone samples are highly coherent in the angular domain. Assuming proper importance sampling, most of the cone directions will cluster around specific angles due to BSDF importance sampling, next event estimation on camera, or importon guidance (this will be discussed in next sections), and we feel that the gain in performance is not worth the additional cost in memory and construction time. Moreover, higher dimensional trees are often less efficient due to *curse of dimensionality*.

During the camera path sampling process, we query the photon cone map and obtain a list of photon cones that are simultaneously within a certain radius in the spatial domain (σ) and a certain angle in the angular domain ($t\left(\sigma\right)$). With the ability to efficiently execute nearest photon cone queries, it is now possible to perform various operations on the list for each path vertex, including importance sampling according to cone radiance/importance.

4.1.2 Controlling Variables

Till now, we have introduced many variables during the derivation of PCF 3. A summary of selected variables that are essential for controlling the rendering algorithm is given in the following table 4.1, most of them also appear as input parameters for the renderer in

doi:10.6342/NTU201901709

our implementation.

Mathematical Symbol	Description	Range
N	Number of refinement iterations performed	≥ 1
M	Number of photon cones used in each iterations	≥ 1
S	Number of samples per pixel	≥ 1
σ_1	The initial contributing radius of each photon cone in the spatial domain	$[0, \infty)$
θ_1	The initial algular bandwidth where $t(\sigma)$ is derived from	$[0, 2\pi]$
α	Controlling the rate of shrinkage of kernel bandwidth	$(0, 1)$
p	Controlling the relative rate of shrinkage between spatial and angular domain	$[1, \infty)$

Table 4.1: Summarizing important variables for controlling the PCF rendering algorithm.

For scenes with simple and moderate complexities, using $M = 2M$, $S = 4$, $\sigma_1 \approx 0.1$, $\theta_1 \approx 18$ (degrees), $\alpha \approx 0.2$ and $p \in [1, 2]$ should be a good start. Increasing θ_1 to more than 60 degrees is not recommended as it tends to over blur angular features such as specular highlights and would require a prohibitedly large N for the rendered image to converge.

α and p play an important role in the rate of convergence of the algorithm. Smaller α gives faster overall bandwidth shrinkage and larger α gives the opposite effect. However, one should not specify overly small α values as it will result in a large variance in radiance estimation. This is hard to recover from, even with a large number of iterations, and we can afford to trade some bias for a smaller variance to obtain a cleaner image. On the other hand, p signifies a tradeoff of variance and bias between spatial and angular domain of photon cones. Larger p makes angular bandwidth shrink faster relative to spatial bandwidth, which is good for surfaces with high-frequency angular lighting phenomena (for example, mirrors). Smaller p makes spatial bandwidth shrink faster relative to angular bandwidth and can better resolve fine textures. The relation of α and p can be easily understood from the illustration (fig. 4.1).

doi:10.6342/NTU201901709

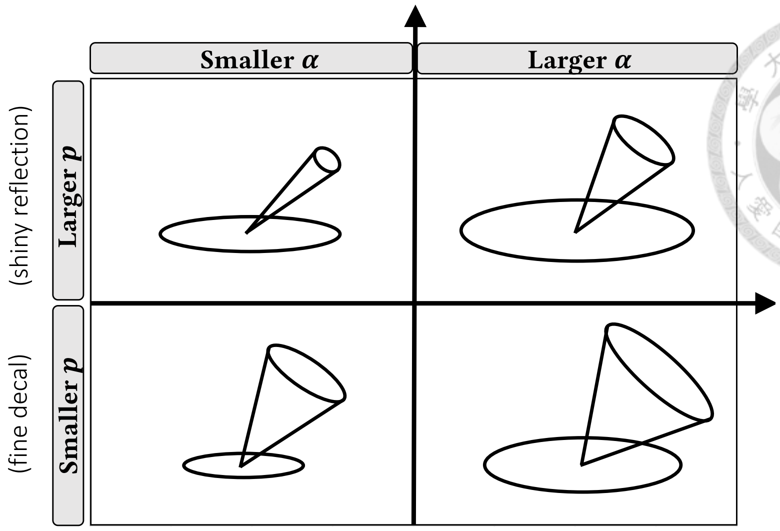


Figure 4.1: The effect of α and p on the shape of the kernel. The disk and cone represent spatial and angular bandwidth of photon cones, respectively.

4.2 Guiding Cone Samples

Although it is possible to obtain a rendered image with global illumination effects by directly visualize the photon cone map since it is essentially a light field representation, the result can be fairly noisy and may take quite a lot of time to converge due to the stochastic nature of the cone positions and directions. This can be seen in fig. 4.2.

Several sampling techniques for photon cones are proposed to greatly increase their effectiveness in representing the surface energy function. In the following sections, we introduced three kinds of sampling techniques, including BSDF importance sampling (4.2.1), camera-based next event estimation (4.2.2) and importon based importance sampling (4.2.3).

4.2.1 BSDF Importance Sampling

A typical approach is to importance sample the BSDF with or without Lambert's cosine term. The rationale behind this sampling technique is that if the BSDF has a large value on ω_o given ω_i and a position x, then the radiance function $L_o\left(x,\omega\right)$ may, too, has a large value on ω_o. As $L_o\left(x,\omega_o\right)$ is likely to have higher contribution to the final image, we should sample ω_o more comparing to other directions centered on x. The situation is especially true for surfaces with low roughnesses since their BSDF often contains concentrated peaks

doi:10.6342/NTU201901709

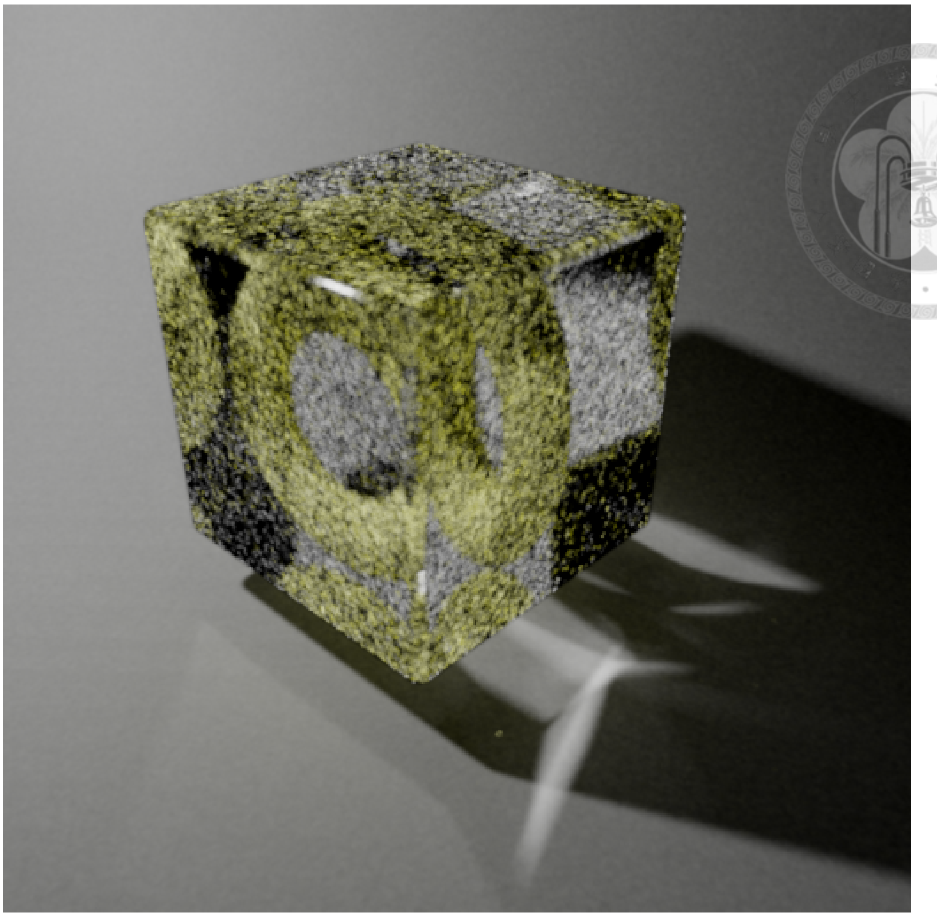


Figure 4.2: Directly visualize Hachisuka's torus scene using photon cone map, without applying any sampling techniques mentioned later. The image is refined with 5000 iterations and still contains objectionable noise, especially on the glass enclosure.

in the reflected directions. Sample directions that are slightly off-peak will likely yield only a tiny contribution even precisely aimed at the camera.

As an example, suppose we want to sample a direction from a surface material modeled with (normalized) Blinn-Phong BRDF. A corresponding probability density function for sampling the BRDF would be

$$pdf\left(\theta\right) = \frac{n+1}{2\pi}\cos^{n}\left(\theta\right), \tag{4.1}$$

where θ is the angle between surface normal and half vector (the midway direction of incident and reflection direction), and n is the specular exponent. Generating samples that conform to $pdf\left(\theta\right)$ will require two uniformly distributed random numbers ξ_1 and ξ_2

doi:10.6342/NTU201901709

in $[0, 1)$ and mapped to spherical coordinates (θ, ϕ):

$$\begin{aligned} \theta &= \arccos\left(\xi_1^{\frac{1}{n+1}}\right) \\ \phi &= 2\pi\xi_1. \end{aligned} \tag{4.2}$$

Applying this sampling strategy to all kinds of BSDF supported by our rendering system [3], we can directly visualize the photon cones that are aligned to importance-sampled BSDF directions (fig. 4.3).

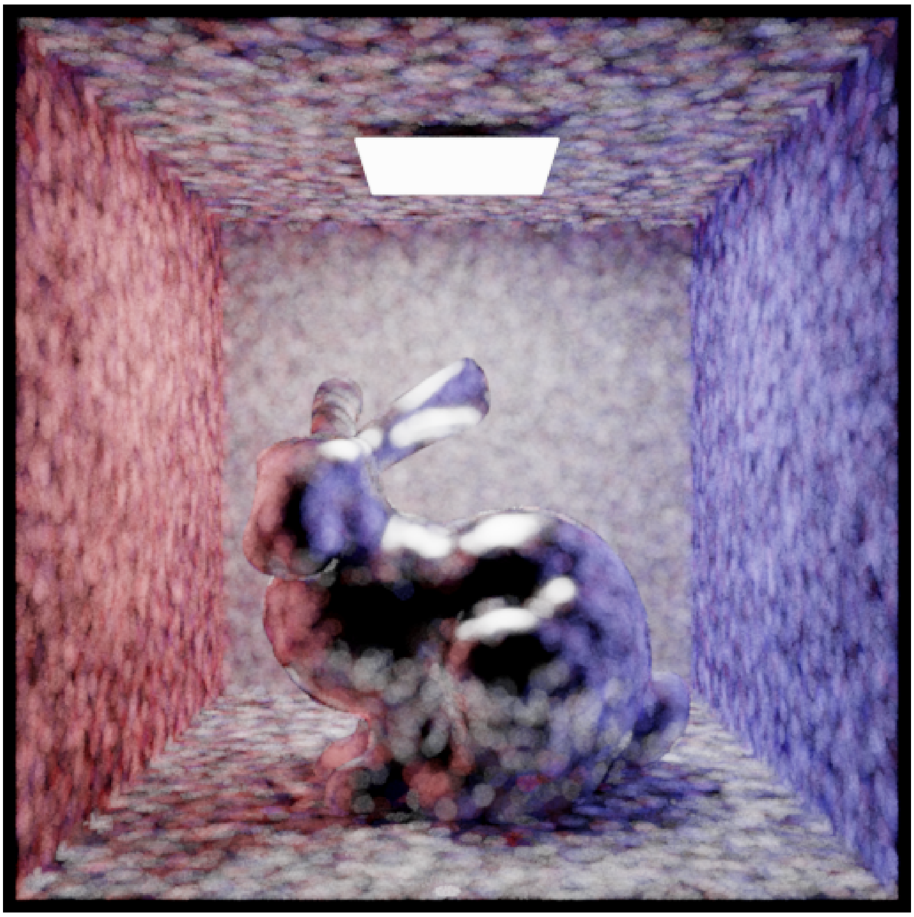


Figure 4.3: Directly visualize a cornell box scene with shiny bunny inside. The direction of photon cones are sampled based on BSDF importance sampling. We apply only 40 iterations on this image to better demonstrate the noise in rendered image.

4.2.2 Direct Camera Connections

The solution mentioned in Section 4.2.1 is however not true for rough materials. Rough material tends to have a comparatively even contribution with any set of the direction pair (ω_i, ω_o) across the unit sphere centered on x. An extreme example is the Lambertian diffuse model which has a BRDF that is view-independent. For instance, the brightness of every position will not change in a static scene with Lambertian material only, no matter the position and viewing angle of the observer. In such a situation, we should directly point the photon cones from rough materials to the camera.

Directly pointing a photon cone to camera position can be seen as a form of next event estimation (NEE), except this is done in a reversed way comparing to the canonical approach. To make camera NEE sample unbiased, we utilize a sharp distribution around $\vec{V}$, the vector pointing to camera (fig. 4.4). In our implementation, we use the aforementioned Blinn-Phong distribution (eq. (4.1)) for the sampling routine.

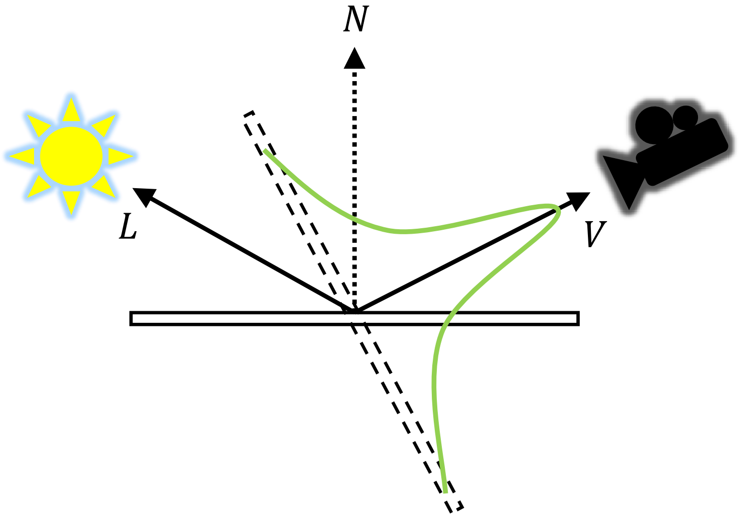


Figure 4.4: By constructing a distribution that points to the camera, we are able to importance sample the direction of camera. It is worth noting that the portion below the hemisphere defined by $\vec{N}$ will be mirrored around the origin if such sample is detected; we do so to make the camera NEE sample unbaised.

doi:10.6342/NTU201901709

On each path vertex, we are now able to sample cone directions based on two techniques: BSDF importance sampling and camera NEE. If we directly store two photon cones, each with their full energy, the resulting will be twice as bright as it should be. This is due to the fact that each technique is by itself an unbiased strategy (using either of the technique will result in a correctly converged image). An elegant solution called multiple importance sampling (MIS) is introduced by Veach, which we will talk about in Section 4.3. Weighting the photon cones from each technique appropriately using MIS, a more converged image can be obtained with the same amount of photon cones (fig. 4.5).

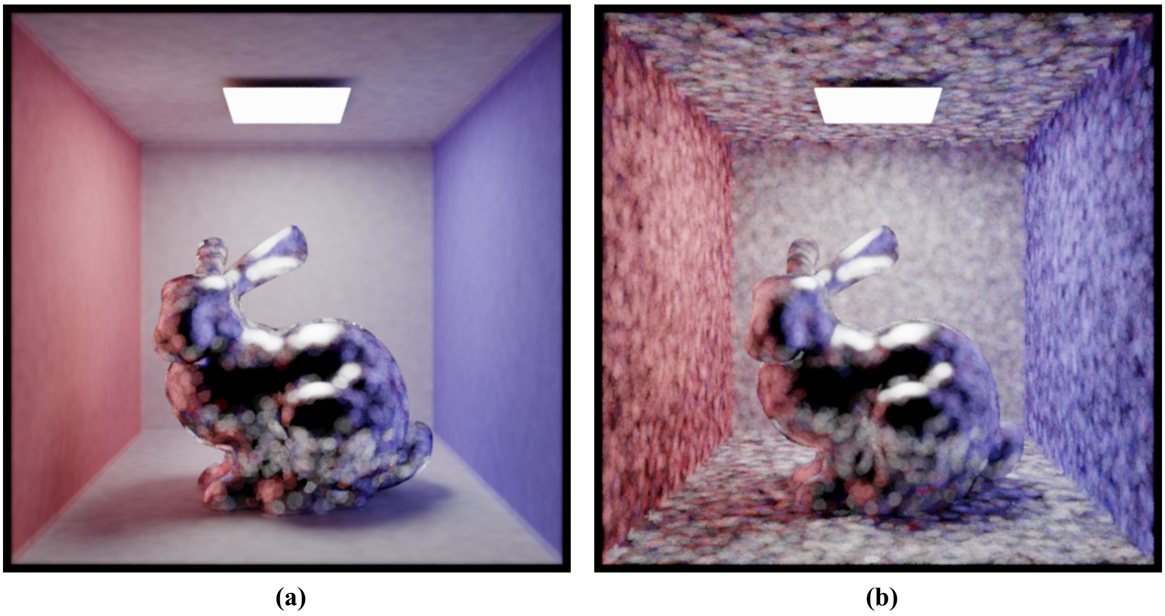


Figure 4.5: (a) Camera NEE and BSDF importance sampling combined using MIS. (b) BSDF importance sampling only. Both images use the same amount of photon cones. (40 iterations)

4.2.3 Conical Importon Map

A significant amount of energy may arrive at the image plane indirectly. Let us consider a polished metallic sphere placing on the ground. Its bottom part will reflect energy coming from the ground plane, which in turn requires one or more bounces to reach from the light source (fig. 4.6).

For shiny materials like the sphere in fig. 4.6, it would be wise to not directly sample photon cones on such surfaces for short path lengths, as normal path tracing can handle

doi:10.6342/NTU201901709

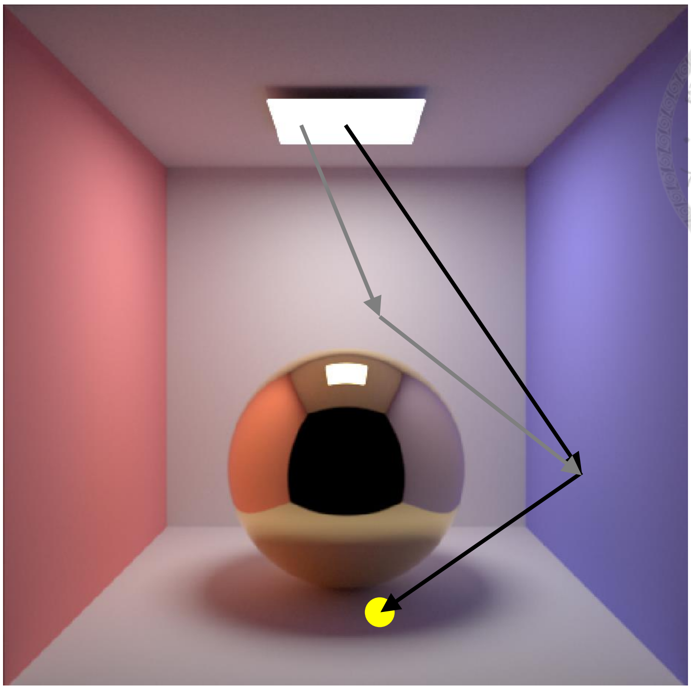


Figure 4.6: Light energy may arrive at the ground (yellow dot) directly from the blue wall (the black path), or indirectly through the white wall in the background (the gray path).

these BSDFs extremely well. We can do path tracing as usual and sample the photon cone map on the second or third bounces according to eq. (3.14). For this to work, it would be desirable to have photon cones on the ground pointing towards the reflected direction of the viewing vector, such that the view path is more likely to carry non-zero contributions; otherwise, randomly distributed photon cones is unlikely to be properly aligned and will result in larger radiance estimation variance. Neither the techniques described earlier were effective on aligning photon cones along reflected (can be one or more times) view vectors since they use only local information around the path vertex for sampling.

We can leverage the adjoint of light transport, the *camera importance* transport, to solve the problem (fig. 4.7). Tracing camera importance can be intuitively understood as emitting the value of sensor response function just like radiance except it quantifies the local importance of the camera. The adjoint of photon cones happens to be easily extensible for importance transport. The result is an acceleration structure similarly constructed as photon cones, but filled with *conical importons*. Our approach is similar to the approach by Heinrich and Werner [17], except that we do not reproject (splat) the samples to adapt

doi:10.6342/NTU201901709

our cone bandwidth according to local sample density.

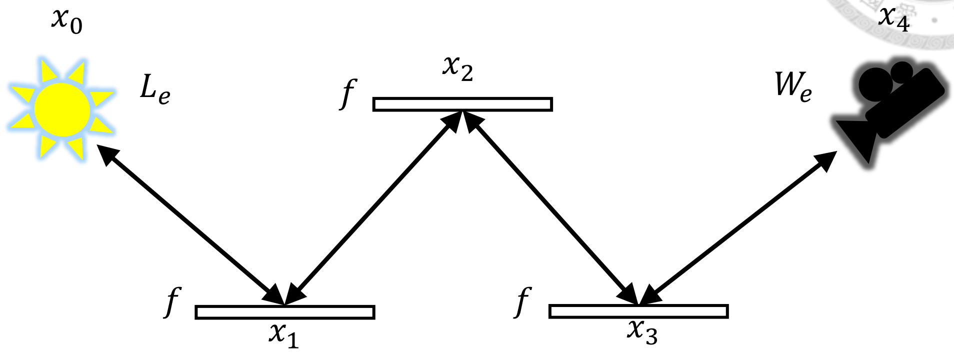


Figure 4.7: The beauty of symmetricity in light transport. Light energy (L_e) coming from x_0 to x_4 is equivalent to tracing camera importance (W_e) from x_4 to x_0.

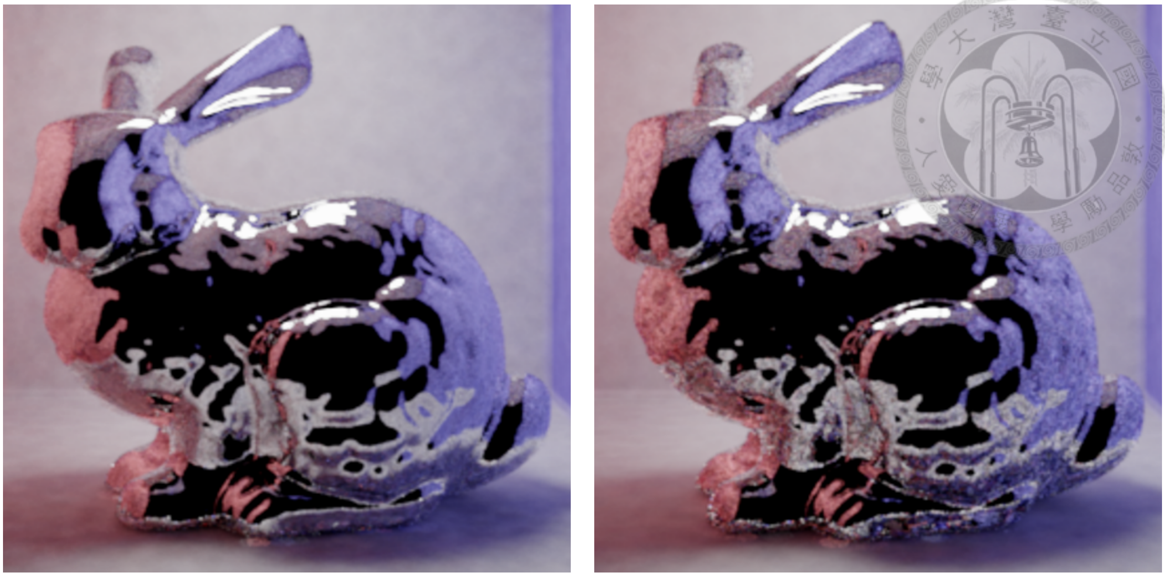
Different to photon cones, we use a single floating-point value to store camera importance in conical importons because only the magnitude of the importance is relevant to perform importance sampling given a list of nearby conical importons (by constructing a conical importon map and execute a nearest neighbor query). The result is quite effective.

4.3 Multiple Importance Sampling

We have now introduced three kinds of photon cone sampling techniques, each with its advantages. However, as already mentioned, neither of the techniques best sample the rendering equation [1]. Veach proposed a solution to combine the benefits from each of the sampling techniques called *multiple importance sampling* (MIS) [31, Chapter 9]. We will show how to combine the aforementioned techniques to dramatically increase the performance of photon cone map and the corresponding results in this section.

[1] Ideally, the conical importon technique will result in perfect importance sampling. This is only possible if we have an infinite amount of importons and a cone bandwidth close to zero. The use of a finite cone extent to reconstruct the importance function can make it inferior to other techniques on shiny surfaces and other high frequency features.

doi:10.6342/NTU201901709



(a) (b)

Figure 4.8: (a) Indirectly sample photon cone map using conical importon guided cones (with BSDF importance sampling and camera NEE). (b) Without the guidance of conical importons. Both images use the same amount of photon cones. (40 iterations)

MIS states that a new unbiased estimator $\mathcal{E}$ can be constructed by

$$\mathcal{E} = \sum_{i=1}^{n} \frac{1}{n_i} \sum_{j=1}^{n_i} w_i \left(\boldsymbol{X}_{i,j} \right) \frac{F \left(\boldsymbol{X}_{i,j} \right)}{P_i \left(\boldsymbol{X}_{i,j} \right)}, \tag{4.3}$$

where n is the number of participating techniques and w_i is a corresponding weight for each sample. In order to maintain $\mathcal{E}$'s unbiasedness, we should let $\sum_{i=1}^{n} w_i(x) = 1$ whenever $F(x) \neq 0$ and $w_i(x) = 0$ whenever $P_i(x) = 0$. We enforce MIS'ed photon cone samples to satisfy these conditions in our implementation.

4.3.1 Binary Techniques

We can combine any pair of the aforementioned techniques to form a new one that retains benefits from both techniques via MIS (see fig. 4.5 as an example). For instance, if combining BSDF importance sampling and camera NEE for cone directions is needed, a suitable weighting function must be chosen. We use the *power heuristic* proposed by

doi:10.6342/NTU201901709

Veach and Guibas

$$w_i\left(x\right) = \frac{pdf_i^{\beta}\left(x\right)}{\sum_{j=1}^{n} pdf_j^{\beta}\left(x\right)} \tag{4.4}$$

with $n = 2$ and $\beta = 2$.

4.3.2 Triplet Techniques

Combining conical *importon* based importance sampling with BSDF and camera importance sampling for cone directions is also possible. This ends up being our main sampling strategy due to their robustness in handling most lighting configurations, though our API still retain the ability to freely turn any of the MIS'ed techniques off since for some scenes this can be beneficial. A constructed conical importon map can approximate the incoming importance from the camera and we can construct a sample distribution out of it to guide photon cones. This can be useful if we lookup the photon cone map on indirect bounces (fig. 4.8).

Our implementation of triplet MIS corresponds to the weighting function 4.4 with $n = 3$ and again $\beta = 2$. In our tests, $\beta = 2$ slightly outperforms other exponents no matter the involved number of sampling techniques. Generally speaking, storing conical importons on direct hits (the first intersection found when tracing from a camera) is not necessary if camera NEE is in use while tracing photon cones; since camera NEE would be far more efficient in guiding samples toward the camera. Similarly, we do not store importon records on delta BSDFs if BSDF importance sampling is turned on.

4.4 Possible Extensions

Extending our photon cone focusing rendering algorithm should not be a difficult task, as the framework is fairly flexible. For instance, optimizing parameter choosing with ray differentials can be effective since eye path is traced just like normal path tracing, and σ_1 can be determined accordingly. It is also possible to independently determine an initial bandwidth for each path length. The properties of surface material can be taken into account when deciding θ_1, e.g., using smaller θ_1 values for glossy BSDFs and larger ones for rough/diffusive BSDFs.

Also, in order to handle impossible-to-sample paths for XPPM methods, PCF can serve as a plug-in method for LSSSE paths and others. A more difficult extension would be generalizing PCF to work with VCM (as another sampling technique in addition to PPM and BDPT).

doi:10.6342/NTU201901709

doi:10.6342/NTU201901709

Chapter 5

Robust Light Transport with Photon Cones

We demonstrate image rendered with the photon cone focusing (PCF) algorithm in this chapter. PCF is a rendering algorithm that can handle complex lighting phenomena and scene setups that are considered ”hard” by (local) path sampling transport algorithms. In Section 5.1.1, we provide a brief overview of Veach’s path regular expression and points out several shortcomings of path sampling and photon mapping methods. We analyze our method quantitatively and qualitatively in Section 5.3.1 and Section 5.3.2, respectively. Finally, the limitations of our method are discussed in Section 5.3.3.

5.1 Characteristics

The PCF rendering algorithm eliminates all possible singularities in a scene representation due to photon cone’s non-zero bandwidth in both spatial and angular dimensions. Any path regular expressions 5.1.1 can be captured by our method. In the following sections, we use Veach’s path regular expressions to better understand PCF’s capabilities.

doi:10.6342/NTU201901709

5.1.1 Path Regular Expression

Veach's *full-path regular expression* [31, Chapter 8] is an extended Heckbert notation for describing path properties including emitters and sensors. The endpoints of a path are capped with the letter L and E, where the former denotes an emitter and the later denotes a sensor. Two extra letters are introduced for denoting diffuse/glossy scattering events (D) and specular events (S). Emitters and sensors are also appended with two additional letters to represent their positional and directional properties. For instance, we can write LDD for diffusive area lights, LSD for point lights, and LDS for directional lights. Sensors are similarly expressed as, for example, DDE for canonical cameras, DSE for pinhole cameras, and SDE for cameras with orthographic projection. The events in-between the emitter and the sensor are denoted as Heckbert's original notation. As a result, $LDD\,(D|S)^*\,DSE$ is an expression for the set of all possible paths generated from a diffusive area light through scene geometries and landed on a pinhole camera.

Limitations of Path Sampling

There exist paths that are not samplable by (local) path sampling methods, and such scenario can be quite common in day-to-day scenes. An example is a framed drawing (D) with glass cover (S) illuminated by LEDs (LSD). The drawing has a rough surface that scatters incoming energy went from LEDs and through the glass cover. However, problems occurred as scattered energy refracted through the glass cover and tries to connect to a pinhole camera (DSE); since it has zero probability to randomly have a refracted scattering direction that happens to go through an infinitesimal aperture. The same issue persists if importance is being transported. The path between the camera and the LEDs has the expression SDS, and the full expression $LSDS\,(DS)^+\,DSE$ for all bounces in the frame is impossible to sample with any *(*local sampling) methods. Any sharp surface BSDF illuminated by small light sources can make path tracing (including BDPT) to fail completely (path connections do not align with a BSDF's spike). The problem is shown in fig. 5.1. Note that there exist other geometry configurations that cause the same problem, and different light transport algorithms can introduce their own set of unsamplable

doi:10.6342/NTU201901709

paths.

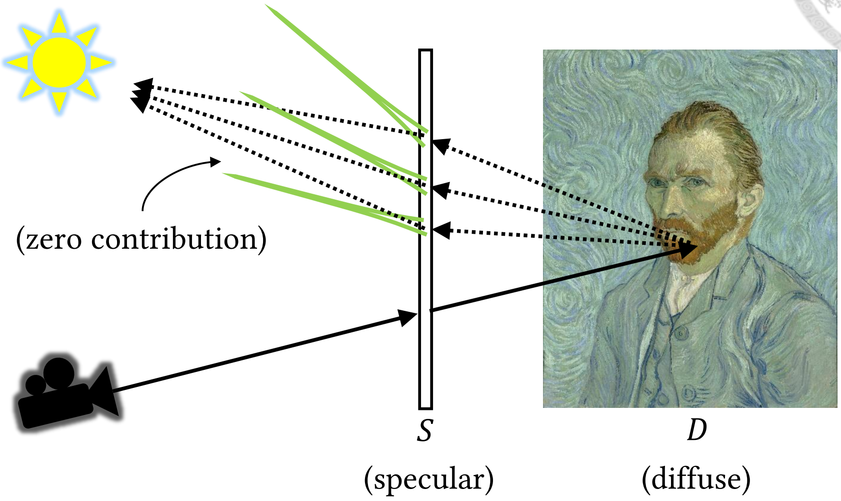


Figure 5.1: The problem of SDS paths makes Vincent van Gogh's painting placed behind a glass plane unrenderable. A BSDF that is sharp enough (green lines) can easily make NEE to fail.

Limitations of Photon Mapping

Though photon mapping methods partially solve the problem of SDS paths by merging nearby path vertices (fig. 5.2), making the painting property illuminated by potentially small light sources through a glass plane with specular BSDF. However, not all the paths are properly handled and some lighting features are still missing. To elaborate, reflections on the glass plane will be completely black since traditional photon maps cannot estimate radiance on specular surfaces (and being extremely inefficient on a nearly specular one). The teaser figure in [23] clearly shows the issue in a real-world scenario where the reflection of a LED panel is completely missing in path tracing and photon mapping methods. Similar problems exist on complex luminaires such as automotive tail lights [25] and chandeliers [33].

Our explanation on such issues can be derived from the measurement interpretation in Section 2.2. Note that we can interpret photon mapping as constructing a hypothetical

doi:10.6342/NTU201901709

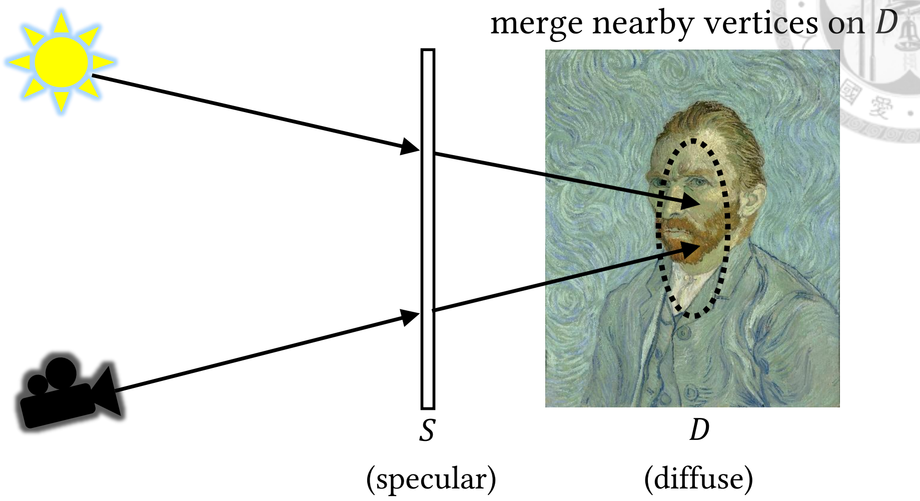


Figure 5.2: Non-zero contribution can be obtained by merging nearby (region enclosed by dots) path vertices on diffusive surfaces.

sensor on the sceme manifold $\mathcal{M}$, where we can measure any quantity using the sensor response function

$$W_t\left(x_{\ell+1} \rightarrow x'_\ell\right) = \mathcal{K}\left(x'_\ell - x_\ell\right) f\left(x_{\ell+1} \rightarrow x'_\ell \rightarrow x_{\ell-1}\right) \cos\left(\theta_i\right). \tag{5.1}$$

Inside W_t, surface BSDF f is being included as part of the sensor response function. Even though $\mathcal{K}(x)$ is a kernel with finite support, the introduction of f makes W_t once again prone to unsamplable delta distributions. For example, on the surface of mirrors, we have the following BRDF

$$f\left(x_{\ell+1} \rightarrow x_\ell \rightarrow x_{\ell-1}\right) = \frac{\delta\left(\left(x_\ell \rightarrow x_{\ell-1}\right) - \text{reflect}\left(x_{\ell+1} \rightarrow x_\ell\right)\right)}{\cos\left(\theta_i\right)}, \tag{5.2}$$

which contains a delta function that is unable to resolve by $\mathcal{K}(x)$ alone.

5.2 Dissolving Singularities

Our formulation of photon cones removes singularities in both spatial and angular domains. We call this singularity *dissolving* as their removal is done with a blurring effect applied. Photon cones can be useful in capturing difficult lighting configurations since sampling a photon cone map not only implies the reuse of nearby paths, it can be done on all kinds of BSDF including delta distributions. In fig. 5.3, the problem of SDS paths is illustrated in a concise way, where connection from x_2 to x_1 and x_3 must always fail for local path sampling methods. Vertex merging on either x_1 or x_3 is not feasible as well. Our formulation ensures that sampling photon cone maps can succeed in any type of BSDFs.

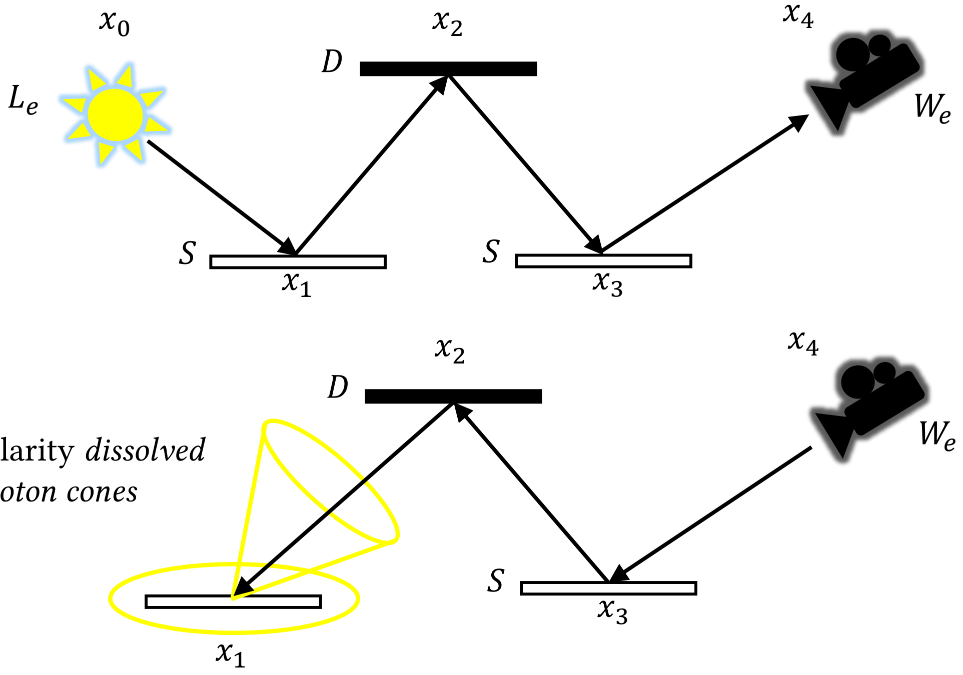


Figure 5.3: Our formulation of photon cones ensures successful vertex merging on any type of surface material (bottom). Connecting and merging vertices on either x_1 or x_3 is impossible for traditional approaches (top).

5.3 Comparisons

In the following sections, quantitative and qualitative analyses of the rendered images produced by our algorithm are made.

doi:10.6342/NTU201901709

5.3.1 Quantitative Analyses

In Section 3.5, we concluded that PCF is consistent, i.e., converges to the correct result in the limit. We now verify this property by performing several experiments that examine the error produced by our method as it approaches the ground truth (see Figure 5.4 through 5.7).

For the following scenes, we thank professor Hachisuka for providing the torus and box scene. In Figure 5.4 we marked three (color-coded) probe locations that are representative to the scene (A: on the dim silhouette of glass reflection; B: a bright spot in the caustics; C: on the surface of the enclosed torus). Figure 5.5 shows that our algorithm correctly estimate surface radiance as the result converges to actual value with the shrinking spatial and angular bandwidths of the kernel.

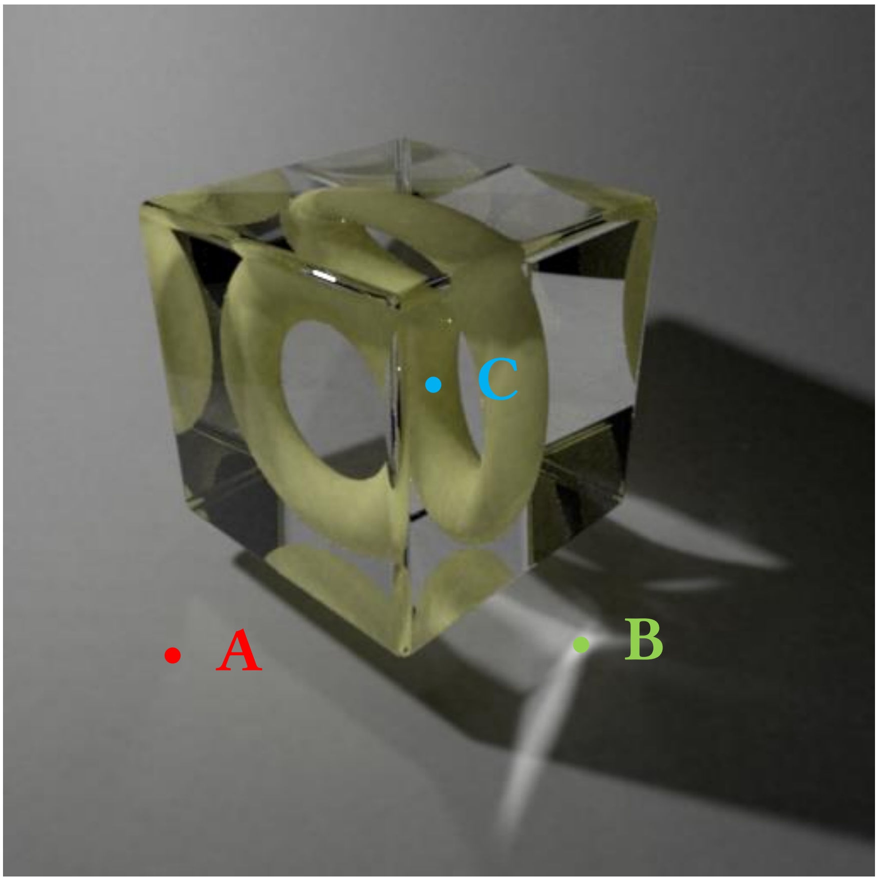


Figure 5.4: Hachisuka's torus scene rendered with PCF. Three color-coded locations are analyzed in Figure 5.5. (4000 iterations with $\sigma_1 = 0.1$, 1M cones per iteration)

Similarly, we show in Figure 5.6 through Figure 5.7 that our algorithm is consistent

doi:10.6342/NTU201901709

in the limit (A: on the caustics; B: on the reflected caustics; C: on a surface dominated by indirect lighting).

Note in both tests, we use $S = 2$.

5.3.2 Qualitative Analyses

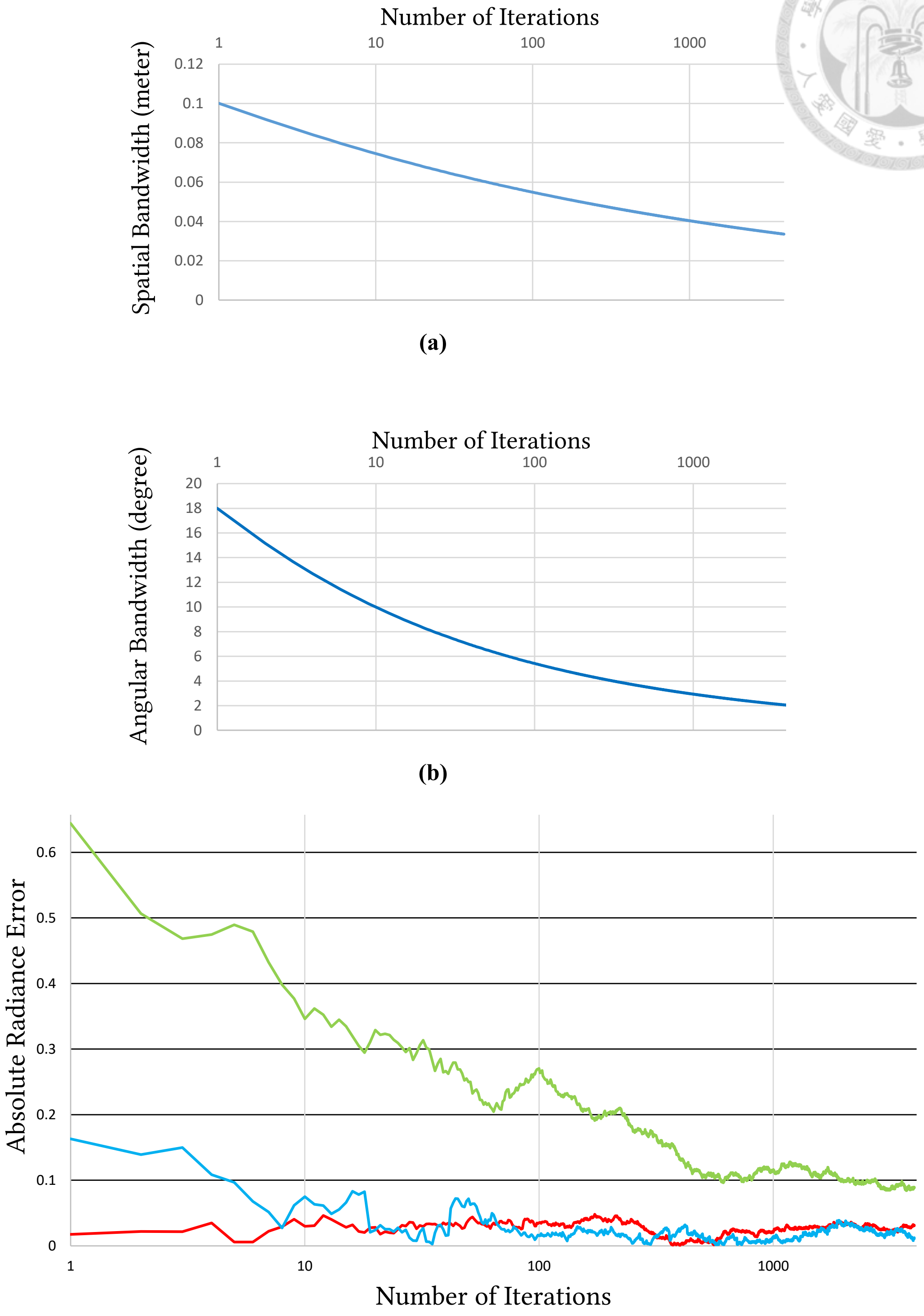
We show that our method has superior convergence rate in terms of the smoothness of lighting and captured light paths, with a scene that is dominated by shiny materials (with a GGX roughness close to 0). See Figure 5.8 to understand the arrangement of the geometries in the scene. A series of side-by-side comparisons are given (see Figure 5.9 to 5.13). The image on the left is ground truth rendered by path tracing using one million samples per pixel (with *next event estimation*, NEE, on). All images on the right are rendered with an Intel Core i7-6800k processor (6C12T) with 10 threads.

From the results, we can see our method robustly captures reflections on both the pyramids and the bunny. With less rendering time, PCF can resolve most lighting features. As the scene is full of shiny surfaces, BDPT (in Figure 5.10) and SPPM (in Figure 5.13) are essentially degenerated to unidirectional path tracing and incurs large sample variances that manifest themselves as noises/fireflies.

5.3.3 Limitations

Our photon cone radiance estimate introduces boundary biases in the angular domain in addition to the spatial domain. This can result in light leaking and inaccurate highlight in some situations. The rate of convergence is also slower since the kernel has more dimensions than in canonical density estimation methods. In Figure 5.14, we subtract a ground truth image of Hachisuka's box scene with an image rendered by PCF. It clearly shows the additional form of biases produced by our method (note the boundary between adjacent geometries).

doi:10.6342/NTU201901709



(a)

(b)

(c)

Figure 5.5: Chart showing that the error induced by photon cone radiance estimation approaches zero. (a) (b) Recorded bandwidth shrinkage information during the experiment. (c) Absolute error on the color-coded locations in Figure 5.4.

doi:10.6342/NTU201901709

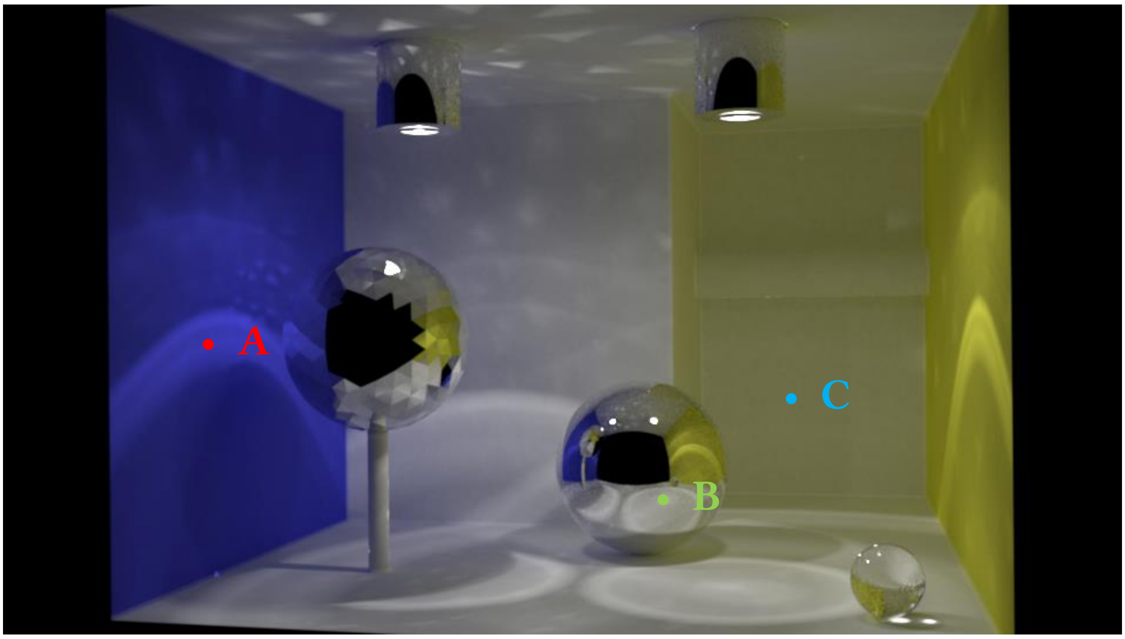
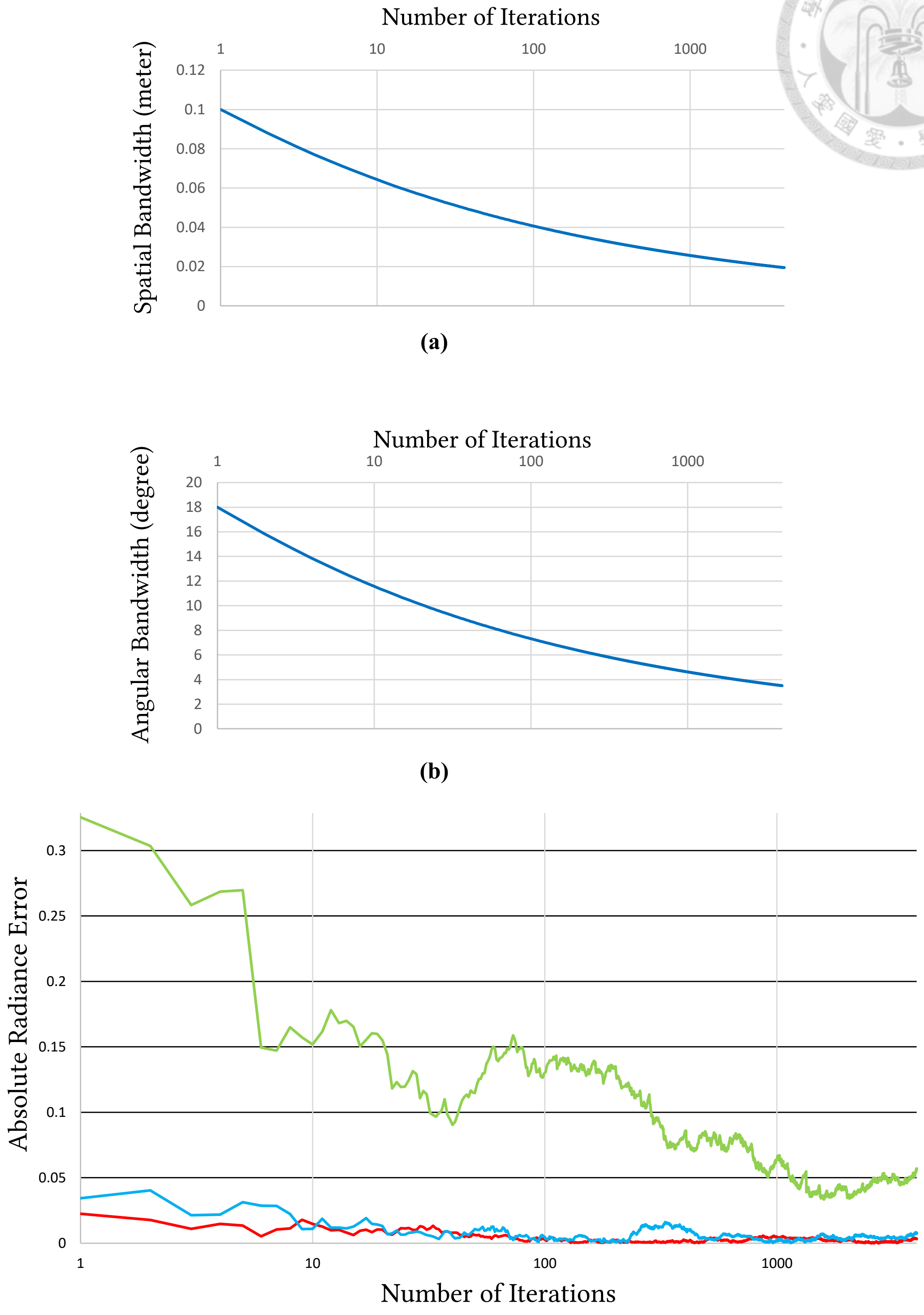


Figure 5.6: Hachisuka's box scene rendered with PCF. Three color-coded locations are analyzed in Figure 5.7. (4000 iterations with $\sigma_1 = 0.1$, 4M cones per iteration)

doi:10.6342/NTU201901709



(a)

(b)

(c)

Figure 5.7: Chart showing that the error induced by photon cone radiance estimation approaches zero. (a) (b) Recorded bandwidth shrinkage information during the experiment. (c) Absolute error on the color-coded locations in Figure 5.6.

doi:10.6342/NTU201901709

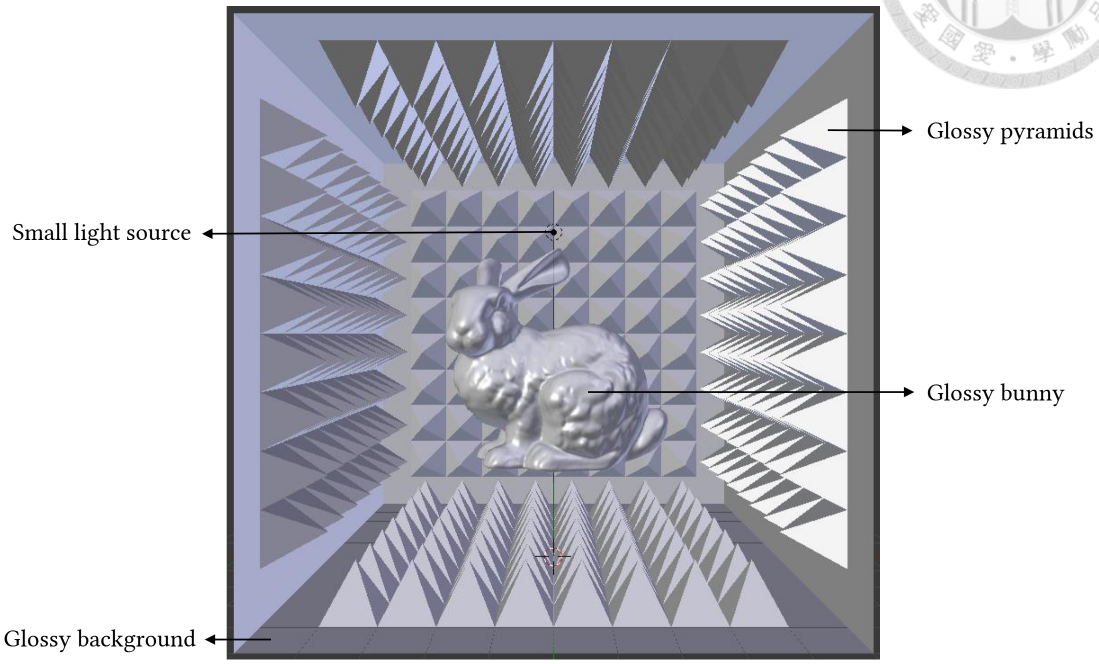
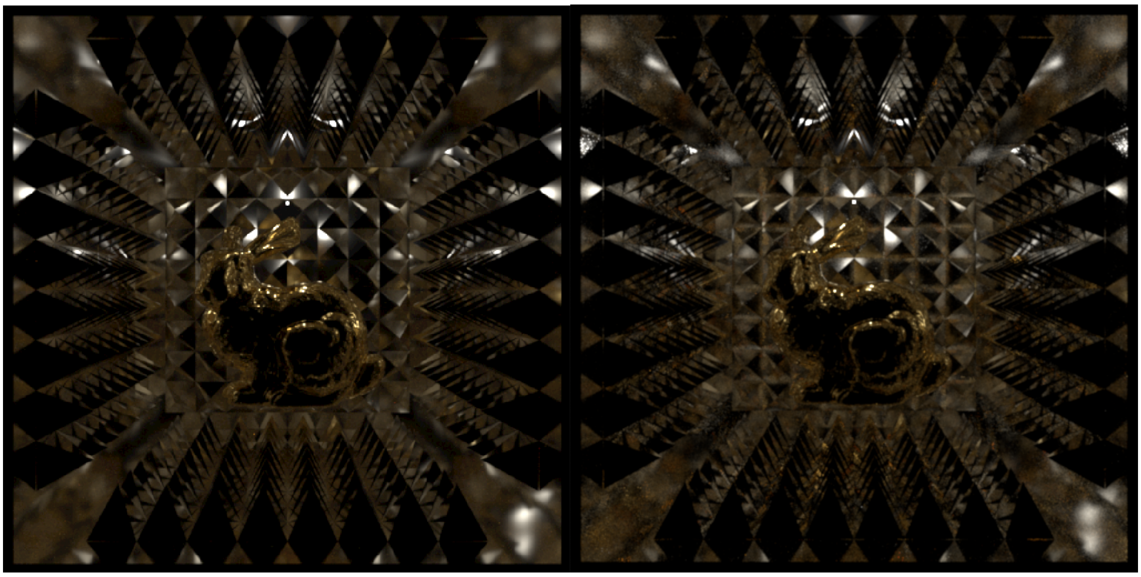


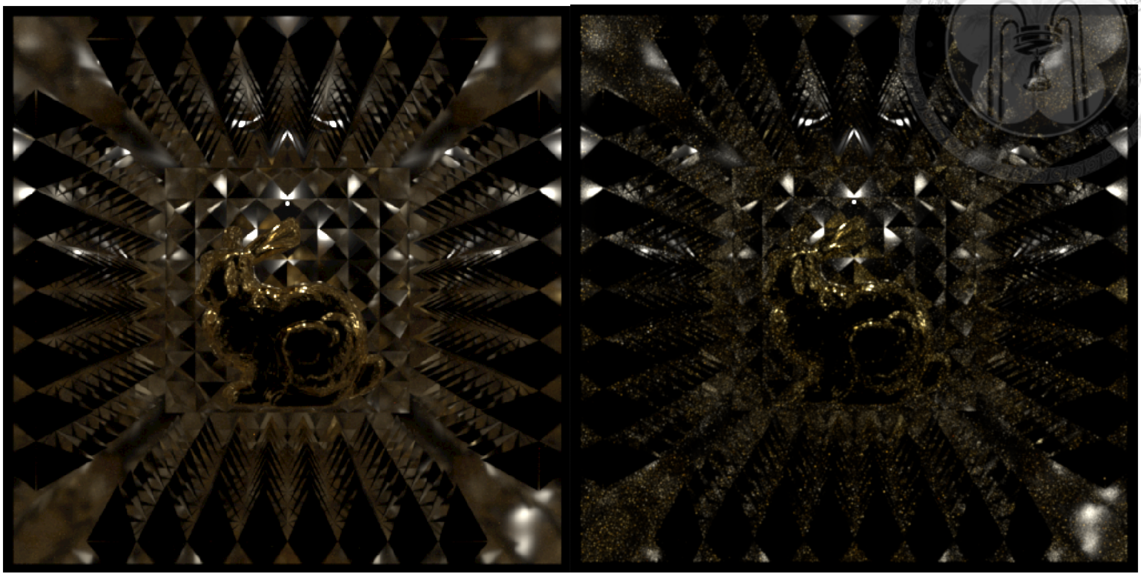
Figure 5.8: The scene that we perform our qualitative analyses on.



(a) PTNEE, spp: 1M

(b) PCF, iteration: 100, photon cones: 1M/iteration, 13.4 min.

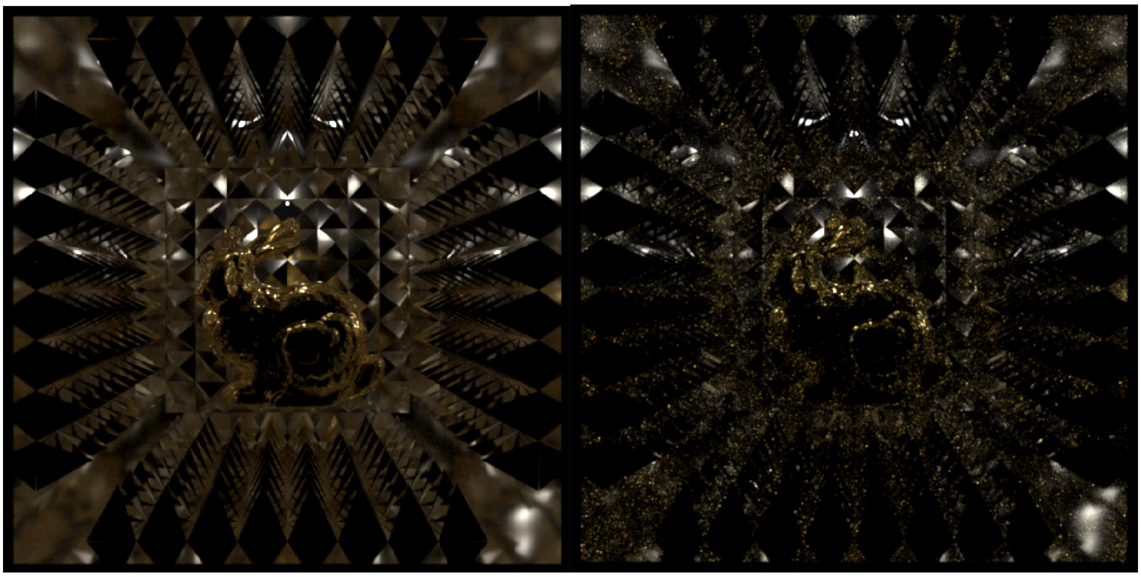
Figure 5.9: Comparison between ground truth and PCF (ours).



(a) PTNEE, spp: 1M

(b) BDPT, spp: 512, 15.2 min.

Figure 5.10: Comparison between ground truth and BDPT.

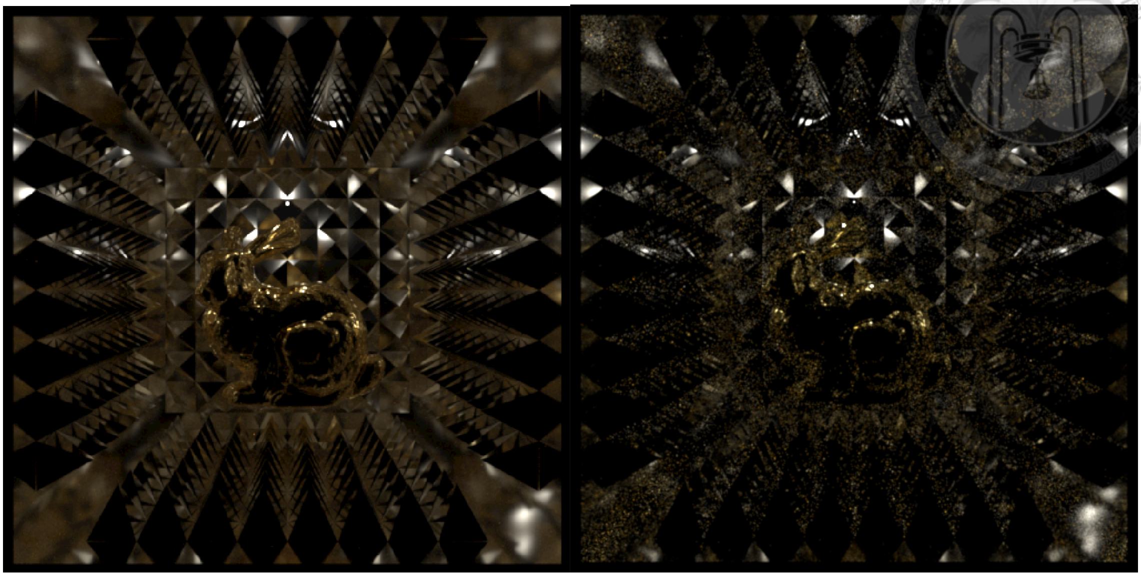


(a) PTNEE, spp: 1M

(b) PSSMLT, spp: 512, 19.3 min.

Figure 5.11: Comparison between ground truth and PSSMLT.

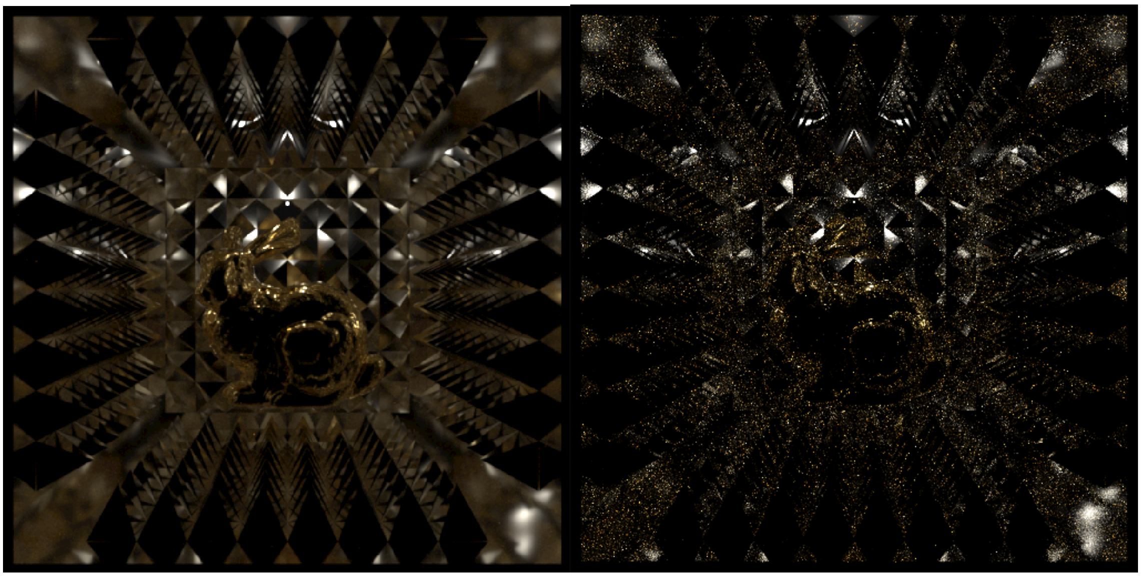
doi:10.6342/NTU201901709



(a) PTNEE, spp: 1M

(b) ERPT, spp: 512, 23.9 min.

Figure 5.12: Comparison between ground truth and ERPT.



(a) PTNEE, spp: 1M

(b) SPPM, iteration: 512, photons: 200k/iteration, 17 min.

Figure 5.13: Comparison between ground truth and SPPM.

doi:10.6342/NTU201901709

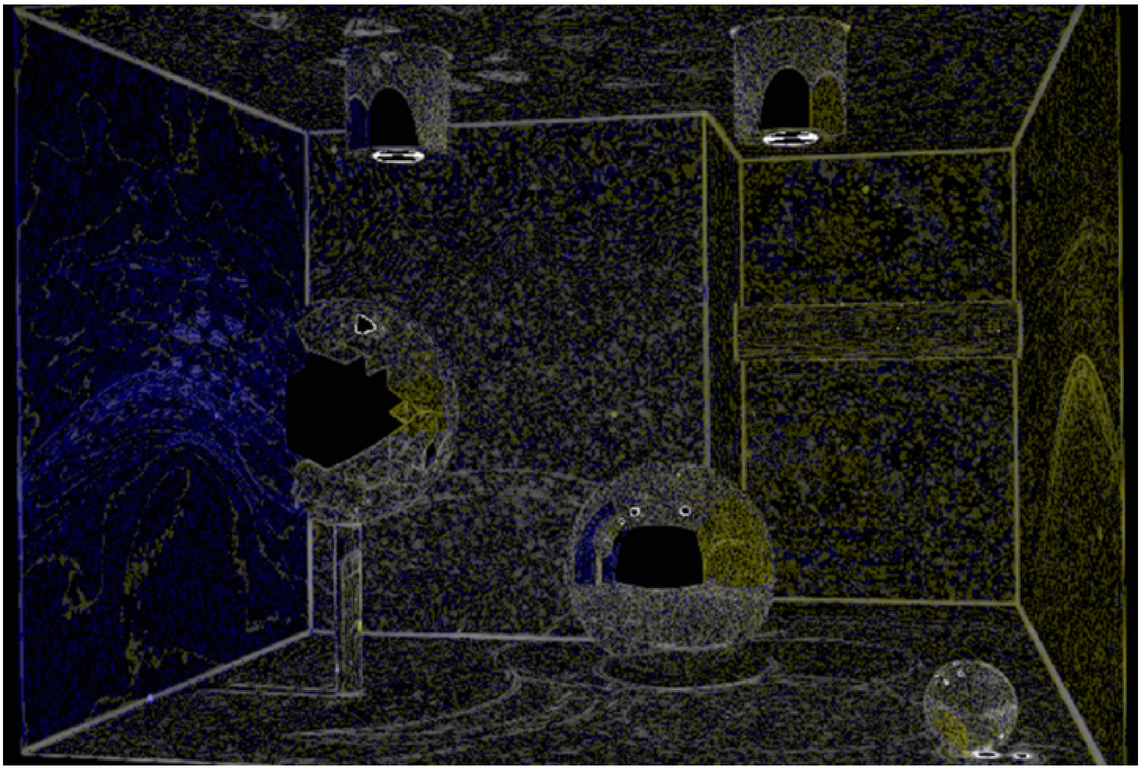


Figure 5.14: Image difference of ground truth (SPPM) and PCF. In total, 200G photons are used for SPPM and 16G photon cones for PCF. (contrast adjusted to better show the difference)

doi:10.6342/NTU201901709

Chapter 6

Conclusions

We conclude by providing a more in-depth summary than the ones given in Chapter 1. Major results derived from this thesis are reviewed and potential improvements, extensions, and future work are explored, if any.

In Chapter 2, we review photon mapping methods with two popular interpretations: energy packet and measurement. Detailed explanations on how photon mapping is formulated are given, building a solid foundation for understanding advanced density estimation algorithms and our proposed *photon cone focusing* rendering algorithm. We mathematically proved in Chapter 3 that estimating any quantity with photon cones will lead to a consistent result as the case of progressive photon mapping. Chapter 4 provides enhancements on our algorithms, including multiple importance sampling with three different techniques for photon cones. Notably, we use the adjoint of photon cones, conical importons, to guide cone samples. In Chapter 5, rendered results are given and comparisons to other rendering algorithm are done. We also showed that our method indeed converges to ground truth by experimental means in addition to mathematical proof.

In short, this thesis describes a new global illumination method that achieves:

- Vertex merge on any BSDF types
- Handles difficult lighting phenomena
- Does not require material evaluation on the visualization pass
- Converges to the ground truth

6.1 Future Work

We already mentioned several possibilities on extending PCF in Section 4.4. An interesting idea would be unifying PCF as an additional sampling technique to the VCM method. Bouchard et al. [2] introduced the groundwork for combining bidirectional path tracing, photon mapping, and directional regularization. Our method can be seen as directional regularization with path reuse and has the potential to be included as the fourth technique for MIS.

Kaplanyan and Dachsbacher also mentioned that dropping a photon map after each iteration can be inefficient as they can be potentially reused for lower variance in the final section of their paper [22]. We did try to alternate between different values of samples per pixel in each iteration of our algorithm (the S parameter), and noticed that using $S = 1$ rarely produce an image with the lowest variance; usually a value of $S = 2$ to 4 works the best. Trying to derive the optimal S can also be a direction worth investing.

doi:10.6342/NTU201901709

Appendix A

Path Tracing

Path tracing is a rendering algorithm based on Monte-Carlo integration. It is comparatively simple to implement than other global illumination methods, handles a wide variety of light paths in a unified framework, and has predictable convergence on most scenes. Apart from the scene description, path tracing requires only an acceleration structure built upon scene geometries, which makes it a good *plug-in* method for rendering specific light paths; for example, using path tracing to render direct lighting and photon mapping for the indirect bounces. In this appendix, we give a brief overview of the theoretical basis of the path tracing method.

A.1 Radiance and Throughput

The rendering equation (or light transport equation, LTE) proposed by Kajiya et al. [21] can be written as follows

$$L_o\left(x, w_o\right) = L_e\left(x, w_o\right) + \int_{\Omega} L_i\left(x, w_i\right) f\left(x, w_i, w_o\right) \cos\theta_i dw_i, \tag{A.1}$$

where L_i and L_o are the incident and outgoing radiance, respectively. The integral also involves a quantity $f\left(x, w_i, w_o\right)$ commonly known as BSDF (bi-directional scattering distribution function). In the case of multiple bounces, we should expand L_i into another integral with respect to point x' (the point ω_i points to) and let $L_i\left(x, w_i\right) = L_o\left(x', w_o\right)$. This

doi:10.6342/NTU201901709

is a recursive process. Recursively expand the LTE and using x_i with the combination of an arrow to conveniently represent both position and direction, we get

$$\begin{aligned}
&L_o\left(x_1 \rightarrow x_0\right) = L_e\left(x_1 \rightarrow x_0\right) \\
&\quad + \int_{\Omega_1} L_e\left(x_2 \rightarrow x_1\right) f\left(x_2 \rightarrow x_1 \rightarrow x_0\right) \cos\left(\theta_1\right) dw_1 \\
&\quad + \int_{\Omega_1} \int_{\Omega_2} L_e\left(x_3 \rightarrow x_2\right) f\left(x_3 \rightarrow x_2 \rightarrow x_1\right) \cos\left(\theta_2\right) f\left(x_2 \rightarrow x_1 \rightarrow x_0\right) \cos\left(\theta_1\right) dw_2 dw_1 \\
&\quad + \cdots .
\end{aligned} \tag{A.2}$$

The expanded LTE (A.2) is equivalent to the path integral formulation introduced by Veach [31, Section 8.2], which is an infinite-dimensional integral. It states that the outgoing radiance with the direction $x_1 \rightarrow x_0$ from point x_1 is the amount of emitted radiance from x_1 plus the amount that bounced once, and plus the amount that bounced twice, and so on. The integral for n bounces ($n \geq 1$) can be written compactly as

$$\begin{aligned}
&L_o\left(x_1 \rightarrow x_0\right) = L_e\left(x_1 \rightarrow x_0\right) \\
&\quad + \sum_{n=1}^{\infty} \left\{ \int_{\Omega^n} L_e\left(x_{n+1} \rightarrow x_n\right) \cdot \prod_{i=1}^{n} \Big[f\left(x_{i+1} \rightarrow x_i \rightarrow x_{i-1}\right) \cos\left(\theta_i\right) \Big] d\omega^n \right\}
\end{aligned} \tag{A.3}$$

and can be solved using the Monte-Carlo estimator

$$L_o\left(x_1 \rightarrow x_o\right) \approx L_e\left(x_1 \rightarrow x_o\right) + \frac{1}{M} \sum_{j=1}^{M} \sum_{n=1}^{\infty} \frac{L_e\left(X_{j,n+1} \rightarrow X_{j,n}\right) \cdot T\left(\boldsymbol{X}_j^n\right)}{p\left(\boldsymbol{X}_j^n\right)}, \tag{A.4}$$

where $p\left(X_j^n\right)$ is the probability density of sampling the length n path and $T\left(X_j^n\right)$ is the path throughput

$$T\left(\boldsymbol{x}^n\right) = \prod_{i=1}^{n} \left(f\left(x_{i+1} \rightarrow x_i \rightarrow x_{i-1}\right) \cos\left(\theta_i\right)\right). \tag{A.5}$$

A.2 The Measurement Equation

Note that equation (A.3) represents only the radiance from x_1 to x_0. In case of rendering, normally a camera is placed in the scene and we would like to record the energy every sensor takes to form an image. The measurement equation [31] describes the situation where an arbitrary quantity Ψ is being measured by some sensor with a finite area

$$\Psi = \int_{A_{\text{sensor}}} \int_{\Omega} W_e \left(x_1 \rightarrow x_0\right) L_i \left(x_1 \rightarrow x_0\right) d\omega dA. \tag{A.6}$$

W_e can be thought of as the sensor response function. Ideally, the sensor measures radiance and W_e should be defined accordingly.

doi:10.6342/NTU201901709

Appendix B

Supplemental Derivations for PCF

B.1 Expressing Variance of the Estimation Error

We start from the equation (3.15) that represents the estimation error:

$$\varepsilon_\sigma = \frac{1}{M}\sum_{j=1}^{M}\mathcal{K}_\sigma\left(\Psi_j - \psi\right)\mathcal{L}_j - L_o\left(\psi\right). \tag{B.1}$$

By treating particle locations Ψ_j and weights $\mathcal{L}_j$ as independent and identically distributed (i.i.d.) random variables, and let $p\left(\psi\right)$ be the probability density for Ψ, we can derive the variance of the radiance estimate with common identities

$$\begin{aligned}
\mathrm{Var}[\varepsilon_\sigma] &= \mathrm{Var}\left[\frac{1}{M}\sum_{j=1}^{M}\mathcal{K}_\sigma\left(\Psi - \psi\right)\mathcal{L} - L_o\left(\psi\right)\right] \\
&= \mathrm{Var}\left[\frac{1}{M}\sum_{j=1}^{M}\mathcal{K}_\sigma\left(\Psi - \psi\right)\mathcal{L}\right] \\
&= \frac{1}{M^2}\mathrm{Var}\left[\sum_{j=1}^{M}\mathcal{K}_\sigma\left(\Psi - \psi\right)\mathcal{L}\right] \\
&= \frac{1}{M^2}\sum_{j=1}^{M}\mathrm{Var}[\mathcal{K}_\sigma\left(\Psi - \psi\right)\mathcal{L}] \\
&= \frac{1}{M}\mathrm{Var}[\mathcal{K}_\sigma\left(\Psi - \psi\right)\mathcal{L}].
\end{aligned} \tag{B.2}$$

Recall the identity[1]

$$\mathrm{Var}[XY] = \mathrm{E}[X]^2\mathrm{Var}[Y] + \mathrm{Var}[X]\left(\mathrm{E}[Y]^2 + \mathrm{Var}[Y]\right),$$

and let $X = \mathcal{K}_\sigma\left(\Psi - \psi\right)$ as well as $Y = \mathcal{L}$, we have

$$\mathrm{Var}[\varepsilon_\sigma] = \frac{1}{M}\left\{\mathrm{E}[\mathcal{K}_\sigma]^2\mathrm{Var}[\mathcal{L}] + \mathrm{Var}[\mathcal{K}_\sigma]\left(\mathrm{E}[\mathcal{L}]^2 + \mathrm{Var}[\mathcal{L}]^2\right)\right\}, \tag{B.3}$$

where arguments for $\mathcal{K}_\sigma$ are omitted for brevity.

B.2 Expressing Expected Estimation Error

We start with equation (3.15) and again treats Ψ_j and $\mathcal{L}_j$ as i.i.d. random variables, the expected error can be simplified as

$$\begin{aligned}
\mathrm{E}[\varepsilon_\sigma] &= \mathrm{E}\left[\frac{1}{M}\sum_{j=1}^{M}\mathcal{K}_\sigma\left(\Psi - \psi\right)\mathcal{L} - L_o\left(\psi\right)\right] \\
&= \mathrm{E}\left[\frac{1}{M}\sum_{j=1}^{M}\mathcal{K}_\sigma\left(\Psi - \psi\right)\mathcal{L}\right] - L_o\left(\psi\right) \\
&= \frac{1}{M}\mathrm{E}\left[\sum_{j=1}^{M}\mathcal{K}_\sigma\left(\Psi - \psi\right)\mathcal{L}\right] - L_o\left(\psi\right) \\
&= \frac{1}{M}\sum_{j=1}^{M}\mathrm{E}[\mathcal{K}_\sigma\left(\Psi - \psi\right)\mathcal{L}] - L_o\left(\psi\right) \\
&= \mathrm{E}[\mathcal{K}_\sigma\left(\Psi - \psi\right)\mathcal{L}] - L_o\left(\psi\right) \\
&= \mathrm{E}[\mathcal{K}_\sigma\left(\Psi - \psi\right)]\mathrm{E}[\mathcal{L}] - L_o\left(\psi\right)
\end{aligned} \tag{B.4}$$

Note that the derived results (B.3) and (B.4) are of the same form to the ones made by Knaus and Zwicker [24]. But in our derivation, we are using a higher-dimensional (joint position directional space) kernel with outgoing radiance samples which can be seen as a generalization to the original formulation.

[1] True for independent X and Y.

B.3 Expected Value of the Kernel

We will begin with the Hessian matrix $H = \nabla\nabla^T$ for spherical coordinates[2]. Recall that gradient in spherical coordinates is

$$\nabla p_d = \frac{\partial p_d}{\partial r}\hat{r} + \frac{1}{r}\frac{\partial p_d}{\partial \theta}\hat{\theta} + \frac{1}{r\sin(\theta)}\frac{\partial p_d}{\partial \phi}\hat{\phi},$$

from which H_d can be derived (note that in our case r component can be eliminated as $r = 1$ and p_d has a zero derivative with respect to r):

$$H_d = \begin{bmatrix} \dfrac{\partial^2 p_d}{\partial \theta^2} & \dfrac{1}{\sin(\theta)}\dfrac{\partial^2 p_d}{\partial \theta \partial \phi} \\ \dfrac{1}{\sin(\theta)}\dfrac{\partial^2 p_d}{\partial \phi \partial \theta} & \dfrac{1}{\sin^2(\theta)}\dfrac{\partial^2 p_d}{\partial \phi^2} \end{bmatrix}. \tag{B.5}$$

In the analysis of $\mathrm{E}[\mathcal{K}_\sigma(\Psi - \psi)]$, we limit ourselves to the terms up to the second-order ones since they form the first non-vanishing moments in integral (3.22) (in addition to the zeroth moment). It is unfortunate that if we expand the density function at the origin of the kernel, there is a singularity directly lies on that location. To elaborate, ∇p_d has a singularity on the direction ϕ if being evaluated with $\theta = 0$, making the denominator zero. We can approach this problem in two directions. The first possibility is we can simply assume that the density variations (with respect to kernel bandwidth σ) involving the ϕ component is at most equal to the θ component; or being an isotropic distribution, i.e., independent of ϕ. They both allow ignoring ϕ, at a cost that the behavior of highly anisotropic distribution is not captured. The second possibility is a more general one that employs *small-angle approximation* on $\mathcal{K}_\sigma$ as well as on $\mathcal{S}$. Since we are concerned about the expected value after a large number of iterations which corresponds to small bandwidths (section 3.4) and consequently small angles. We will explore both of them shortly.

Substituting equation (3.23) into (3.22), and note that a hemisphere can be symmetrically integrated under the domain $D_s = \{-\pi/2 \leq \theta \leq \pi/2, -\pi/2 \leq \phi \leq \pi/2\}$. Eliminating all odd-moment integrals with the first proposed method in mind to tackle the

[2] We omitted the trivial case of Cartesian Hessian.

singularity, we arrive at the form

$$\begin{aligned}\mathrm{E}[\mathcal{K}_\sigma(\Psi-\psi)] = &\int_{\mathcal{S}} \mathcal{K}_\sigma(\psi')\, p_m(0)\, p_d(0)\, d\psi' \\ &+ \int_{\mathcal{S}} \mathcal{K}_\sigma(\psi')\, p_m(0)\, \frac{1}{2!}(\omega')^T H_d(0)(\omega')\, d\psi'. \\ &+ \int_{\mathcal{S}} \mathcal{K}_\sigma(\psi')\, p_d(0)\, \frac{1}{2!}(x')^T H_m(0)(x')\, d\psi'.\end{aligned} \tag{B.6}$$

For the first part in the RHS, the integrand is a constant and will integrate to $p_m(0)\, p_d(0)$ (recall equation 3.5). For the second and the third part, they are equivalent to the following integrals after expanding the products:

$$\frac{p_m(0)}{2! \cdot C_\sigma} \cdot \frac{\partial^2 p_d(0)}{\partial \theta^2} \int_{\mathcal{S}} \theta'^2 d\psi', \tag{B.7}$$

and

$$\frac{p_d(0)}{2! \cdot C_\sigma} \cdot \left[\frac{\partial^2 p_m(0)}{\partial u^2} \int_{\mathcal{S}} u'^2 d\psi' + \frac{\partial^2 p_m(0)}{\partial v^2} \int_{\mathcal{S}} v'^2 d\psi'\right], \tag{B.8}$$

with $x = (u, v)$. Integrating (B.7) over $\mathcal{S}$, and expanding C_σ with $t(\sigma)$ written as t_σ for clarity, we have

$$\begin{aligned}&\frac{p_m(0)}{4\pi^2\sigma^2[1-\cos(t_\sigma)]} \cdot \frac{\partial^2 p_d(0)}{\partial \theta^2} \int_{\mathcal{M}} \int_{\Omega} \theta'^2 dA' d\omega' \\ =& \frac{p_m(0)}{4\pi[1-\cos(t_\sigma)]} \cdot \frac{\partial^2 p_d(0)}{\partial \theta^2} \int_{\Omega} \theta'^2 d\omega' \\ =& \frac{p_m(0)}{4\pi[1-\cos(t_\sigma)]} \cdot \frac{\partial^2 p_d(0)}{\partial \theta^2} \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} \int_{-t_\sigma}^{t_\sigma} \theta'^2 |\sin(\theta')| d\theta' d\phi' \\ =& \frac{p_m(0)}{4[1-\cos(t_\sigma)]} \cdot \frac{\partial^2 p_d(0)}{\partial \theta^2} \left[\int_0^{t_\sigma} \theta'^2 \sin(\theta')\, d\theta' - \int_{-t_\sigma}^{0} \theta'^2 \sin(\theta')\, d\theta'\right] \\ =& \frac{p_m(0)}{4[1-\cos(t_\sigma)]} \cdot \frac{\partial^2 p_d(0)}{\partial \theta^2} \left[-2\left(t_\sigma^2-2\right)\cos(t_\sigma) + 4t_\sigma \sin(t_\sigma) - 4\right].\end{aligned} \tag{B.9}$$

Performing a series expansion at $t_\sigma = 0$, the result can be expressed as

$$\frac{p_m(0)}{4} \cdot \frac{\partial^2 p_d(0)}{\partial \theta^2} \left[t_\sigma^2 + O\left(t_\sigma^4\right)\right] \approx C_d \cdot t_\sigma^2, \tag{B.10}$$

where C_d is a constant. For equation (B.8), the term that is related to u can be directly computed as

$$
\begin{aligned}
&\frac{p_d\left(0\right)}{4\pi^2\sigma^2\left[1-\cos\left(t_\sigma\right)\right]}\cdot\frac{\partial^2 p_m\left(0\right)}{\partial u^2}\int_{\mathcal{M}}\int_{\Omega}u'^2dA'd\omega'\\
&=\frac{p_d\left(0\right)}{2\pi\sigma^2}\cdot\frac{\partial^2 p_m\left(0\right)}{\partial u^2}\int_{\mathcal{M}}u'^2dA'\\
&=\frac{p_d\left(0\right)}{2\pi\sigma^2}\cdot\frac{\partial^2 p_m\left(0\right)}{\partial u^2}\int_{-\sigma}^{\sigma}\int_{-\sqrt{\sigma^2-v'^2}}^{\sqrt{\sigma^2-v'^2}}u'^2du'dv'\\
&=\frac{p_d\left(0\right)}{2\pi\sigma^2}\cdot\frac{\partial^2 p_m\left(0\right)}{\partial u^2}\cdot\frac{\pi\sigma^4}{4}\\
&=C_u\cdot\sigma^2,
\end{aligned}
\tag{B.11}
$$

and the v-related term can be derived similarly as $C_v\cdot\sigma^2$, where C_u and C_v are both constants. We can now write the expected value as

$$
\begin{aligned}
\mathrm{E}[\mathcal{K}_\sigma\left(\Psi-\psi\right)]&=\int_{\mathcal{S}}\mathcal{K}_\sigma\left(\psi'-\psi\right)p\left(\psi'\right)d\psi'\\
&\approx p_m\left(0\right)p_d\left(0\right)+C_d\cdot t_\sigma^2+\left(C_u+C_v\right)\cdot\sigma^2\\
&=p_m\left(0\right)p_d\left(0\right)+C_k\cdot\sigma^2,
\end{aligned}
\tag{B.12}
$$

where C_k is a collective constant. The last step is valid under the implicit constraint $t\left(\sigma\right)=O\left(\sigma\right)$ which should be obeyed in addition to (3.7).

As mentioned earlier, another possible derivation is to employ small-angle approximation on the directional component in $\mathcal{S}$. Our 4-D kernel and $p\left(\psi\right)$ can thus be trivially integrated over $\mathbb{R}^4$ without singularities in gradient and unfactorable jacobians with density transformations. Furthermore, the kernel can now be approximated as

$$
\frac{1}{C_\sigma}\approx\frac{1}{\pi^2\sigma^2t_\sigma^2}
\tag{B.13}
$$

with decent precision in the region around $\omega=\left(0,0\right)$. It is also possible to let $t\left(\sigma\right)$ as a polynomial with arbitrary degrees as long as it satisfies all aforementioned constraints. The analysis of $\mathrm{E}[\mathcal{K}_\sigma\left(\Psi-\psi\right)]$ then follows the same procedure in this section and will arrive at the same result as (B.12). Namely, $\mathrm{E}[\mathcal{K}_\sigma\left(\Psi-\psi\right)]\approx p_m\left(0\right)p_d\left(0\right)+C_k'\cdot\sigma^2$ with a different constant C_k'.

doi:10.6342/NTU201901709

Bibliography

[1] B. Bitterli and W. Jarosz. Beyond points and beams: Higher-dimensional photon samples for volumetric light transport. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 36(4):1–12, July 2017. 19

[2] G. Bouchard, J.-C. Iehl, V. Ostromoukhov, and P. Poulin. Improving robustness of monte-carlo global illumination with directional regularization. In *SIGGRAPH Asia 2013 Technical Briefs*, SA '13, pages 22:1–22:4. ACM, 2013. 4, 66

[3] T.-C. Chang. Photon renderer, 2016–2019. https://github.com/TzuChieh/Photon-v2. 42

[4] X. Deng, S. Jiao, B. Bitterli, and W. Jarosz. Photon surfaces for robust, unbiased volumetric density estimation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 38(4), jul 2019. 19

[5] L. Fascione, J. Hanika, M. Leone, M. Droske, J. Schwarzhaupt, T. Davidovič, A. Weidlich, and J. Meng. Manuka: A batch-shading architecture for spectral path tracing in movie production. *ACM Transactions on Graphics*, 37(3):31:1–31:18, Aug. 2018. 1, 18

[6] K. X. Fu-Jun Luan, Li-Fan Wu. Anisotropic density estimation for photon mapping. *Computational Visual Media*, 1(3):221–228, Sep 2015. 14

[7] I. Georgiev, T. Ize, M. Farnsworth, R. Montoya-Vozmediano, A. King, B. V. Lommel, A. Jimenez, O. Anson, S. Ogaki, E. Johnston, A. Herubel, D. Russell, F. Ser-

doi:10.6342/NTU201901709

vant, and M. Fajardo. Arnold: A brute-force production path tracer. *ACM Transactions on Graphics*, 37(3):32:1–32:12, Aug. 2018. 1

[8] I. Georgiev, J. Křivánek, T. Davidovič, and P. Slusallek. Light transport simulation with vertex connection and merging. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 31(6):192:1–192:10, Nov. 2012. 4, 18

[9] T. Hachisuka. *Robust Light Transport Simulation Using Progressive Density Estimation*. PhD thesis, University of California at San Diego, 2011. AAI3473619. 7

[10] T. Hachisuka and H. W. Jensen. Stochastic progressive photon mapping. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 28(5):130:1—130:8, Dec. 2009. 3, 17

[11] T. Hachisuka, S. Ogaki, and H. W. Jensen. Progressive photon mapping. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 27(5):130:1—130:8, Dec. 2008. 3, 17, 29

[12] T. Hachisuka, J. Pantaleoni, and H. W. Jensen. A path space extension for robust light transport simulation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 31(6):191:1–191:10, Jan. 2012. 4, 18

[13] V. Havran, J. Bittner, R. Herzog, and H.-P. Seidel. Ray maps for global illumination. In *RT_EGSR03*, pages 43–54, June 2005. 15

[14] V. Havran, R. Herzog, and H.-P. Seidel. Fast final gathering via reverse photon mapping. *Computer Graphics Forum*, 24(3):323–332, Sept. 2005. 17

[15] R. Herzog. *Advanced Density Estimation Techniques for Global Illumination*. PhD thesis, Saarland University (MPI Informatik), 2005. 22

[16] R. Herzog, V. Havran, S. Kinuwaki, K. Myszkowski, and H.-P. Seidel. Global illumination using photon ray splatting. *Computer Graphics Forum (Proceedings of Eurographics)*, 26(3):503–513, 2007. 15

doi:10.6342/NTU201901709

[17] H. Hey and W. Purgathofer. Importance sampling with hemispherical particle footprints. In *Proceedings of the 18th Spring Conference on Computer Graphics*, SCCG '02, pages 107–114. ACM, 2002. 45

[18] H. W. Jensen. Global illumination using photon maps. In W. Hansmann, W. T. Hewitt, W. Purgathofer, X. Pueyo, and P. Schröder, editors, *Rendering Techniques (Proceedings of the Eurographics Workshop on Rendering)*, pages 21–30, Vienna, June 1996. Springer-Verlag. 7, 8, 19

[19] H. W. Jensen and N. J. Christensen. Photon maps in bidirectional Monte Carlo ray tracing of complex objects. *Computers & Graphics*, 19(2):215–224, Mar. 1995. 3

[20] H. W. Jensen, F. Suykens, and P. H. Christensen. A practical guide to global illumination using photon mapping. In *ACM SIGGRAPH Course Notes*. ACM Press, Aug. 2001. 8

[21] J. T. Kajiya. The rendering equation. *Computer Graphics (Proceedings of SIGGRAPH)*, 20(4):143–150, Aug. 1986. 7, 8, 67

[22] A. S. Kaplanyan and C. Dachsbacher. Adaptive progressive photon mapping. *ACM Transactions on Graphics*, 32(2):16:1–16:13, Apr. 2013. 4, 18, 66

[23] A. S. Kaplanyan and C. Dachsbacher. Path space regularization for holistic and robust light transport. *Computer Graphics Forum (Proceedings of Eurographics)*, 32(2):63–72, 2013. 4, 19, 53

[24] C. Knaus and M. Zwicker. Progressive photon mapping: A probabilistic approach. *ACM Transactions on Graphics*, 30(3):25:1–25:13, May 2011. 3, 17, 25, 30, 31, 33, 34, 36, 72

[25] S. Kniep, S. Häring, and M. Magnor. Efficient and accurate rendering of complex light sources. In *Proceedings of the Twentieth Eurographics Conference on Rendering*, EGSR'09, pages 1073–1081. Eurographics Association, 2009. 4, 53

[26] M. Pharr, W. Jakob, and G. Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, San Francisco, CA, USA, 3 edition, 2016. 15

[27] L. Schjøth, J. R. Frisvad, K. Erleben, and J. Sporring. Photon differentials. In *Proceedings of GRAPHITE*, volume 1, page 179, Perth, Australia, 2007. ACM Press. 14

[28] P. Shirley, B. Wade, P. M. Hubbard, D. Zareski, B. Walter, and D. P. Greenberg. Global illumination via density-estimation. In P. M. Hanrahan and W. Purgathofer, editors, *Photorealistic Rendering Techniques (Proceedings of the Eurographics Workshop on Rendering)*, pages 219–230. Springer-Verlag, 1995. 8

[29] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*, volume 26 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, New York, NY, 1986. 15, 22

[30] B. Spencer and M. W. Jones. Progressive photon relaxation. *ACM Trans. Graph.*, 32(1):7:1–7:11, Feb. 2013. 19

[31] E. Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. Ph.D. Thesis, Stanford University, United States – California, Dec. 1997. 3, 10, 15, 25, 46, 52, 68, 69

[32] E. Veach and L. J. Guibas. Optimally combining sampling techniques for Monte Carlo rendering. In *Annual Conference Series (Proceedings of SIGGRAPH)*, volume 29, pages 419–428. ACM Press, Aug. 1995. 18

[33] E. Velázquez-Armendáriz, Z. Dong, B. Walter, and D. P. Greenberg. Complex luminaires: Illumination and appearance rendering. *ACM Trans. Graph.*, 34(3):26:1–26:15, May 2015. 4, 53

[34] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance. Microfacet models for refraction through rough surfaces. In *Rendering Techniques (Proceedings of the Eu-*

doi:10.6342/NTU201901709

rographics Symposium on Rendering), pages 195–206. Eurographics Association, 2007. 22