Guided Lens Sampling for Efficient Monte Carlo Circle-of-Confusion Rendering

JIAWEI HUANG, International Digital Economy Academy, China
SHAOKUN ZHENG, Tsinghua University, China
KUN XU, Tsinghua University, China
YOSHIFUMI KITAMURA, Tohoku University, Japan
JIAPING WANG*, International Digital Economy Academy / DAIR, China

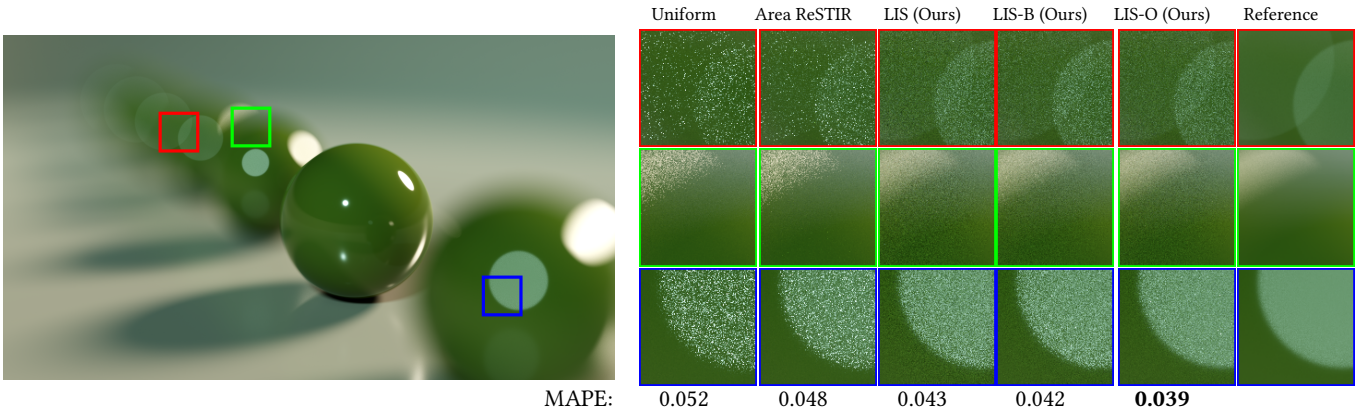


Fig. 1. A defocused view rendered using a thin lens model with 256 samples per pixel (SPP). Uniform lens sampling produces significant noise in complex regions with defocused highlight spots. Area ReSTIR [Zhang et al. 2024], a path resampling approach, reduces variance to some extent. In contrast, our lens importance sampling (LIS) method excels at sampling high-contribution regions, achieving a notably more accurate result.

We introduce a guided lens sampling method for efficient rendering of circles of confusion (CoCs). While traditional Monte Carlo techniques simulate depth-of-field (DoF) effects by perturbing camera rays at the lens, uniform lens sampling often results in significant noise by failing to prioritize rays toward highlight regions in the scene. Although path guiding has proven effective for global illumination by learning importance distributions for incoming radiance, no comparable guiding technique for CoCs exists, primarily due to the strong parallax between adjacent pixels. We model highlight spots in world space using a globally shared radiance field, which is then transformed into lens space through a bipolar-cone projection to guide camera ray generation. We implement this theory using 3D Gaussians, achieving fast, robust guiding with minimal computational and storage overhead, making it suitable for production rendering. We also propose two extensions to further enhance local adaptation. Our experiments show that this approach significantly improves the sampling efficiency for CoC rendering.

ACM Reference Format:
Jiawei Huang, Shaokun Zheng, Kun Xu, Yoshifumi Kitamura, and Jiaping Wang. 2025. Guided Lens Sampling for Efficient Monte Carlo Circle-of-Confusion Rendering. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3721238.3730608

*Jiaping Wang (jiapw.cg@gmail.com) is the corresponding author.

Jiawei Huang, Hengqin-Macau Digital and Artificial Intelligence Research, International Digital Economy Academy, Shenzhen, China, huangjiawei@idea.edu.cn. Shaokun Zheng, BNRist, Department of CS&T, Tsinghua University, Beijing, China, zsk20@mails.tsinghua.edu.cn. Kun Xu, BNRist, Department of CS&T, Tsinghua University, Beijing, China, xukun@tsinghua.edu.cn. Yoshifumi Kitamura, Tohoku University, Sendai, Japan, kitamura@riec.tohoku.ac.jp. Jiaping Wang, Hengqin-Macau Digital and Artificial Intelligence Research, International Digital Economy Academy, Shenzhen, China, jiapw.cg@gmail.com.

SIGGRAPH Conference Papers '25, Vancouver, BC, Canada

ACM ISBN 979-8-4007-1540-2/25/08
https://doi.org/10.1145/3721238.3730608

1 Introduction

Circles of confusion (CoCs) are a captivating visual effect in photography, occurring when out-of-focus highlight points become large blurred patterns in the image due to shallow *depth of field* (DoF). The Japanese term *bokeh*, meaning "confusion" or "dullness," is often used by photographers to describe the aesthetic quality of CoCs.

In rendering, DoF effects have been simulated since the introduction of *distributed ray tracing* [Cook et al. 1984], which perturbs the origins of camera rays at the lens. However, naïve sampling with a uniform disk distribution can result in significant variance, as only a small fraction of camera rays are likely to intersect the tiny highlight spots that contribute most to the CoC effect. An alternative approach, commonly used in video games and many production renderings, is to generate an all-in-focus image using a pinhole camera model and then synthesize DoF in post-processing, often with a

depth-conditioned filter [Potmesil and Chakravarty 1981]. While this method is efficient and noise-free, it can introduce noticeable artifacts, such as halos around thin objects like hair.

While importance sampling of lenses remains underexplored, some research efforts can be seen as attempts to alleviate the low convergence efficiency of CoC-involved Monte Carlo rendering. For instance, several Monte Carlo denoisers explicitly address the handling of distributed effects [Zwicker et al. 2015]. Recently, path resampling has also been extended to CoC rendering [Zhang et al. 2024]. However, it still relies on uniform sampling, and the low quality of the base distribution limits the improvements introduced by resampling.

In this paper, we derive a generalized lens-space importance sampling formulation from the *measurement equation* [Veach 1998] and present an unbiased guiding technique that *explicitly* importance samples the lens. This approach more effectively distributes camera rays toward highlight spots in the scene, enabling efficient CoC-involved Monte Carlo rendering. Inspired by *path guiding* techniques in enhancing the rendering efficiency of global illumination, our method adopts a similar pipeline that iteratively refines the importance sampling distribution using previously rendered samples.

However, the straightforward idea of directly maintaining the sampling distribution in the lens space might not be viable. Even neighboring pixels on the film can exhibit drastically varying distributions of incoming radiance on the lens. Such strong parallax presents significant challenges for fitting and sampling within reasonable computational and storage overhead.

We address this problem by maintaining a shared world-space radiance field, represented by 3D Gaussians that approximate highlight spots. A novel bipolar-cone projection dynamically culls and transforms this field into lens-space distributions during sampling, and two refinement techniques further mitigate parallax and occlusion for improved local adaptation. Integrated into a production GPU path tracer, our method delivers robust and efficient CoC rendering with minimal overhead, outperforming uniform lens sampling and path resampling across diverse scenes.

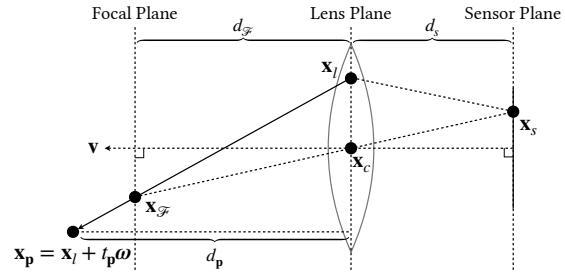
To summarize, our main contributions include:

- A generalized lens-space importance sampling formulation derived from the measurement equation to guide camera ray generation for efficient circle-of-confusion rendering;
- A bipolar-cone projection scheme to connect spatial highlight representations to local lens-space distributions; and
- A robust, practical pipeline that integrates guided lens sampling into a production GPU renderer, with refinement extensions to handle parallax and occlusion.

2 Related Work and Background

2.1 Depth-of-Field Rendering and Circles of Confusion

Lens simulation plays an important role in Monte Carlo rendering. Generally, two approaches are used: the thin-lens model [Cook et al. 1984], and the simulation of a complex lens system [Hullin et al. 2012; Kolb et al. 1995]. While the latter provides extreme realism, most movie productions opt for the thin-lens model because of its efficiency and ease of use. As shown in Fig. 2, in the thin-lens model, all camera rays originating from the lens disk pass through the focal point $\mathbf{x}_{\mathcal{F}}$ of a sensor point $\mathbf{x}_s$ and contribute to its measurement. Naively distributing rays within the lens is inefficient for rendering pronounced CoC effects, as there is a low probability of finding rays that contribute significantly to the effect. However, to the best of our knowledge, little work has been done to improve sampling efficiency in this context.



(a)

Symbol	Description
$\mathbf{x}_c$	Center of the lens
$\mathbf{x}_s$	A point on the sensor
$\mathbf{x}_{\mathcal{F}}$	Focused point of $\mathbf{x}_s$ on the focal plane
$\mathbf{x}_l$	A point on the lens
$\mathbf{x}_\mathbf{p}$	A 3D point in the world space
$\mathbf{v}$	Optical axis
$\boldsymbol{\omega}$	Direction of camera ray
$t_\mathbf{p}$	Ray distance (depth)
$d_{\mathcal{F}}$	Focus distance
d_s	Distance between sensor plane and the lens
$d_\mathbf{p}$	Distance between point p and the lens plane
r	Radius of the lens
A	Area of the lens

(b)

Fig. 2. Setup of the thin-lens model for a single sensor point $\mathbf{x}_s$. Each camera ray originates on the lens disk, converges at $\mathbf{x}_{\mathcal{F}}$ on the focal plane, and then diverges. Here, $\mathbf{x}_{\mathcal{F}}$ is the intersection of the line passing through $\mathbf{x}_s$ and the lens center $\mathbf{x}_c$ with the focal plane. Denotations involved in our research are described in (b).

Due to the inefficiency of synthesizing DoF effects via Monte Carlo simulation, post-processing methods are favored in video games and much of production rendering. Potmesil and Chakravarty [1981] demonstrated that DoF effects can be synthesized by distributing energy in the image plane, which can be interpreted as filtering. Other post-processing approaches [Leimkühler et al. 2018; Mitchell and Netravali 1988] also apply filtering to reduce noise and enhance image quality, but they often fail to capture the visually appealing characteristics of CoCs. With the advancement of deep learning, data-driven approaches have been explored [Ignatov et al. 2020; Luo et al. 2023]. However, their heavy reliance on datasets may conflict with the demands of production rendering for novel visual designs. Worse still, post-processing approaches are known to have accuracy issues, as they cannot correctly capture view-dependent effects, such as partially occluded spots and specular reflections. Therefore, for high-quality production rendering, distributed ray tracing remains the typical choice despite its low efficiency.

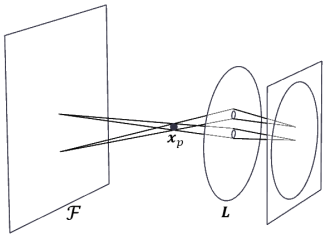


Fig. 3. For the same highlight point that creates a CoC, the optimal distribution on lens for different locations on sensor varies significantly, posing challenges for modeling with existing path guiding techniques.

2.2 Importance Sampling

Importance sampling is a key technique for variance reduction in Monte Carlo integration. In rendering, it has been extensively applied to BRDF sampling [Lawrence et al. 2004], light sampling [Veach and Guibas 1995b], volume rendering [Kulla and Fajardo 2012], and even pixel filtering [Ernst et al. 2006]. However, applying importance sampling directly to lens systems for DoF rendering remains largely unexplored, due to the difficulty of modeling the spatially varying distribution.

ReSTIR methods [Bitterli et al. 2020; Lin et al. 2022; Talbot et al. 2005] introduce *resampled importance sampling* [Talbot et al. 2005] into real-time path tracing. *Area ReSTIR* [Zhang et al. 2024] extends this concept to DoF and anti-aliasing by reusing samples from neighboring pixels. However, because it relies on uniform lens sampling, the benefits of resampling can be limited by the quality of the initial samples. In contrast, we directly importance sample the lens. Furthermore, our technique is orthogonal to Area ReSTIR and can be integrated to provide higher-quality candidates for resampling.

2.3 Path Guiding

To address the challenge of determining appropriate distributions a priori, adaptive importance sampling, often referred to as *path guiding* in the rendering context, has been introduced to iteratively fit and refine the importance sampling distribution. Our method can also be classified as a guiding technique that progressively learns the distribution for importance sampling a lens.

The choice of distribution representation is a key factor influencing both the learning efficiency and sampling quality of a path guiding method. Some approaches employ explicit data structures for partitioning spatial and directional distributions. For instance, Müller et al. [2017] introduced the spatio-directional tree (SD-Tree) structure in *Practical Path Guiding* (PPG), and Lu et al. [2024] proposed fitting the global radiance distribution as voxels. Many methods also leverage parameterized distribution models, with the exponential family playing a central role [Dodik et al. 2022; Huang et al. 2024a; Ruppert et al. 2020; Vorba et al. 2014].

As shown in Fig. 3, the target sampling distribution in lens space varies significantly across the sensor plane, making direct fitting challenging. Therefore, we chose to learn a global distribution in world space and developed a technique to transform it into the local lens space when sampling camera rays. Inspired by Huang et al. [2024b], we found 3D Gaussians well-suited for this purpose.

3 Method Formulation

3.1 Generalized Lens-Space Importance Sampling

As discussed in § 2.1, DoF effects are typically simulated using a thin-lens camera model, where camera ray origins are uniformly sampled on the lens disk. In the following, we demonstrate that by deriving from the *measurement equation* [Veach 1998], lens sampling can be generalized to accommodate arbitrary sampling distributions in the lens space. This formulation reveals that uniform sampling is a sub-optimal sampling technique for a specific sample weighting function, with its inefficiency stemming from neglecting the distribution of incoming radiance at the lens.

3.1.1 The measurement equation for thin-lens cameras. The sensor measurement $I(\mathbf{x}_s)$ at a point $\mathbf{x}_s$ is the result of an integral over the weighted incoming radiance carried by rays passing through the lens at $\mathbf{x}_l$. In the thin-lens model, all rays that reach the sensor point $\mathbf{x}_s$ must converge at the same point $\mathbf{x}_\mathcal{F}$ on the focal plane. As shown in Fig. 2 (a), for a thin lens centered at the $\mathbf{x}_c$ and a sensor point $\mathbf{x}_s$ on the sensor, the corresponding focused point $\mathbf{x}_\mathcal{F}$ on the focal plane is a function of $\mathbf{x}_s$:

$$\mathbf{x}_\mathcal{F} = \mathcal{F}(\mathbf{x}_s) = \mathbf{x}_c + \frac{d_\mathcal{F}}{\cos\theta_l} \cdot \frac{\mathbf{x}_c - \mathbf{x}_s}{\|\mathbf{x}_c - \mathbf{x}_s\|^2}, \tag{1}$$

where $d_\mathcal{F}$ denotes the distance from the focal plane to the lens; and θ_l the angle between $\mathbf{x}_c - \mathbf{x}_s$ and the optical axis. For any point $\mathbf{x}_l$, the ray should converge at $\mathbf{x}_\mathcal{F}$, hence

$$\boldsymbol{\omega} = \frac{\mathbf{x}_\mathcal{F} - \mathbf{x}_l}{\|\mathbf{x}_\mathcal{F} - \mathbf{x}_l\|^2}. \tag{2}$$

With $L(\mathbf{x}_l)$ being the radiance reaching $\mathbf{x}_l$ from direction $\boldsymbol{\omega}$, we formally define the measurement equation for sensor point $\mathbf{x}_s$ as

$$I(\mathbf{x}_s) = \int_\Omega W_e(\mathbf{x}_s, \boldsymbol{\omega}_s)\, L(\mathbf{x}_l)\, |\cos\theta_s|\, \mathrm{d}\boldsymbol{\omega}_s, \tag{3}$$

where Ω denotes the integral domain of all possible ray directions from the lens disk to $\mathbf{x}_\mathcal{F}$; $\boldsymbol{\omega}_s$ the corresponding ray direction on the sensor side; $W_e(\mathbf{x}_s, \boldsymbol{\omega}_s)$ is the sensor response function; and $|\cos\theta_s|$ accounts for the angle between $\boldsymbol{\omega}_s$ and the optical axis.

To facilitate the derivation that follows, we now fix a sensor point $\mathbf{x}_s$ and suppress its explicit appearance in the notation for clarity. All subsequent expressions in this section are implicitly conditioned on this fixed $\mathbf{x}_s$, and the formulation remains valid for arbitrary positions on the sensor plane.

3.1.2 Lens-space importance sampling. To calculate Eq. (3) with Monte Carlo integration, we may use some distribution $p_A(\mathbf{x})$, which is defined on the lens area A, to sample points on the lens and obtain an estimate:

$$\langle I(\mathbf{x}_s)\rangle = \frac{1}{N}\sum_{i=1}^{N} \frac{W_e(\boldsymbol{\omega}_s^i)\,|\cos\theta_s^i|}{T(\mathbf{x}_l^i)} \cdot \frac{L(\mathbf{x}_l^i)}{p_A(\mathbf{x}_l^i)}, \tag{4}$$

where

$$T(\mathbf{x}_l^i) = \frac{\|\mathbf{x}_\mathcal{F} - \mathbf{x}_l^i\|^2}{|\cos\theta^i|} \tag{5}$$

is the inverse of Jacobian of the area-to-solid-angle transformation.

The first term of the summand in Eq. (4) can be interpreted as a sample weighting function for the lens samples (note that for any point $\mathbf{x}_l$ there is always a corresponding $W(\boldsymbol{\omega})$):

$$W(\mathbf{x}_l) = \frac{W_e(\boldsymbol{\omega}_s)\,|\cos\theta_s|}{T(\mathbf{x}_l)}. \tag{6}$$

Therefore, given a weighting function, we can use arbitrary lens-space distributions to sample a lens point $\mathbf{x}_l$, construct the camera ray $(\mathbf{x}_l + t\boldsymbol{\omega})$, and evaluate its contribution with path tracing.

This perspective naturally leads to the objective of our lens-space importance sampling technique: by learning a lens-space sampling distribution $\hat{p}$ that (approximately) satisfies

$$\hat{p}(\mathbf{x}_l) \propto W(\mathbf{x}_l) \cdot L(\mathbf{x}_l), \tag{7}$$

there is a high likelihood that we can reduce the estimation variance in CoC-involved Monte Carlo rendering.

3.1.3 Revisiting uniform lens sampling. As readers may notice, conventional renderers [Pharr et al. 2023] that use uniform lens sampling, $p_U(\mathbf{x}_l) = \frac{1}{A}$, typically average radiance samples as follows:

$$\langle I(\mathbf{x}_s)\rangle_U = \frac{1}{N}\sum_{i=1}^{N} L(\mathbf{x}_l^i) = \frac{1}{N}\sum_{i=1}^{N} W_U(\mathbf{x}_l^i)\cdot\frac{L(\mathbf{x}_l^i)}{p_U(\mathbf{x}_l^i)}. \tag{8}$$

This implies a specific choice for the sample weighting function:

$$W_U(\mathbf{x}_l) = p_U(\mathbf{x}_l) = \frac{1}{A}. \tag{9}$$

From the perspective of lens-space importance sampling, as defined in Eq. (7), using the uniform disk distribution fits perfectly with this specific W_U. However, it completely ignores the incoming radiance at the lens, leading to inefficiency in the presence of CoCs.

Notably, although our lens-space importance sampling formulation can adapt to other sample weighting functions, to align with the rendering results of the de facto uniform lens sampling scheme, we use the same weighting function as in Eq. (9). In this case, the objective distribution previously defined in Eq. (7) becomes

$$\hat{p}'(\mathbf{x}_l) \propto W_U(\mathbf{x}_l)\cdot L(\mathbf{x}_l) = \frac{1}{A}\cdot L(\mathbf{x}_l) \propto L(\mathbf{x}_l) \tag{10}$$

and we may estimate the measurement integral as

$$\langle I(\mathbf{x}_s)\rangle' = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{A}\cdot\frac{L(\mathbf{x}_l^i)}{\hat{p}'(\mathbf{x}_l^i)}. \tag{11}$$

3.2 Bipolar-Cone Distribution Projection

In practice, directly fitting the lens-space distribution $\hat{p}'(\mathbf{x}_l)$, which corresponds to a given sensor point $\mathbf{x}_s$, is inherently challenging. This is due to the significant variability of $\hat{p}'(\mathbf{x}_l)$ across different sensor points $\mathbf{x}_s$, making it difficult to model and fit accurate distributions across the entire sensor plane.

To avoid these issues, we propose a bipolar-cone projection technique that transforms world-space spatial distributions into the local lens space. Our key insight is that, by leveraging the focusing characteristics of the thin-lens model, the $\mathbf{x}_s$-dependent distribution $\hat{p}'(\mathbf{x}_l)$ can be decomposed into an $\mathbf{x}_s$-conditioned projection of an $\mathbf{x}_s$-*independent* spatial distribution.

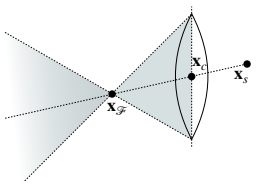


Fig. 4. The visible volume for point $\mathbf{x}_s$ on the sensor can be described as a bipolar cone (masked in shadow).

3.2.1 The bipolar-cone visibility volume. Since all rays reaching the sensor at $\mathbf{x}_s$ must converge at $\mathbf{x}_{\mathcal{F}}$, as shown in Fig. 4, only spatial points within the bipolar-cone volume $V(\mathbf{x}_{\mathcal{F}})$, formed by connecting the circumference of the lens to $\mathbf{x}_{\mathcal{F}}$ and extending the cone, can contribute to the rendering. Based on this model, we can define a culling function $C(\mathbf{x})$, conditioned on $\mathbf{x}_{\mathcal{F}}$, to decide if a world-space point $\mathbf{x}$ lies within $V(\mathbf{x}_{\mathcal{F}})$. C can be practically evaluated by checking whether the line through $\mathbf{x}$ and $\mathbf{x}_{\mathcal{F}}$ intersects the lens plane within radius r, therefore it also has a directional form $C(\boldsymbol{\omega})$.

3.2.2 Decomposition of the lens-space sampling distribution. We define a world-space continuous radiance field $R_o : \mathbb{R}^3\times\mathbb{S}^2 \to \mathbb{R}_{\geq 0}$, where $R_o(\mathbf{x},\boldsymbol{\omega})$ denotes the outgoing radiance at spatial position $\mathbf{x}$ in direction $\boldsymbol{\omega}$. R_o is treated as a volumetric field, continuous and integrable over $\mathbb{R}^3$. Since all radiance along paths converging at $\mathbf{x}_{\mathcal{F}}$ contributes to the measurement at $\mathbf{x}_s$, with proper culling, it can be expressed as

$$I(\mathbf{x}_s) = \int_{\mathbb{R}^3} W(\mathbf{x})\,C(\mathbf{x})\,R_o(\mathbf{x},\omega_{\mathbf{x}_{\mathcal{F}}\to\mathbf{x}})\,d\mathbf{x}. \tag{12}$$

Alternatively, from the perspective of rays originating at $\mathbf{x}_{\mathcal{F}}$, Eq. (12) can be reformulated as a spherical integral:

$$I(\mathbf{x}_s) = \int_{\Omega}\int_0^{\infty} W(\boldsymbol{\omega})\,C(\boldsymbol{\omega})\,R_o(\mathbf{x}_{\mathcal{F}}+t\boldsymbol{\omega},\boldsymbol{\omega})\,t^2\,\mathrm{d}t\,\mathrm{d}\boldsymbol{\omega}. \tag{13}$$

This perspective leads to a spherical distribution:

$$p_I(\omega) = c\cdot W(\boldsymbol{\omega})\,C(\boldsymbol{\omega})\left(\int_0^{\infty} R_o(\mathbf{x}_{\mathcal{F}}+t\boldsymbol{\omega},\boldsymbol{\omega})\,t^2\mathrm{d}t\right), \tag{14}$$

where $c = \frac{1}{I(\mathbf{x}_s)}$ is the normalizing constant. On the other hand, we also have

$$I(\mathbf{x}_s) = \int_A W(\mathbf{x}_l)\cdot L(\mathbf{x}_l)\,\mathrm{d}\mathbf{x}_l. \tag{15}$$

Since Eq. (12) and (15) calculate the same integral with different measurements, we derive the relationship between distributions with accounting for the change in measurement:

$$p_I(\mathbf{x}_l) = c\cdot W(\mathbf{x}_l)\,L(\mathbf{x}_l) = [p_I(\boldsymbol{\omega}) + p_I(-\boldsymbol{\omega})]\;\frac{\cos\theta}{\|\mathbf{x}_{\mathcal{F}}-\mathbf{x}_l\|^2}, \tag{16}$$

where θ is the angle between $\mathbf{x}_{\mathcal{F}}-\mathbf{x}_l$ and the optical axis, and in the thin-lens model, both $\boldsymbol{\omega}$ and $-\boldsymbol{\omega}$ lead to the same $\mathbf{x}_l$. By incorporating R_o , we obtain a projection:

$$p_I(\mathbf{x}_l) = c\cdot W(\boldsymbol{\omega})\,C(\boldsymbol{\omega})\left(\int_{-\infty}^{\infty} R_o(\mathbf{x}_{\mathcal{F}}+t\boldsymbol{\omega},\boldsymbol{\omega})\,t^2\mathrm{d}t\right)\frac{\cos\theta}{\|\mathbf{x}_{\mathcal{F}}-\mathbf{x}_l\|^2}, \tag{17}$$

We refer to Eq. (17) as the *bipolar-cone projection* (BCP). The distribution $p_I(\mathbf{x}_l)$ on the left-hand side is exactly what we seek in Eq. (7). This projection shows that lens sampling can be guided using a known world-space radiance field, which effectively translates to importance sampling directions at the focused point $\mathbf{x}_{\mathcal{F}}$. Here, the radiance field R_o is an $\mathbf{x}_s$-*independent* spatial distribution, and with our $\mathbf{x}_s$-conditioned projection, we obtain the optimal distribution $\hat{p}$ for importance sampling the lens disk. With R_o being globally defined and consistent, it is feasible to obtain a close fit of it. In the following sections we detail the practical implementation of this approach in path tracing.

4 Practical Implementation with 3D Gaussians

In Eq. (17), three components require actual implementation: the weighting function W, the culling function C, and the radiance field R_o. While for conventional rendering, W is typically constant in conventional rendering and can be ignored in the distribution, the implementation of C and R_o needs special attention. In this section we detail the practical implementation of the radiance field and the corresponding culling function using ordinary pixel samples.

4.1 Choice of the Radiance Field Representation

We need to find a representation to model R_o. BCP requires that the chosen representation must (1) be a spatial representation; (2) be integrable over space in closed form; (3) support sampling and PDF evaluation with respect to direction at a given point.

We found that 3D Gaussians serve our purpose well. While 3D Gaussians have been widely used to model spatial radiance distributions in rendering, they alone do not fully satisfy all of the required properties. However, a recently discovered property of 3D Gaussians, as introduced by Huang et al. [2024b], effectively bridges this gap. Specifically, it enables the closed-form computation of the integral of *isotropic 3D Gaussians* along a ray and supports directional sampling, making it an ideal fit for our research. Below, we briefly review the definition and beneficial properties of 3D Gaussians.

4.1.1 Definition and spatial integral. An isotropic 3D Gaussian is

$$G(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}\right), \tag{18}$$

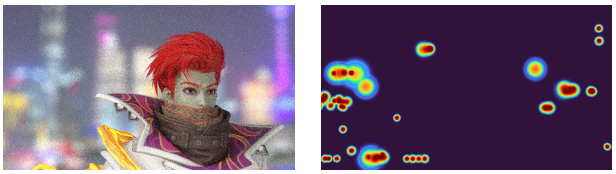
where $\boldsymbol{\mu}$ denotes the mean position and σ is the standard deviation. We thus express R_o as a weighted sum of isotropic 3D Gaussians:

$$R_o(\mathbf{x}, \boldsymbol{\omega}) = \sum_{i=1}^{N} w_i\, G_i(\mathbf{x}), \tag{19}$$

where $G_i(\mathbf{x})$ denotes the spatial profile of the i-th Gaussian. This representation simplifies R_o by assuming directional uniformity, which holds in practical applications.

An isotropic 3D Gaussian can be integrated over space:

$$\int_{R^3} G(\mathbf{x})\, \mathrm{d}\mathbf{x} = (2\pi\sigma^2)^{3/2}. \tag{20}$$



(a) (b)

Fig. 5. After each 1-SPP path tracing iteration (a), we refine a 2D record buffer and convert it to a spatial radiance field of the high-energy regions represented in isotropic 3D Gaussians (b).

4.1.2 Directional distribution. Given a fixed point $\mathbf{o}$ as the origin, the directional distribution of a 3D Gaussian is

$$\begin{aligned} F_o(\omega) = \frac{1}{c_G} \cdot \Big[& \sigma^2 e^{-\frac{d^2}{2\sigma^2}}\, d\cos\theta_o \\ & + \sqrt{\frac{\pi}{2}}\sigma e^{-\frac{d^2\sin^2\theta_o}{2\sigma^2}} \left(\sigma^2 + d^2\cos^2\theta_o\right) \\ & \times \left(1 + \operatorname{erf}\left(\frac{d\cos\theta_o}{\sqrt{2}\sigma}\right)\right) \Big], \end{aligned} \tag{21}$$

where $d = \|\mathbf{o} - \boldsymbol{\mu}\|$, θ_o is the angle between $\boldsymbol{\omega}$ and $\mathbf{o} - \boldsymbol{\mu}$, and c_G is the integral of the 3D Gaussian given in Eq. (20).

4.1.3 Directional sampling. To sample directions from a 3D Gaussian from a specific observation point, we sample a 3D point from the 3D Gaussian. The distribution of direction pointing to it from the observation point obeys Eq. (21)

4.1.4 Culling for bipolar-cone projection. We use a specific culling function for a 3D Gaussian. For the mean position $\boldsymbol{\mu}$, we project it to the lens plane as $\boldsymbol{\mu}_l$, and we cull the 3D Gaussian as a whole:

$$C_G(\mathbf{x} \mid \mathbf{x}_{\mathcal{F}}) = \begin{cases} 1, & \text{if } \|\mathbf{x}_c - \boldsymbol{\mu}_l\| < r + \sigma \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

4.1.5 Using 3D Gaussian mixture. A radiance field can be regarded as a 3D Gaussian mixture model (GMM). Despite we apply BCP to it, we sample and calculate the PDF like conventional GMMs.

4.2 Iterative Fitting of the Radiance Field

Now we describe the workflow to fit the 3D-Gaussian radiance field using pixel samples, progressively during rendering, as shown in Fig. 5. The key idea is that we build a record buffer based on pixel samples and extract 3D Gaussians based on it.

4.2.1 Record buffer construction. We construct a *fixed-size* 2D record buffer, where each element stores auxiliary data, including position, radiance, and sample count. In each path tracing iteration, we generate camera rays using the thin-lens model and reproject the primary hit positions into raster space as if using a pinhole camera. We then accumulate the auxiliary data into the corresponding element.

4.2.2 Spotlight discovery. To ensure the final collection of 3D Gaussians compact, we identify the interested elements, which represent the "spotlights", within the record buffer. We apply a filter to select elements with high radiance that are significantly offset from the focal plane. To determine whether a point is *significantly offset* from the focal plane, we compute the size of the CoC for a point at a

given distance using the thin lens formula for circle of confusion diameter, adapted to our notation:

$$\mathrm{CoC} = \left| \frac{2r \cdot (d_{\mathbf{p}} - d_{\mathcal{F}})}{d_{\mathbf{p}} \cdot (d_{\mathcal{F}} - d_s)} \cdot d_s \right|, \tag{23}$$

where r is the lens radius, d_s is the distance between the sensor plane and the lens, $d_{\mathcal{F}}$ is the distance between the focal plane and the lens, and $d_{\mathbf{p}}$ is the distance from the point to the lens plane. We preserve a buffer element when the CoC exceeds a manually set threshold, indicating that the point is significantly defocused. These filtered elements correspond to areas of high visual importance.

4.2.3 Element labeling and merging. Treating every buffer element as an independent Gaussian would result in an unnecessarily large collection. Instead, we label the high-energy elements and merge 2D-connected ones into patches, using the algorithm by Rosenfeld and Pfaltz [1966].

4.2.4 Conversion from patches to 3D Gaussians. After identifying the high-energy patches, we approximate each as a 3D Gaussian. First, we compute a bounding sphere for the entire patch by combining smaller AABBs, one for each element in the record buffer. The center of each element's AABB is its recorded position, and its extent is the approximated width of a pixel at the given distance:

$$w = \frac{s \cdot t \cdot \cos\theta}{d_s}. \tag{24}$$

Where s is the sensor-space width and t is the recorded depth. The mean $\boldsymbol{\mu}$ of the Gaussian is set to the center of the AABB, and the standard deviation $\sigma = \alpha R_b$ is determined by the bounding sphere radius R_b, where α is a hyperparameter. A small value of α results in more concentrated samples but also more outliers. In practice, we set $\alpha = 0.5$.

5 Refinement Extensions for Local Adaptation

Although a naive application of BCP with 3D Gaussians can already guide lens sampling effectively, there are still two major limitations related to parallax and occlusion handling. These limitations arise because 3D Gaussians represent the radiance field in a view-independent manner and we assume it's constantly visible.

We propose two extensions to refine the post-BCP distribution, mitigating these issues and improving the overall sampling efficiency. The key idea is that with fitted parameterized distribution $p(x;\phi)$, we further alter it to $q(x;\gamma)$ in some way so that is a closer fit to the radiance L (note that γ does not have to be in the same form of ϕ). To achieve this, we adopt the learning technique proposed by Müller et al. [2019]. For each sampled radiance L, we optimize the one-sample estimate of Kullback-Leibler divergence (KL-div) using the *Adam* [Kingma and Ba 2014] optimizer:

$$\nabla_\gamma D_{KL}(L(\mathbf{x}_l) \| q(\mathbf{x}_l;\gamma)) = -\mathbb{E}\left[L \nabla_\gamma \log[q(\mathbf{x};\gamma)]\right]. \tag{25}$$

Each of the extensions described below employs this optimization technique but with different forms of $q(\mathbf{x};\gamma)$.

5.1 Blending Weight Optimization

At pixels where the fitted 3D Gaussian is occluded at a given $\mathbf{x}_l$, importance sampling the radiance field yields a degenerate result compared to uniform sampling. To address this, we blend the fitted distribution with uniform $p_U(\mathbf{x}_l)$ using a blending weight β:

$$q(p_l) = \beta \hat{p}'(\mathbf{x}_l;\phi) + (1-\beta) p_U(\mathbf{x}_l). \tag{26}$$

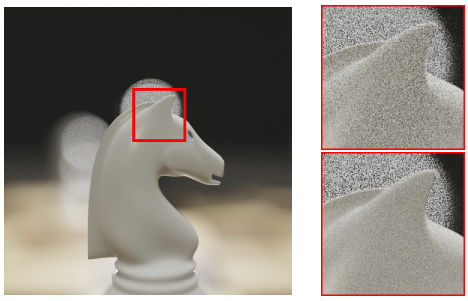


Fig. 6. Sampling efficiency with optimized blending weight. Left: ground truth with occluded highlights marked. Right: sampling with fixed $\beta = 0.95$ (top) and online-optimized β (bottom). A fixed high β reduces background CoC variance but undersamples the foreground. Optimizing β adaptively mitigates occlusion variance for more balanced and efficient sampling.

Inspired by previous work [Huang et al. 2024a; Müller et al. 2017], we optimize β per pixel, rather than using a fixed value. This approach identifies occlusion by driving β to a low value when the fitted spot is occluded, favoring uniform distribution instead. Fig. 6 illustrates how the optimized blending weight improves sampling efficiency with partially occluded 3D Gaussians. An additional buffer (12 bytes per pixel) is required to store and optimize β.

5.2 Polynomial 3D Gaussians

The 3D Gaussian fitting described in § 4.2 assumes that highlight spots are view-independent; however, in practice, many highlights are view-dependent (e.g., glossy or refracted surfaces), causing parallax issues. We address this by applying a dynamic offset to the 3D Gaussians' μ, which adapts to the current sampled sensor point:

$$\mu' = \mu + D(\mathbf{x}_r). \tag{27}$$

Here, $\mathbf{x}_r = (x_{r,x}, x_{r,y})$ represents the rasterization-space 2D position of the sensor point $\mathbf{x}_s$. The offset $D(\mathbf{x}_r)$ is a 3D vector derived from $\mathbf{x}_r$, reflecting the view-dependent nature of the highlight spots. As a special case, $D(\mathbf{x}_r) = \mathbf{0}$ represents view-independent 3D Gaussians.

$D(\mathbf{x}_r)$ should be parameterized, compact, flexible, and expressive. We use polynomials, which satisfy all these requirements. Let

$$D(\mathbf{x}_r) = \left(D_x(\mathbf{x}_r), D_y(\mathbf{x}_r), D_z(\mathbf{x}_r)\right). \tag{28}$$

Each componentis an independent polynomial:

$$D_c(\mathbf{x}_r) = \sum_{i+j \le n} a_{ij}^{(c)} x_{r,x}^i x_{r,y}^j, \quad c \in \{x, y, z\}. \tag{29}$$

Here, n is the polynomial degree, and $\{a_{ij}^{(c)}\}$ are the polynomial coefficients for each dimension c. This formulation ensures that $D(\mathbf{x}_r)$ is a flexible, compact, and expressive representation of the offset, dynamically adapting to view-dependent effects. In our practice, we found that a low degree $n = 2$ is significant.

Each 3D Gaussian is paired with polynomial parameters that adapt to the sensor's position and view-dependent effects. During lens sampling, we select the 3D Gaussian with the highest PDF for

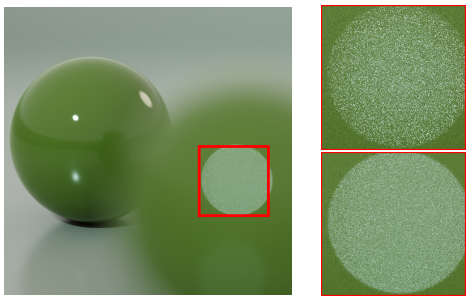


Fig. 7. Rendering a specular surface with 256 SPP. Left: ground truth showing view-dependent highlights. Right: lens sampling using a fixed 3D Gaussian (top) versus a polynomial-offset 3D Gaussian (bottom). The polynomial offset better captures parallax, reducing noise and improving convergence.

the sampled ray and optimize its parameters based on the associated radiance. Fig. 7 compares rendering results with fixed and polynomial-offset 3D Gaussians, highlighting the improved efficiency in handling view-dependent highlights.

6 Experiments

6.1 Implementation Details

We implement our method in Smaray, a GPU production renderer. Our method requires a 68MB buffer to store the scratch and result data for fitting. Our radiance field fitting and guiding adds a constant overhead of 15ms/SPP when rendering a 1920×1080 image, with an additional 2ms for blending weight optimization and 20ms for polynomial optimization. The online optimization for blending weight and polynomials is implemented with the auto-differentiation feature of LuisaCompute [Tong et al. 2023; Zheng et al. 2022], with constant learning rates of 0.1 and 0.01, respectively. We use fixed blending weight $\beta = 0.8$ when experimenting without optimization.

Since we use a trivial culling function in our implementation, when sampling the 3D Gaussians for lens, a sample outside the lens disk can still be generated. In such cases, we accumulate a zero-value sample in radiance estimation to ensure unbiasedness. At pixels where all 3D Gaussians are culled (indicating no high-energy spots), the distribution falls back to uniform.

The global 3D Gaussian fitting can be performed at each rendering iteration; however, the polynomial offset for 3D Gaussians requires a static initial set. When enabling it, we only perform global fitting for the first 8 iterations. Afterwards, the Gaussians are fixed, and the polynomials are optimized at each iteration.

6.2 Rendering Results

To evaluate our method, we rendered a set of defocused scenes at 256 samples per pixel (SPP) and compared the results against high-sample reference images. All tests were conducted on a conventional PC with an Intel i7-9700K CPU, an NVIDIA RTX 4070 GPU, and 56GB of DDR4 memory. The evaluation metric is mean absolute percentage error (MAPE), computed on raw outputs to reflect the quality of the rendered images. While computational performance, in terms of rendering time, is relevant in certain contexts, the raw performance is heavily influenced by implementation details, particularly with GPU path tracers. Thus, we focus on sampling efficiency over computation times in this study.

Area ReSTIR [2024] improves lens sampling efficiency through path resampling. Although this technique is orthogonal to our approach, we implemented and compared it within the same GPU renderer for fair evaluation. Since offline rendering lacks the "temporal" variance present in real-time contexts, we limited the implementation to spatial reuse. To ensure robust comparison, we increased the resampling budget to 32 neighboring pixels (8× the original implementation) with a gather radius of 50 pixels. Adjustments were made for compatibility while remaining faithful to the core methodology. The final implementation demonstrates similar performance to our method with the 2 extensions.

6.2.1 Variance reduction. With the same SPP, our method significantly improves rendering quality at CoC regions without affecting other areas. With a shallow DoF, uniform sampling struggles to capture the CoCs, as shown in Fig. 8 (a) and Fig. 8 (d). In contrast, our method reduces the error by over 75%. Efficiency gains are demonstrated in the convergence graph in Fig. 9: our method accelerates error reduction, and the error quickly converges to a lower level.

6.2.2 Blending weight optimization. Using a fixed blending weight causes degenerate sampling at the edge area of a CoC, or when the highlight point is actually occluded by other objects at a different lens path. In Fig. 8 (a), the red CoC is partially occluded by the machine, and the quality of our naive approach shows poorer result than uniform method, and it shows a clear circle pattern. This is where our blending weight optimization excels.

6.2.3 Polynomial finetuning for indirect highlights. In Fig. 1, the glossy ball on the foreground shows a large CoC. The actual specular point is view-dependent, therefore, fitting a fixed region as 3D Gaussian suffers from parallax issue. Our polynomial finetuning strategy adapts the global 3D Gaussians to local ones, enabling close approximation of the actual location for view-dependent highlights.

6.2.4 Comparison with Area ReSTIR. Quantitatively, our method consistently outperforms Area ReSTIR in convergence across all test scenes, particularly in regions with large CoCs, where spatial reuse efficiency declines. Further increasing the resampling budget leads to diminishing returns and introduces performance bottlenecks. In Fig. 1 scene, our approach successfully handles view-dependent highlights, which Area ReSTIR struggles with. We also observe some regions, typically with large-area bright surfaces, where Area ReSTIR's result is more converged. A sophisticated implementation combining both methods is promising and left for future work.

6.3 Distribution

We evaluate the distributions fitted by different methods for a specific pixel during rendering (see Fig. 10 (a)). The lens space of the center pixel is rasterized into a 128×128 grid, with the radiance at each texel recorded as the reference distribution. Our method's actual distribution is approximated by sampling 2^{16} rays and histogramming the results at texels (Fig. 10 (c)). The same process is applied to Area ReSTIR for comparison.

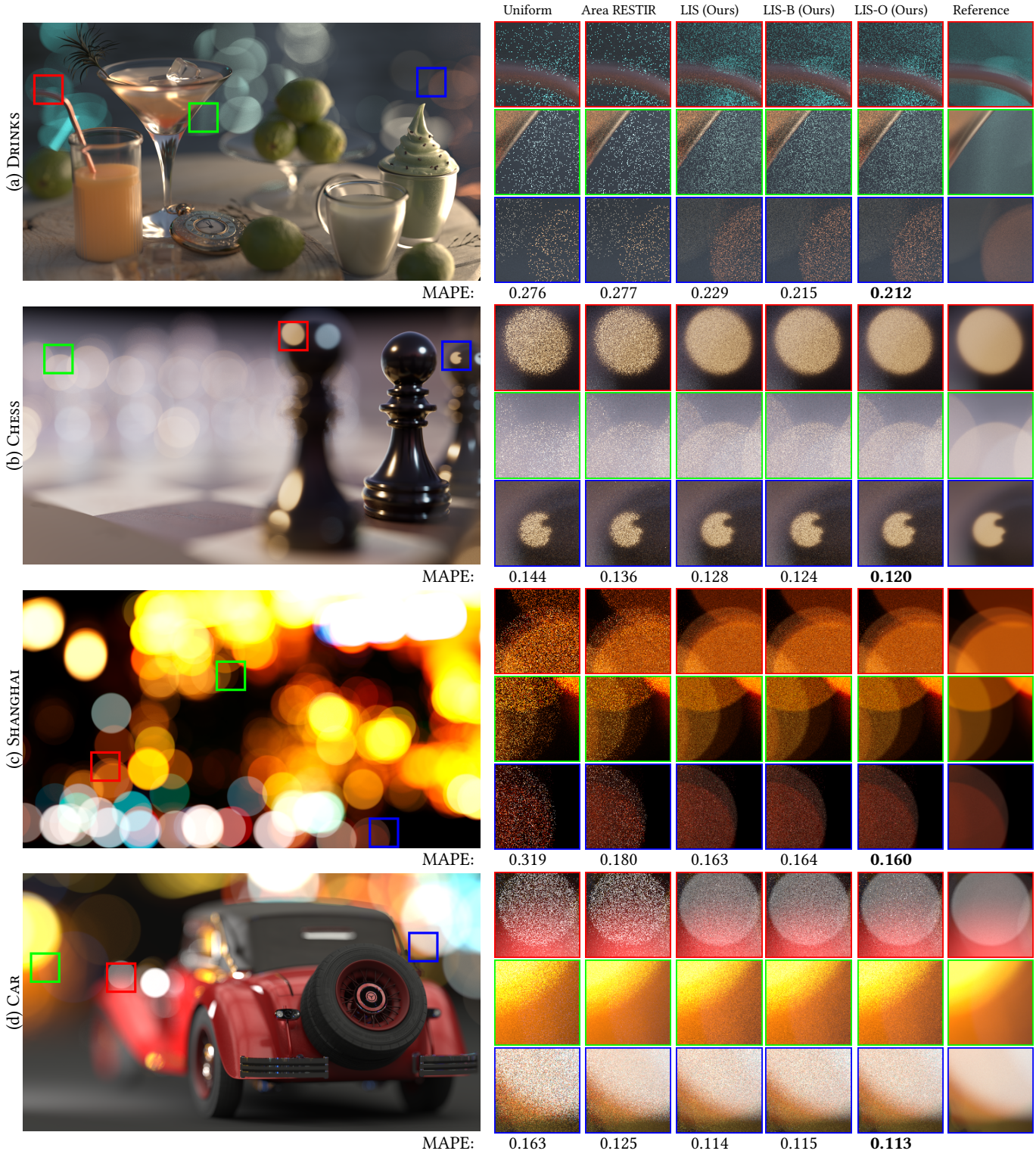
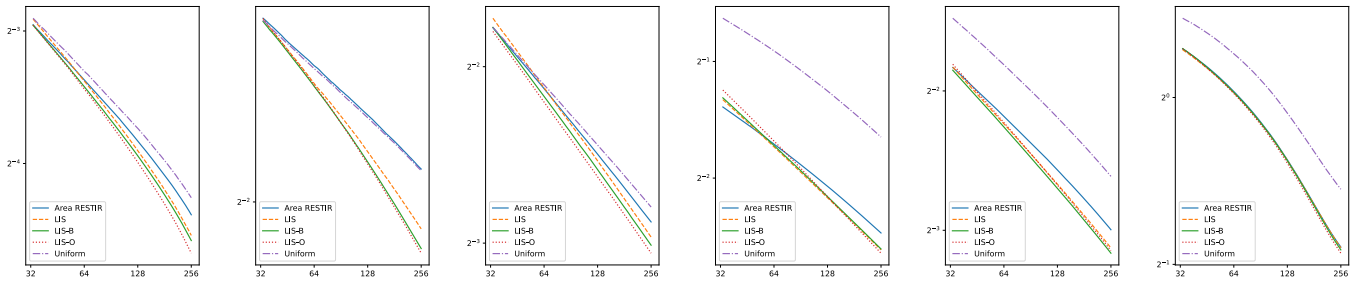


Fig. 8. Direct comparison of rendering results between our guided lens sampling (LIS) and two previous methods: uniform disk and Area ReSTIR [2024]. LIS is evaluated in three versions: the basic implementation with 3D Gaussian fitting (LIS), with optimized blending weight (LIS-B), and with full online optimization (LIS-O). Our method shows significant improvements in rendering defocused circles of confusion (CoCs), achieving lower mean absolute percentage error (MAPE) across various scenes, as shown in the detailed close-ups for each sample region.



(a) Ball (b) Drinks (c) Chess (d) Shanghai (e) Car (f) Zero Day

Fig. 9. Convergence graph of rendering using our method (LIS, LIS-B, and LIS-O) and uniform sampling. Horizontal axis represents sample per pixel and Vertical axis represents mean absolute percentage error (MAPE) calculated with raw rendering results. LIS quickly improves sampling efficiency, while after steps of fine-tuning, LIS-O further accelerates the convergence. The improvement introduced by LIS-O is more obvious at where specular highlight spots exists, such as (a) and (c).

KL-div calculations show that our method aligns more closely with the reference distribution. In contrast, Area ReSTIR exhibits a greater deviation, likely due to its search radius (50 pixels) being smaller than the CoCs, limiting sample exploration. Increasing the radius may mitigate this but reduces the probability of finding high-contribution samples, as only 32 are resampled.

6.4 Failure Cases and Limitations

The 3D Gaussians are generated from a 2D record buffer, meaning only one Gaussian can be created per texel. When two highlight spots at different depths project onto the same texel, a conflict arises, leading to degraded results (see Fig. 11). This could be addressed by using a 3D representation, like a voxel grid, but overlapping highlight spots at different depths are rare. We consider a 3D record buffer worthwhile only in cases of significant spatial overlap, as it introduces additional computational cost in both highlight detection and component merging.

The alpha estimation for each pixel is biased due to out-of-lens samples drawn from 3D Gaussians (see Fig. 12). While we can accumulate zero for radiance estimation in these cases, alpha cannot be estimated the same way. In production rendering, alpha values can be independently approximated, offering a practical workaround.

7 Conclusion and Future Work

We proposed a new guided lens sampling method. We reviewed the measurement equation to formulate lens sampling and derived bipolar-cone projection to convert a world-space radiance field to the lens-space sampling distribution. Our practical implementation, along with two extensions, showed that our method significantly improves the rendering efficiency of CoCs, compared to the status quo of standard uniform sampling, and also outperforms the path resampling method, through real-world rendering experiments.

Our work is orthogonal to existing techniques related to CoCs, such as path resampling [Zhang et al. 2024] and bidirectional rendering [Georgiev et al. 2012; Veach and Guibas 1995a]. Combining our approach with these methods offers an interesting research direction. Beyond the thin-lens model, our method could also be extended to more realistic lens systems, potentially through manifold exploration [Hanika et al. 2015].

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Project No. 62372257).

References

Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. 2020. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Trans. Graph.* 39, 4, Article 148 (Aug. 2020), 17 pages. doi:10.1145/3386569.3392481

Robert L. Cook, Thomas Porter, and Loren Carpenter. 1984. Distributed ray tracing. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '84)*. Association for Computing Machinery, New York, NY, USA, 137–145. doi:10.1145/800031.808590

Ana Dodik, Marios Papas, Cengiz Öztireli, and Thomas Müller. 2022. Path Guiding Using Spatio-Directional Mixture Models. *Computer Graphics Forum* (2022). doi:10.1111/cgf.14428

Manfred Ernst, Marc Stamminger, and Gunther Greiner. 2006. Filter Importance Sampling. In *2006 IEEE Symposium on Interactive Ray Tracing*. 125–132. doi:10.1109/RT.2006.280223

Iliyan Georgiev, Jaroslav Křivánek, Tomas Davidovic, and Philipp Slusallek. 2012. Light transport simulation with vertex connection and merging. *ACM Trans. Graph.* 31, 6 (2012), 192–1.

Johannes Hanika, Marc Droske, and Luca Fascione. 2015. Manifold next event estimation. In *Computer graphics forum*, Vol. 34. Wiley Online Library, 87–97.

Jiawei Huang, Akito Iizuka, Hajime Tanaka, Taku Komura, and Yoshifumi Kitamura. 2024a. Online Neural Path Guiding with Normalized Anisotropic Spherical Gaussians. *ACM Trans. Graph.* 43, 3, Article 26 (apr 2024), 18 pages. doi:10.1145/3649310

Jiawei Huang, Hajime Tanaka, Taku Komura, and Yoshifumi Kitamura. 2024b. Online Photon Guiding with 3D Gaussians for Caustics Rendering. arXiv:2403.03641 [cs.GR] https://arxiv.org/abs/2403.03641

Matthias B Hullin, Johannes Hanika, and Wolfgang Heidrich. 2012. Polynomial Optics: A construction kit for efficient ray-tracing of lens systems. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 1375–1383.

Andrey Ignatov, Jagruti Patel, and Radu Timofte. 2020. Rendering Natural Camera Bokeh Effect with Deep Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 1676–1686. doi:10.1109/CVPRW50498.2020.00217

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014). http://arxiv.org/abs/1412.6980

Craig Kolb, Don Mitchell, and Pat Hanrahan. 1995. A realistic camera model for computer graphics. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 317–324.

Christopher Kulla and Marcos Fajardo. 2012. Importance Sampling Techniques for Path Tracing in Participating Media. *Computer Graphics Forum* 31, 4 (2012), 1519–1528. doi:10.1111/j.1467-8659.2012.03148.x arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2012.03148.x

Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. 2004. Efficient BRDF importance sampling using a factored representation. In *ACM Transactions on Graphics (TOG)*, Vol. 23. ACM, 496–505.

Thomas Leimkühler, Hans-Peter Seidel, and Tobias Ritschel. 2018. Laplacian kernel splatting for efficient depth-of-field and motion blur synthesis or reconstruction. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–11.

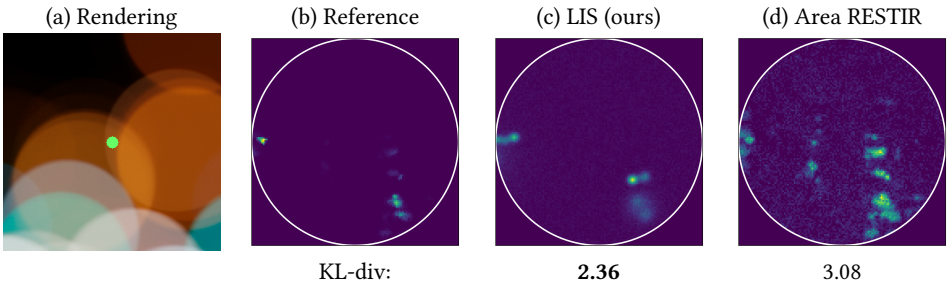


Fig. 10. Comparison of lens distributions for a specific pixel in the scene. The rendering result highlights the focus region (a). The reference distribution (b) is generated by rasterizing the lens space into a 128×128 grid and recording radiance per texel. Our method's distribution (c) is approximated by sampling 2^{16} rays and histogramming the results at texels. Area RESTIR's distribution (d) is similarly approximated. KL-divergence values indicate that our method (LIS) achieves a closer fit to the reference distribution compared to Area RESTIR.

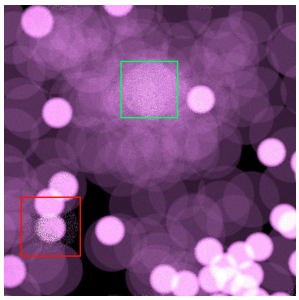


Fig. 11. Defocused rendering with numerous tiny bright spheres. In the color-boxed areas, overlapping highlight spots at different depths project onto the same texels, causing undersampling and degraded results. This limitation arises from the use of a 2D record buffer to generate 3D Gaussians, which cannot distinguish overlapping highlights at varying depths.

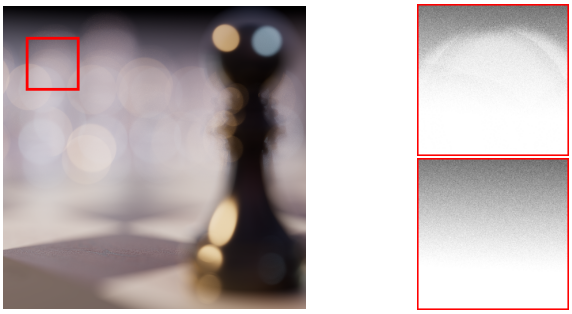


Fig. 12. Alpha estimation bias caused by out-of-lens samples drawn from 3D Gaussians. When rendering with an empty background and non-unity alpha (left), our method (top right) produces biased alpha values, resulting in circular-pattern artifacts. Compared to the reference (bottom right), these artifacts arise due to out-of-range samples drawn from the 3D Gaussians. In production rendering, alpha values can be approximated independently to mitigate this issue efficiently.

Daqi Lin, Markus Kettunen, Benedikt Bitterli, Jacopo Pantaleoni, Cem Yuksel, and Chris Wyman. 2022. Generalized resampled importance sampling: foundations of ReSTIR. *ACM Trans. Graph.* 41, 4, Article 75 (July 2022), 23 pages. doi:10.1145/3528223.3530158

Haolin Lu, Wesley Chang, Trevor Hedstrom, and Tzu-Mao Li. 2024. Real-Time Path Guiding Using Bounding Voxel Sampling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 43, 4 (2024). doi:10.1145/3658203

Xianrui Luo, Juewen Peng, Ke Xian, Zijin Wu, and Zhiguo Cao. 2023. Defocus to focus: Photo-realistic bokeh rendering by fusing defocus and radiance priors. *Information Fusion* 89 (2023), 320–335. doi:10.1016/j.inffus.2022.08.023

Don P Mitchell and Arun N Netravali. 1988. Reconstruction filters in computer graphics. *ACM SIGGRAPH Computer Graphics* 22, 4 (1988), 221–228.

Thomas Müller, Markus Gross, and Jan Novák. 2017. Practical Path Guiding for Efficient Light-Transport Simulation. *Comput. Graph. Forum* 36, 4 (jul 2017), 91–100. doi:10.1111/cgf.13227

Thomas Müller, Brian Mcwilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. 2019. Neural Importance Sampling. *ACM Trans. Graph.* 38, 5, Article 145 (oct 2019), 19 pages. doi:10.1145/3341156

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2023. *Physically based rendering: From theory to implementation.* MIT Press.

Michael Potmesil and Indranil Chakravarty. 1981. A lens and aperture camera model for synthetic image generation. In *Proceedings of the 8th Annual Conference on Computer Graphics and Interactive Techniques* (Dallas, Texas, USA) *(SIGGRAPH '81)*. Association for Computing Machinery, New York, NY, USA, 297–305. doi:10.1145/800224.806818

Azriel Rosenfeld and John L. Pfaltz. 1966. Sequential Operations in Digital Picture Processing. *J. ACM* 13, 4 (Oct. 1966), 471–494. doi:10.1145/321356.321357

Lukas Ruppert, Sebastian Herholz, and Hendrik P. A. Lensch. 2020. Robust Fitting of Parallax-Aware Mixtures for Path Guiding. *ACM Trans. Graph.* 39, 4, Article 147 (aug 2020), 15 pages. doi:10.1145/3386569.3392421

Justin F. Talbot, David Cline, and Parris Egbert. 2005. Importance resampling for global illumination. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques* (Konstanz, Germany) *(EGSR '05)*. Eurographics Association, Goslar, DEU, 139–146.

Xiaochun Tong, Hsueh-Ti Derek Liu, Yotam Gingold, and Alec Jacobson. 2023. Differentiable Heightfield Path Tracing with Accelerated Discontinuities. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 19, 9 pages. doi:10.1145/3588432.3591530

Eric Veach. 1998. Robust monte carlo methods for light transport simulation. *PhD thesis, Stanford University* (1998).

Eric Veach and Leonidas Guibas. 1995a. Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques*. Springer, 145–167.

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

Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. 2014. On-Line Learning of Parametric Mixture Models for Light Transport Simulation. *ACM Trans. Graph.* 33, 4, Article 101 (jul 2014), 11 pages. doi:10.1145/2601097.2601203

Song Zhang, Daqi Lin, Markus Kettunen, Cem Yuksel, and Chris Wyman. 2024. Area ReSTIR: Resampling for Real-Time Defocus and Antialiasing. *ACM Trans. Graph.* 43, 4, Article 98 (July 2024), 13 pages. doi:10.1145/3658210

Shaokun Zheng, Zhiqian Zhou, Xin Chen, Difei Yan, Chuyan Zhang, Yuefeng Geng, Yan Gu, and Kun Xu. 2022. LuisaRender: A High-Performance Rendering Framework with Layered and Unified Interfaces on Stream Architectures. *ACM Trans. Graph.* 41, 6, Article 232 (nov 2022), 19 pages. doi:10.1145/3550454.3555463

M. Zwicker, W. Jarosz, J. Lehtinen, B. Moon, R. Ramamoorthi, F. Rousselle, P. Sen, C. Soler, and S.-E. Yoon. 2015. Recent Advances in Adaptive Sampling and Reconstruction for Monte Carlo Rendering. *Comput. Graph. Forum* 34, 2 (May 2015), 667–681.