Hierarchical Light Sampling with Accurate Spherical Gaussian Lighting

YUSUKE TOKUYOSHI, Advanced Micro Devices, Inc., Japan
SHO IKEDA, Advanced Micro Devices, Inc., Japan
PARITOSH KULKARNI*, Advanced Micro Devices, Inc., Canada
TAKAHIRO HARADA, Advanced Micro Devices, Inc., USA

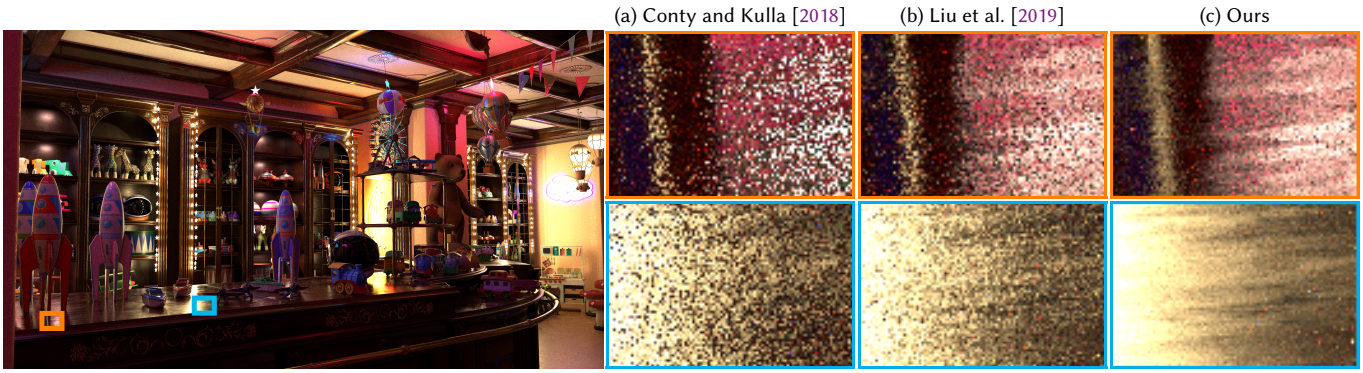


Fig. 1. Path tracing (10 s) with previous hierarchical light sampling methods (a, b) and our method (c) using one light sample per tree-traversal query (3840×2160 pixels, AMD Radeon™ RX 7900 XTX GPU). Our light sampling method significantly reduces undesirable noise (i.e., Monte Carlo variance), especially for glossy highlights at grazing angles, by accounting for the anisotropic scattering characteristics of microfacet BRDFs.

Importance sampling using a light tree (i.e., a hierarchy of light clusters) has been widely used for many-light rendering. This technique samples a light source by stochastically traversing the tree according to the *importance* of each node. While this importance should be close to the illumination integral for each node's light cluster, it is infeasible to compute the exact solution. Therefore, existing methods used a rough approximation (e.g., upper bound), which results in significant Monte Carlo (MC) variance, especially for high-frequency microfacet BRDFs at grazing angles. In this paper, we present a more accurate approximation of the importance based on *spherical Gaussians* (SGs). Our method represents a light cluster with an SG light for each node, and analytically approximates the product integral of the SG light and a BRDF. Although high-quality SG lighting approximations have been studied, they could not be used for the node importance due to violations of an unbiased sampling constraint. To improve the sampling quality and satisfy the constraint for anisotropic microfacet BRDFs, we introduce a new high-quality SG lighting approximation by extending an *NDF filtering* method that has been used for specular antialiasing. For diffuse surfaces, we also present a simpler and more accurate SG lighting than the state-of-the-art SG approximation, satisfying the constraint. Using our method, we can efficiently reduce the MC variance for many-light scenes with modern physically plausible materials.

CCS Concepts: • **Computing methodologies** → **Ray tracing**.

Additional Key Words and Phrases: importance sampling, many lights, microfacet BRDF, NDF filtering, spherical Gaussians, von Mises–Fisher distribution

ACM Reference Format:
Yusuke Tokuyoshi, Sho Ikeda, Paritosh Kulkarni, and Takahiro Harada. 2024. Hierarchical Light Sampling with Accurate Spherical Gaussian Lighting. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24), December 3–6, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3680528.3687647

1 Introduction

Hardware ray tracing and Monte Carlo (MC) integration are widely used for photorealistic rendering nowadays. However, for scenes with many emissive objects, developing an efficient direct illumination algorithm is still a challenging problem. Since it is infeasible to trace shadow rays and evaluate contributions for all the lights, we should sample an important light source efficiently. The ideal light sampling is according to the product of the incoming radiance from each light and the bidirectional reflectance distribution function (BRDF) at a shading point, but the exact solution of this product importance sampling is also infeasible. In addition, modern physically plausible materials further complicate the problem due to the high-frequency and anisotropic scattering characteristics of microfacet BRDFs [Cook and Torrance 1982]. In this paper, we tackle this product importance sampling problem for all-frequency BRDFs.

*Paritosh Kulkarni is now at Qualcomm.

Authors' Contact Information: Yusuke Tokuyoshi, Advanced Micro Devices, Inc., Japan, yusuke.tokuyoshi@amd.com; Sho Ikeda, Advanced Micro Devices, Inc., Japan, sho.ikeda@amd.com; Paritosh Kulkarni, Advanced Micro Devices, Inc., Canada; Takahiro Harada, Advanced Micro Devices, Inc., USA, takahiro.harada@amd.com.

SA Conference Papers '24, December 3–6, 2024, Tokyo, Japan

Importance sampling using a light tree (i.e., a hierarchy of light clusters) [Conty and Kulla 2018] is widely used for many-light rendering nowadays. This method samples a light source by stochastically traversing the tree according to the *importance* of each node. By using an importance value close to the illumination integral for each light cluster, we can approximately perform product importance sampling. However, previous sampling methods used rough approximations for the importance, such as using upper bounds, ignoring BRDFs [Conty and Kulla 2018; Lin and Yuksel 2020; Yuksel 2021], and representing anisotropic reflections with isotropic lobes [Liu et al. 2019]. Since these approximation errors were particularly large near the root node, previous methods had to adaptively split light clusters during tree traversal to reduce the MC variance. Such adaptive splitting generates multiple light samples per tree-traversal query, increasing computational cost and implementation complexity. Efficient splitting is also a challenging problem as studied in lightcuts-based methods [Huo et al. 2020; Nabata et al. 2016; Vévoda et al. 2018; Walter et al. 2006, 2005; Wang et al. 2021].

To render high-quality images even with one light sample per tree-traversal query (Fig. 1), we introduce an accurate importance approximation based on *spherical Gaussians* (SGs) [Tsai and Shih 2006]. SGs have often been used to approximate all-frequency illumination integrals [Wang et al. 2009; Xu et al. 2013]. However, existing accurate SG lighting approximations cannot be used for light-tree-based hierarchical importance sampling because they can violate the constraint of unbiased importance sampling. These SG approximations can produce zero (or negative) importance due to approximation error, and thus the probability distribution of samples cannot cover the integral domain. To use SGs for hierarchical importance sampling, this paper introduces new high-quality SG lighting approximations that satisfy the constraint.

Our SG lighting for anisotropic microfacet BRDFs is based on a filtering method for a microfacet normal distribution function (NDF) [Kaplanyan et al. 2016]. While this NDF filtering was developed for specular antialiasing, we extend the filtering method for SG lighting. Our method increases the roughness parameter while preserving the NDF model to approximate the SG lighting integral. Therefore, we can obtain accurate importance, especially for NDFs with longer tails than the SG, such as the GGX NDF [Trowbridge and Reitz 1975; Walter et al. 2007]. In this paper, we also present a new approximation for the product integral of an SG and a clamped cosine for diffuse SG lighting, which is simpler and more accurate than the state-of-the-art approximations [Meder and Brüderlin 2018; Tokuyoshi 2022] and satisfies the unbiased sampling constraint. By using our SG lighting methods for the importance of each node, we can approximately perform product importance sampling with less error than the previous importance approximations.

Our contributions are as follows.

- We propose a light sampling method using an SG light tree whose nodes have SG lighting-based importance (§ 3). Our light cluster representation in each node is more compact than the previous representation [Conty and Kulla 2018].
- To improve the importance approximation and avoid a bias for diffuse BRDFs, we introduce a simpler and more accurate approximation than the state-of-the-art method for the product integral of an SG and the clamped cosine term (§ 4).
- For anisotropic microfacet BRDFs, we introduce a high-quality and numerically stable SG lighting approximation based on NDF filtering, while avoiding a bias for importance sampling (§ 5). This glossy SG lighting is the main contribution to improve the sampling quality.
- We demonstrate the efficiency of our method for one light sample per tree-traversal query in experimental results (§ 6).

2 Background

2.1 Related Work

Many-Light Rendering. Rendering using a light tree has been studied in many-light methods [Dachsbacher et al. 2014] for direct illumination and indirect illumination using virtual point lights [Keller 1997]. One such method referred to as lightcuts [Walter et al. 2006, 2005] hierarchically splits light clusters based on an estimated error bound [Huo et al. 2020; Nabata et al. 2016] or adaptive clustering [Vévoda et al. 2018; Wang et al. 2021], and then uses a representative light in each found cluster. Conty and Kulla [2018] developed a practical importance sampling method using a light tree, taking multiple importance sampling (MIS) [Veach and Guibas 1995] into account for next event estimation in path tracing. Yuksel [2021] and Lin [2020] avoided the correlation of MC variance in lightcuts by stochastically sampling representative lights as in Conty and Kulla [2018]. The effectiveness of the light-tree-based importance sampling was demonstrated by Moreau et al. [2019] for real-time rendering. Another many-light sampling method orthogonal to light trees is spatiotemporal reservoir resampling (ReSTIR) [Bitterli et al. 2020] that reuses candidate samples from spatiotemporal neighbors. Although ReSTIR massively increases the number of candidates for resampled importance sampling [Lin et al. 2022; Talbot 2005], its variance is proportional to the initial candidate sampling technique. To reduce the ReSTIR variance, hierarchical light sampling can be used for initial candidate generation. In this paper, we improve the hierarchical light sampling by using SGs and NDF filtering.

Spherical Gaussians (SGs). SGs based on the von Mises–Fisher (vMF) distribution [1953] are often used to approximate the illumination integral for environment maps [Tsai and Shih 2006], light maps [Currius et al. 2020; Neubelt and Pettineo 2015], area lights [Wang et al. 2009], and virtual lights [Tokuyoshi 2015; Xu et al. 2014]. This is because they have closed-form solutions for the integral, product, and product integral, which are fundamental operations to evaluate illumination integrals. Xu et al. [2013] introduced anisotropic SGs (ASGs) based on the Bingham distribution [1974] and approximated the product and integral for ASGs. For microfacet BRDFs under SG lights, they approximated the reflection lobe with an ASG to convolve it with an SG light. However, the ASG understates its lower hemisphere to zero. It also produces undesirable distortion and underestimation for the ASG reflection lobe (Fig. 6a). Huang et al. [2024] introduced normalized ASGs for neural path guiding, but the product integral to approximate illumination was not presented. For diffuse surfaces, Pettineo and Hill [2016] fitted the

irradiance from an SG light. However, it produced a noticeable negative error in corner cases. Meder and Brüderlin [2018] introduced a more robust approximation than the fitted irradiance by using the hemispherical integral of an SG. The quality and performance of their method were then improved by Tokuyoshi [2022], but the diffuse SG lighting could still produce a small negative value due to the approximation error. These underestimation errors introduce a bias in importance sampling using SG approximations.

NDF Filtering. Parametric NDFs, such as the GGX and Beckmann [1963] distributions, are widely used for microfacet BRDFs in the computer graphics industry. Filtering of such NDFs was introduced for specular antialiasing [Hill and Baker 2012; Kaplanyan et al. 2016; Tokuyoshi and Kaplanyan 2021]. This NDF filtering increases the roughness parameter by approximately convolving the NDF with a pixel footprint in halfvector space. We extend this filtering method for SG lighting by projecting an SG light to halfvector space. There was little work on empirically increasing the roughness to approximate area lighting [Karis 2013], which ignored the transformation between halfvectors and light directions. Our method takes the transformation into account by deriving its Jacobian matrix.

2.2 Importance Sampling Using a Light Tree

Our method is built upon an importance sampling method using a light tree [Conty and Kulla 2018]. This tree-based method clusters lights hierarchically and stores a light cluster in each node during preprocessing. To sample a light, the method traverses the light tree by stochastically selecting a child node in a 1D manner of hierarchical sample warping [Clarberg et al. 2005; McCool and Harwood 1997]. For a binary tree, the probability of selecting a left child is $I_{\text{left}}/(I_{\text{left}} + I_{\text{right}})$, where $I_{\text{left}} \in [0, \infty)$ and $I_{\text{right}} \in [0, \infty)$ are the importance of the left and right child nodes, respectively. To reduce the MC variance, this importance should be approximately proportional to the integral of illumination from the light cluster:

$$I \propto \int_{S^2} L(\mathbf{x}, \mathbf{o}) f(\mathbf{i}, \mathbf{o}) |\mathbf{o} \cdot \mathbf{n}| d\mathbf{o}.$$

For notations used in this paper, please see Table 1. For unbiased rendering, this importance must satisfy the following constraint:

$$I > 0 \quad \text{if} \int_{S^2} L(\mathbf{x}, \mathbf{o}) f(\mathbf{i}, \mathbf{o}) |\mathbf{o} \cdot \mathbf{n}| d\mathbf{o} > 0. \tag{1}$$

Therefore, an efficient approximation that satisfies this constraint is the key to improving the hierarchical light sampling.

Conty and Kulla [2018] approximated the BRDF with the Lambert model, and used the upper bounds of cosine terms by representing the light cluster using an axis-aligned bounding box (AABB) and bounding cones. They also approximated the distance attenuation from the light cluster by the inverse square distance between the shading point and the center of the AABB. Although their approximation satisfies Eq. 1, it produces an enormous error for shading points near the AABB center. Lin and Yuksel [2020] mitigated the distance attenuation error by using the minimum and maximum distances from the shading point to the AABB. However, they still overstated their importance by using the directional bound. To reduce the variance due to the importance error, these previous methods generated multiple samples per tree-traversal query using adaptive tree splitting [Conty and Kulla 2018] or lightcuts. For high-frequency isotropic microfacet BRDFs, Liu et al. [2019] represented the light cluster with a sphere light, and then approximated the illumination integral for the sphere light by using a spherical pivot transformed distribution (SPTD) [Dupuy et al. 2017]. Since the SPTD cannot represent an anisotropic reflection lobe created by the microfacet BRDF at grazing angles, their method can underestimate the importance even for isotropic BRDFs. Therefore, Liu et al. combined their technique with Conty and Kulla [2018]'s technique using MIS. This combination is robust, but it increases the computational cost.

To reduce the MC variance even with one light sample per query, we introduce a more accurate importance approximation than the previous methods, while considering anisotropic microfacet BRDFs.

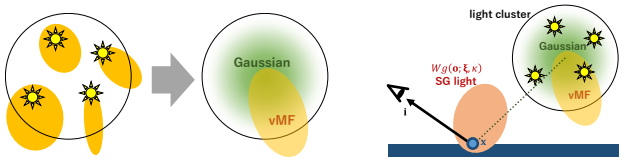
Table 1. Notations used in this paper

Symbol		Description
I	$\in [0, \infty)$	Importance of a light cluster
$\mathbf{x}$	$\in \mathbb{R}^3$	Shading point
$\mathbf{n}$	$\in S^2$	Surface normal
$\mathbf{i}$	$\in S^2$	View direction
$\mathbf{o}$	$\in S^2$	Light direction
$\mathbf{h}$	$\in S^2$	Halfvector between $\mathbf{i}$ and $\mathbf{o}$
$[i_x, i_y, i_z]$	$\in S^2$	Tangent-space view direction
$[h_x, h_y, h_z]$	$\in S^2$	Tangent-space halfvector
$L(\mathbf{x}, \mathbf{o})$	$\in [0, \infty)$	Incoming radiance from the light cluster
$f(\mathbf{i}, \mathbf{o})$	$\in [0, \infty)$	BRDF
$g(\mathbf{o}; \boldsymbol{\xi}, \kappa)$	$\in [0, \infty)$	Spherical Gaussian (SG), $g(\mathbf{o}; \boldsymbol{\xi}, \kappa) = e^{\kappa(\mathbf{o} \cdot \boldsymbol{\xi} - \kappa}$
$\boldsymbol{\mu}$	$\in \mathbb{R}^3$	Spatial mean of a light cluster
σ^2	$\in [0, \infty)$	Spatial variance of a light cluster
$\boldsymbol{\nu}$	$\in S^2$	vMF axis of a light cluster
λ	$\in [0, \infty)$	vMF sharpness of a light cluster
W	$\in [0, \infty)$	Amplitude of an SG light viewed from $\mathbf{x}$
$\boldsymbol{\xi}$	$\in S^2$	Axis of an SG light viewed from $\mathbf{x}$
κ	$\in [0, \infty)$	Sharpness of an SG light viewed from $\mathbf{x}$
$D(\mathbf{h}; \mathbf{A})$	$\in [0, \infty)$	Microfacet normal distribution function (NDF)
$[\alpha_x, \alpha_y]$	$\in (0, 1]^2$	Axis-aligned roughness parameter for the NDF
$\mathbf{A}$		2×2 roughness matrix, $\mathbf{A} = \begin{bmatrix} \alpha_x^2 & 0 \\ 0 & \alpha_y^2 \end{bmatrix}$
$\bar{\mathbf{A}}$		2×2 filtered roughness matrix
$H(\mathbf{o} \cdot \mathbf{n})$	$\in \{0, 1\}$	Heaviside function: 1 if $\mathbf{o} \cdot \mathbf{n} > 0$ and 0 otherwise

3 Spherical Gaussian (SG) Light Tree

3.1 SG Light Clusters

In this paper, we propose an SG light tree to improve the importance approximation for each node. To represent the distribution of a light cluster in each node, we follow virtual SG lights [Tokuyoshi 2015], which are more compact than bounding boxes and cones [Conty and Kulla 2018]. It approximates the distribution of light positions by an isotropic Gaussian (i.e., mean $\boldsymbol{\mu}$ and variance σ^2) and the directional distribution of radiant intensity by a vMF (i.e., normalized SG with axis $\boldsymbol{\nu}$ and sharpness λ) for each cluster (Fig. 2). In our method, this approximation is done during the construction of the light tree. We first build a binary tree of light primitives in an existing manner (e.g., surface area orientation heuristic, SAOH [Conty and Kulla 2018]).



(a) Our light cluster representation (b) SG light given by the cluster

Fig. 2. Our light cluster representation in each node. The spatial and directional distributions in a light cluster are represented by a Gaussian and a vMF, respectively. The product of the two distributions viewed from a shading point gives incoming radiance from the light cluster as an SG light.

Then, similar to mipmapping of virtual SG lights, we calculate the light distribution parameters as well as the bounding sphere of the cluster in each node in a bottom-up fashion.

3.1.1 vMF Distribution for Radiant Intensity. We use Banerjee et al.'s method [2005] to approximate the directional distribution of radiant intensity with a vMF. For this approximation in each node, we compute the weighted average of the radiation directions of lights as follows:

$$\bar{\mathbf{v}} = w_{\text{left}}\bar{\mathbf{v}}_{\text{left}} + w_{\text{right}}\bar{\mathbf{v}}_{\text{right}}, \tag{2}$$

where $w_{\text{left}} = \Phi_{\text{left}}/(\Phi_{\text{left}}+\Phi_{\text{right}})$ and $w_{\text{right}} = \Phi_{\text{right}}/(\Phi_{\text{left}}+\Phi_{\text{right}})$ are weights, $\Phi_{\text{left}} \in [0,\infty)$ and $\Phi_{\text{right}} \in [0,\infty)$ are radiant flux, and $\bar{\mathbf{v}}_{\text{left}} \in [-1,1]^3$ and $\bar{\mathbf{v}}_{\text{right}} \in [-1,1]^3$ are the average directions for left and right child nodes, respectively. Then, we convert the average direction $\bar{\mathbf{v}}$ to vMF axis $\mathbf{v}$ and sharpness λ as follows:

$$\mathbf{v} = \frac{\bar{\mathbf{v}}}{\|\bar{\mathbf{v}}\|}, \quad \lambda = \frac{3\|\bar{\mathbf{v}}\| - \|\bar{\mathbf{v}}\|^3}{1-\|\bar{\mathbf{v}}\|^2}. \tag{3}$$

For an area light source in each leaf node, we use $\bar{\mathbf{v}} \approx 0.5\mathbf{n}$ to roughly fit the vMF to Lambert's cosine.

3.1.2 Spatial Mean and Variance for Light Positions. In each node, we compute the spatial mean of light positions by the weighted average of light positions weighted by radiant flux as follows:

$$\boldsymbol{\mu} = w_{\text{left}}\boldsymbol{\mu}_{\text{left}} + w_{\text{right}}\boldsymbol{\mu}_{\text{right}}, \tag{4}$$

where $\boldsymbol{\mu}_{\text{left}} \in \mathbb{R}^3$ and $\boldsymbol{\mu}_{\text{right}} \in \mathbb{R}^3$ are the spatial mean of light positions for left and right child nodes, respectively. For a triangle light in each leaf node, the spatial mean within the triangle is simply given by $\bar{\boldsymbol{\mu}} = (\mathbf{p}_0+\mathbf{p}_1+\mathbf{p}_2)/3$ where $\mathbf{p}_0$, $\mathbf{p}_1$, and $\mathbf{p}_2$ are vertex positions of the triangle. The spatial variance is computed using the same weights as follows:

$$\sigma_s^2 = w_{\text{left}}\sigma_{s,\text{left}}^2 + w_{\text{right}}\sigma_{s,\text{right}}^2 + w_{\text{left}}w_{\text{right}}\|\boldsymbol{\mu}_{\text{left}} - \boldsymbol{\mu}_{\text{right}}\|^2, \tag{5}$$

where $\sigma_{s,\text{left}}^2 \in [0,\infty)$ and $\sigma_{s,\text{right}}^2 \in [0,\infty)$ are the spatial variance of light positions for left and right child nodes, respectively. For a triangle light in each leaf node, the spatial variance is given by $\sigma_s^2 = \left(\|\mathbf{e}_1\|^2 + \|\mathbf{e}_2\|^2 - (\mathbf{e}_1\cdot\mathbf{e}_2)\right)/18$, where $\mathbf{e}_1 = \mathbf{p}_1 - \mathbf{p}_0$ and $\mathbf{e}_2 = \mathbf{p}_2 - \mathbf{p}_0$.

Although we can use $\sigma^2 = \sigma_s^2$, we also provide another option: $\sigma^2 = 0.5r^2$ where $r \in [0,\infty)$ is the bounding sphere radius. This bound-based variance is of lower quality than σ_s^2 in most cases, but conservative for outliers. In our experiments, we use a hybrid approach: $\sigma^2 = \sigma_s^2(1-c)+0.5r^2c$ where $c = \max(\mathbf{n}\cdot(\mathbf{x}-\boldsymbol{\mu})/\|\mathbf{x}-\boldsymbol{\mu}\|, 0)$. For details, please refer to the supplementary document.

3.2 Importance Based on SG Lighting

During the tree traversal for light sampling, we approximate the incoming radiance $L(\mathbf{x},\mathbf{o})$ from the light cluster by the product of the two distributions in the light direction space. This product yields an SG as follows:

$$L(\mathbf{x},\mathbf{o}) \approx \frac{\Phi g\left(\mathbf{o};-\mathbf{v},\lambda\right) g\left(\mathbf{o};\frac{\boldsymbol{\mu}-\mathbf{x}}{\|\boldsymbol{\mu}-\mathbf{x}\|},\frac{\|\boldsymbol{\mu}-\mathbf{x}\|^2}{\sigma^2}\right)}{2\pi\sigma^2\int_{S^2} g\left(\boldsymbol{\omega};-\mathbf{v},\lambda\right)\mathrm{d}\boldsymbol{\omega}} = W g\left(\mathbf{o};\boldsymbol{\xi},\kappa\right),$$

where $\Phi \in [0,\infty)$ is the radiant flux of the light cluster. The SG light parameters W, $\boldsymbol{\xi}$, and κ are obtained analytically, and a numerically robust implementation is available as open source [Tokuyoshi 2024] (please refer to the supplementary document for details). With this SG light, our importance for each node is given by

$$I \propto\!\!\!\propto \int_{S^2} L(\mathbf{x},\mathbf{o}) f(\mathbf{i},\mathbf{o})|\mathbf{o}\cdot\mathbf{n}|\mathrm{d}\mathbf{o} \approx W\int_{S^2} g\left(\mathbf{o};\boldsymbol{\xi},\kappa\right) f(\mathbf{i},\mathbf{o})|\mathbf{o}\cdot\mathbf{n}|\mathrm{d}\mathbf{o}. \tag{6}$$

Compared to the previous bound-based importance, this SG-based importance avoids the singularity at the cluster center that causes the enormous distance attenuation error, and more accurately takes into account the light distribution within the cluster. To compute this importance analytically while satisfying the constraint of unbiased sampling (Eq. 1), we introduce novel SG lighting approximations.

4 Interpolation-based Diffuse SG Lighting

In this paper, we approximate the diffuse BRDF by the Lambert model $f(\mathbf{i},\mathbf{o}) \propto\!\!\!\propto H(\mathbf{o}\cdot\mathbf{n})/\pi$, and represent the SG lighting by the product integral of an SG and the clamped cosine term. For this product integral, we introduce a simpler and more accurate approximation than previous methods, while satisfying Eq. 1.

Our approximation is derived in a similar manner to the hemispherical integral of an SG [Meder and Brüderlin 2018; Tokuyoshi 2022] that is approximated by an interpolation between the upper hemispherical integral and the lower hemispherical integral. For diffuse SG lighting, we interpolate the cosine-weighted hemispherical integrals. If the SG axis is $\boldsymbol{\xi} = \pm\mathbf{n}$, the product integral of the SG and the clamped cosine has the following closed-form solutions:

$$\hat{B}(\kappa) = \int_{S^2} g(\mathbf{o};\mathbf{n},\kappa)\max(\mathbf{o}\cdot\mathbf{n},0)\mathrm{d}\mathbf{o} = \frac{2\pi\left(\mathrm{e}^{-\kappa}-1+\kappa\right)}{\kappa^2},$$

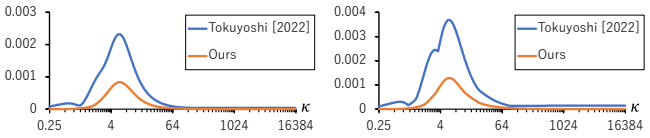
$$\check{B}(\kappa) = \int_{S^2} g(\mathbf{o};-\mathbf{n},\kappa)\max(\mathbf{o}\cdot\mathbf{n},0)\mathrm{d}\mathbf{o} = \frac{2\pi\mathrm{e}^{-\kappa}\left(1-\mathrm{e}^{-\kappa}-\kappa\mathrm{e}^{-\kappa}\right)}{\kappa^2}.$$

The product integral of an SG with an arbitrary axis and the clamped cosine monotonically increases between $\check{B}(\kappa)$ and $\hat{B}(\kappa)$. Therefore, we represent the product integral by interpolating them:

$$\boxed{I \propto\!\!\!\propto \frac{W}{\pi}\int_{S^2} g(\mathbf{o};\boldsymbol{\xi},\kappa)\max(\mathbf{o}\cdot\mathbf{n},0)\mathrm{d}\mathbf{o} = \frac{W}{\pi}\left(\hat{B}(\kappa)u + \check{B}(\kappa)(1-u)\right)}, \tag{7}$$

where the interpolation factor $u \in [0,1]$ is the normalized product integral. In this paper, we approximate this u as follows:

$$u = \frac{\int_{S^2} g(\mathbf{o};\boldsymbol{\xi},\kappa)\max(\mathbf{o}\cdot\mathbf{n},0)\mathrm{d}\mathbf{o} - \check{B}(\kappa)}{\hat{B}(\kappa)-\check{B}(\kappa)} \approx \frac{q(\boldsymbol{\xi}\cdot\mathbf{n},\kappa)-q(-1,\kappa)}{q(1,\kappa)-q(-1,\kappa)},$$



(a) Root mean square error (RMSE) (b) Maximum absolute error

Fig. 3. Plots of RMSEs (a) and maximum errors (b) for the product integral of a normalized SG and the clamped cosine. Errors at different SG axes are aggregated for each SG sharpness κ. Our method produces smaller errors than the state-of-the-art approximation [Tokuyoshi 2022].

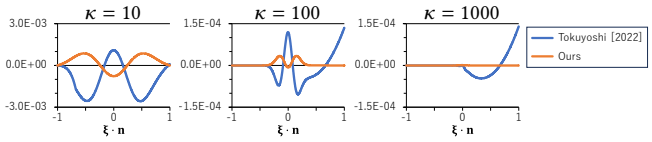


Fig. 4. Plots of signed errors for the product integral of a normalized SG and the clamped cosine. Unlike the previous method (blue line), our approximation (orange line) reduces the error around $\boldsymbol{\xi} \cdot \mathbf{n} = 1$ for large κ.

where $q(\boldsymbol{\xi} \cdot \mathbf{n}, \kappa) \approx \int_{S^2} g(\mathbf{o}; \boldsymbol{\xi}, \kappa) \max(\mathbf{o} \cdot \mathbf{n}, 0) \mathrm{d}\mathbf{o}$ is monotonically increasing for $\boldsymbol{\xi} \cdot \mathbf{n}$. Since $\hat{B}(\kappa) > 0$ and $\check{B}(\kappa) > 0$, the interpolation form (Eq. 7) satisfies Eq. 1 even approximating u. In addition, the approximation error for u approaches zero for $\boldsymbol{\xi} \cdot \mathbf{n} \to \pm 1$. Therefore, we introduce an approximate product integral $q(\boldsymbol{\xi} \cdot \mathbf{n}, \kappa)$ considering the accuracy around $\boldsymbol{\xi} \cdot \mathbf{n} = 0$. For $\kappa \to \infty$, the SG asymptotically approaches a Gaussian on a plane, therefore we approximate the product integral using an analytical convolution of the planar Gaussian and the clamped cosine. To reduce the approximation error around $\boldsymbol{\xi} \cdot \mathbf{n} = 0$, we use a plane parallel to the surface normal $\mathbf{n}$ as follows:

$$q(\boldsymbol{\xi} \cdot \mathbf{n}, \kappa) = \iint_{\mathbb{R}^2} \mathrm{e}^{-t(\kappa)^2 \left(x^2 + (y - (\boldsymbol{\xi} \cdot \mathbf{n}))^2\right)} \max(y, 0) \mathrm{d}x \mathrm{d}y$$
$$\propto t(\kappa)(\boldsymbol{\xi} \cdot \mathbf{n}) \mathrm{erfc}\left(-t(\kappa)(\boldsymbol{\xi} \cdot \mathbf{n})\right) + \frac{\mathrm{e}^{-t(\kappa)^2 (\boldsymbol{\xi} \cdot \mathbf{n})^2}}{\sqrt{\pi}},$$

where $t(\kappa) \in [0, \infty)$ is the inverse standard deviation $\times \sqrt{0.5}$ for the planar Gaussian. For this $t(\kappa)$, we use a fitted approximation: $t(\kappa) \approx \kappa \sqrt{\frac{0.5\kappa^2 + 2.7360833\kappa + 17.021297}{\kappa^3 + 4.0100827\kappa^2 + 15.219156\kappa + 76.087896}}$. In the supplemental material, we provide a numerically stable implementation for our SG lighting as well as previous SG lighting methods.

For the product integral of an SG and the clamped cosine, our approximation computes only one interpolation, whereas the previous approximations [Meder and Brüderlin 2018; Tokuyoshi 2022] computed an SG product and two interpolations. Thus, the implementation of our method is simpler than the previous methods. On an AMD Radeon™ RX 7900 XTX GPU, our method is 1.4 times faster than the state-of-the-art method [Tokuyoshi 2022] thanks to the simplicity. In addition, it also significantly improves the accuracy, as shown in Figs. 3 and 4.

5 NDF Filtering for Glossy SG Lighting

In this section, we introduce an NDF filtering method for SG lighting. While the original NDF filtering projected a pixel footprint into halfvector space for specular antialiasing, our method projects an SG light into halfvector space to analytically compute the lighting integral. Fig. 5 illustrates the previous SG lighting [Xu et al. 2013] and our SG lighting for anisotropic microfacet BRDFs. The previous SG lighting approximated the reflection lobe with an anisotropic SG (ASG) and violated Eq. 1. On the other hand, we preserve the original NDF model while approximately convolving the NDF with the SG light in halfvector space. Our NDF filtering increases the roughness parameter by convolving the NDF with an SG light. Therefore, it reduces the approximation error for NDF models with longer tails than the SG (e.g., the GGX NDF). For such NDFs, the previous ASG approximation can underestimate the tail to zero or nearly zero as shown in Fig. 6a, and thus it can cause a significant MC error for importance sampling. In addition, the ASG approximation distorts anisotropic reflection lobes inappropriately. Since our NDF filtering does not introduce such lobe distortion and underestimation for tails (Fig. 6b), it is suitable for approximating the importance.

5.1 NDF Filtering

Our filtering method is built upon stable NDF filtering [Tokuyoshi and Kaplanyan 2021] on an orthographically projected halfvector space $[h_x, h_y]$. For this filtering, we first represent the NDF with a bivariate Gaussian on the $[h_x, h_y]$ space: $D(\mathbf{h}; \mathbf{A}) \propto \mathrm{e}^{-\frac{1}{2}[h_x, h_y] \Sigma_D^{-1} [h_x, h_y]^\top}$ where Σ_D is the 2×2 covariance matrix. For slope-space elliptical NDFs such as the GGX and Beckmann NDFs, this covariance matrix was derived by Tokuyoshi and Kaplanyan [2021] as follows:

$$\Sigma_D \approx \frac{1}{2}\left(\mathbf{A}^{-1} - \mathbf{E}\right)^{-1}, \tag{8}$$

where $\mathbf{E}$ is the 2×2 identity matrix. Next, we project the SG light to halfvector space $[h_x, h_y]$. The variance of the SG light is given by the reciprocal sharpness $1/\kappa$. We project this variance to halfvector space using the following transformation:

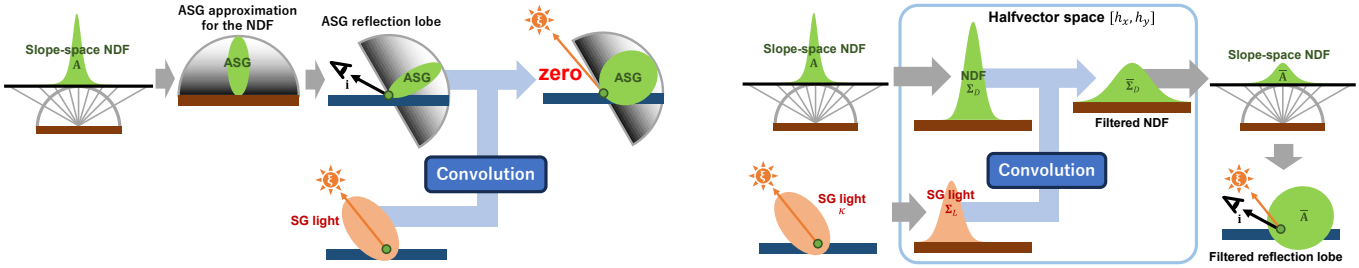
$$\Sigma_L = \mathbf{J} \begin{bmatrix} 1/\kappa & 0 \\ 0 & 1/\kappa \end{bmatrix} \mathbf{J}^\top = \frac{1}{\kappa} \mathbf{J}\mathbf{J}^\top, \tag{9}$$

where $\mathbf{J}$ is the 2×2 Jacobian matrix for the transformation between halfvectors and light directions. We derived this Jacobian matrix for arbitrary reflection and refraction vectors in the supplementary document. Then, we approximately filter the NDF by Gaussian convolution, which simply sums the covariance matrices:

$$\bar{\Sigma}_D \approx \Sigma_D + \Sigma_L = \Sigma_D + \frac{1}{\kappa} \mathbf{J}\mathbf{J}^\top.$$

Although we can use the Jacobian matrix $\mathbf{J}$ at the SG light axis $\mathbf{o} = \boldsymbol{\xi}$, we use $\mathbf{J}$ at the NDF peak $\mathbf{h} = \mathbf{n}$ assuming highly specular surfaces for computational simplicity. In this case, Σ_D and $\mathbf{J}\mathbf{J}^\top$ are independent of lights and are constant during the light tree traversal. After filtering the NDF, we convert the covariance matrix to roughness using the inverse transformation of Eq. 8:

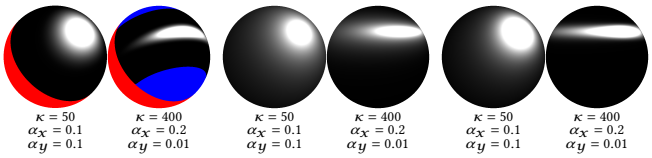
$$\bar{\mathbf{A}} \approx \left((2\bar{\Sigma}_D)^{-1} + \mathbf{E}\right)^{-1}.$$



(a) Previous SG lighting with an ASG approximation [Xu et al. 2013]

(b) Our SG lighting with NDF filtering

Fig. 5. SG lighting for anisotropic microfacet BRDFs. (a) The previous SG lighting first approximates the NDF with an ASG, and projects it to the reflection vector space to approximate the anisotropic reflection lobe with an ASG. Then, it convolves the ASG reflection lobe with an SG light. Instead of approximating the reflection lobe with an ASG, which violates Eq. 1, (b) our method projects the SG light to halfvector space and convolves the NDF with the projected light.



(a) ASG convolution (b) Our NDF filtering (c) Reference

Fig. 6. Visualization of convolved GGX reflection lobes projected onto the tangent plane. (a) ASG convolution [Xu et al. 2013] distorts the anisotropic reflection lobe inappropriately. In addition, the ASG approximation yields zero in the lower hemisphere of the lobe (red area), and thus violates Eq. 1. It also produces zero in the blue area due to a floating point error. (b) Our NDF filtering does not produce such undesirable zero and is closer to the reference (c) than the ASG convolution for anisotropic NDFs.

With this NDF filtering, we approximate the following convolution:

$$\int_{S^2} D\left(\mathbf{h};\mathbf{A}\right) g\left(\mathbf{o};\xi,\kappa\right) \mathrm{d}\mathbf{o} \propto D\left(\frac{\mathbf{i}+\xi}{\|\mathbf{i}+\xi\|};\bar{\mathbf{A}}\right). \tag{10}$$

We use this convolution for the SG lighting integral. For NDFs generalized by the roughness matrix [Heitz 2014], please refer to the supplementary document. The approximation error of Eq. 10 approaches zero for $\kappa \to \infty$. Therefore, our method reduces the error for sharp SG lights unlike the ASG approximation.

5.2 Lighting Approximation with the Filtered NDF

Although we use NDF filtering instead of the ASG approximation that violates Eq. 1, the filtered BRDF lobe can still violate the constraint because the microfacet model has several Heaviside functions. Therefore, we further approximate the reflection lobe using a visible normal distribution function (VNDF) [Heitz 2014] without Heaviside functions (i.e., a VNDF for symmetric microflakes [Heitz et al. 2015]). This VNDF-based reflection lobe $p\left(\mathbf{o};\mathbf{i},\mathbf{A}\right)$ is given by

$$p\left(\mathbf{o};\mathbf{i},\mathbf{A}\right) = \frac{D(\mathbf{m};\mathbf{A})|\mathbf{i}\cdot\mathbf{m}|}{\int_{S^2} D(\boldsymbol{\omega};\mathbf{A})|\mathbf{i}\cdot\boldsymbol{\omega}|\mathrm{d}\boldsymbol{\omega}} \left\|\frac{\partial\mathbf{m}}{\partial\mathbf{o}}\right\|,$$

where the halfvector is flipped by $\mathbf{m} = \mathrm{sgn}(\mathbf{h}\cdot\mathbf{n})\mathbf{h}$ for symmetry, $\|\partial\mathbf{m}/\partial\mathbf{o}\| = 1/(4|\mathbf{i}\cdot\mathbf{m}|)$ is the Jacobian for the transformation between halfvectors and reflection vectors, and the normalization factor is $\int_{S^2} D(\boldsymbol{\omega};\mathbf{A})|\mathbf{i}\cdot\boldsymbol{\omega}|\mathrm{d}\boldsymbol{\omega} = \sqrt{[i_x,i_y]\mathbf{A}[i_x,i_y]^\top + i_z^2}$ for the GGX NDF (for the Beckmann NDF, please refer to the supplementary document). Using this VNDF, we approximate the reflection lobe by

$$f(\mathbf{i},\mathbf{o})|\mathbf{o}\cdot\mathbf{n}| \propto H(\mathbf{o}\cdot\mathbf{n})p\left(\mathbf{o};\mathbf{i},\mathbf{A}\right), \tag{11}$$

where $H(\mathbf{o}\cdot\mathbf{n})$ is the visibility of the reflection lobe in the upper hemisphere. In this paper, we ignore Heaviside functions other than this visibility. By substituting Eq. 11 into Eq. 6, we get

$$I \propto WV \frac{\int_{S^2} p\left(\mathbf{o};\mathbf{i},\mathbf{A}\right) g\left(\mathbf{o};\xi,\kappa\right) \mathrm{d}\mathbf{o}}{\int_{S^2} g\left(\mathbf{o};\xi,\kappa\right) \mathrm{d}\mathbf{o}} \int_{S^2} g\left(\mathbf{o};\xi,\kappa\right) \mathrm{d}\mathbf{o},$$

where $V = \frac{\int_{S^2} H(\mathbf{o}\cdot\mathbf{n})p(\mathbf{o};\mathbf{i},\mathbf{A})g(\mathbf{o};\xi,\kappa)\mathrm{d}\mathbf{o}}{\int_{S^2} p(\mathbf{o};\mathbf{i},\mathbf{A})g(\mathbf{o};\xi,\kappa)\mathrm{d}\mathbf{o}}$ is the filtered visibility. By using NDF filtering (Eq. 10) for $p(\mathbf{o};\mathbf{i},\mathbf{A}) \propto D(\mathbf{m};\mathbf{A})$, we obtain

$$\frac{\int_{S^2} p(\mathbf{o};\mathbf{i},\mathbf{A}) g\left(\mathbf{o};\xi,\kappa\right) \mathrm{d}\mathbf{o}}{\int_{S^2} g\left(\mathbf{o};\xi,\kappa\right) \mathrm{d}\mathbf{o}} \approx p\left(\xi;\mathbf{i},\bar{\mathbf{A}}\right).$$

This yields the following SG lighting approximation:

$$\boxed{I \propto WVp\left(\xi;\mathbf{i},\bar{\mathbf{A}}\right) \int_{S^2} g\left(\mathbf{o};\xi,\kappa\right) \mathrm{d}\mathbf{o}}. \tag{12}$$

By using Eq. 12 for the importance of each light tree node, we improve the light sampling quality for microfacet BRDFs.

Filtered Visibility. The filtered visibility $V \in [0,1]$ could be computed using SSDFs [Wang et al. 2009] by approximating the filter kernel $p(\mathbf{o};\mathbf{i},\mathbf{A})g\left(\mathbf{o};\xi,\kappa\right)$ with an SG as follows:

$$V = \frac{\int_{S^2} H(\mathbf{o}\cdot\mathbf{n})p(\mathbf{o};\mathbf{i},\mathbf{A})g\left(\mathbf{o};\xi,\kappa\right) \mathrm{d}\mathbf{o}}{\int_{S^2} p(\mathbf{o};\mathbf{i},\mathbf{A})g\left(\mathbf{o};\xi,\kappa\right) \mathrm{d}\mathbf{o}} \approx \frac{\int_{S^2} H(\mathbf{o}\cdot\mathbf{n})g(\mathbf{o};\acute{\xi},\acute{\kappa})\mathrm{d}\mathbf{o}}{\int_{S^2} g(\mathbf{o};\acute{\xi},\acute{\kappa})\mathrm{d}\mathbf{o}}.$$

To compute the rightmost without violating Eq. 1, we use a more accurate hemispherical integral [Tokuyoshi 2022] than SSDFs. In addition, unlike Wang et al. [2009], we filter the visibility using a lower-frequency SG kernel than the original anisotropic kernel to conservatively reduce the MC variance. To obtain this lower-frequency kernel, we approximate the reflection lobe with an SG as follows: $p(\mathbf{o};\mathbf{i},\mathbf{A})g(\mathbf{o};\xi,\kappa) \propto g(\mathbf{o};\xi_p,\kappa_p)g(\mathbf{o};\xi,\kappa) \propto g(\mathbf{o};\acute{\xi},\acute{\kappa})$, where the lobe variance $1/\kappa_p$ is equal to or greater than the maximum eigenvalue of the reflection lobe covariance matrix $\mathbf{J}^{-1}\Sigma_D(\mathbf{J}^{-1})^\top$. We get this lobe variance by substituting the maximum roughness $\max(\alpha_x,\alpha_y)$ into $\mathbf{J}^{-1}\Sigma_D(\mathbf{J}^{-1})^\top$. At the perfect specular reflection

Table 2. Computation time for cluster parameter calculation on the CPU.

	Hangarship	Museum	Toyshop	Bistro
Conty and Kulla [2018]	0.73 ms	2.14 ms	20.1 ms	2.49 ms
Our SG light tree	0.49 ms	1.59 ms	13.5 ms	1.75 ms

vector $\boldsymbol{\xi}_p = 2(\mathbf{i}\cdot\mathbf{n})\mathbf{n}-\mathbf{i}$, we obtain $\kappa_p = (1-\max(\alpha_x^2, \alpha_y^2))/(2\max(\alpha_x^2, \alpha_y^2))$. For implementation details, please refer to the supplementary code.

6 Experimental Results

Here we show the rendering results of hierarchical light sampling using different importance approximations with one light sample per tree-traversal query. We compare our method with Conty and Kulla [2018]'s importance (CK2018), Conty and Kulla's importance with Lin and Yuksel [2020]'s distance attenuation heuristic (CK2018+LY2020), and Liu et al. [2019]'s MIS (LXY2019). All materials are Autodesk Standard Surface [Georgiev et al. 2019], and images are rendered at 3840×2160 pixels using HIPRT [Meister et al. 2024] performed on an AMD Radeon™ RX 7900 XTX GPU. For a fair comparison, we build light trees using a high-quality sweep SAOH [Conty and Kulla 2018] in preprocessing to maximize the efficiency of the previous methods. For the SAOH construction time which is common to all the methods, please see the supplementary document. The image quality is evaluated with the root mean square percentage error (RMSPE) and mean absolute percentage error (MAPE) metrics.

Implementation. In this experiment, we use a weighted average of our diffuse importance (Eq. 7) and glossy importance (Eq. 12) weighted by the reflectance of each BRDF lobe. We use the diffuse importance for diffuse and sheen layers in Standard Surface. For two glossy layers in Standard Surface, we merge the two lobes into a single lobe by weighted averaging the roughness matrix $\mathbf{A}$ of each layer to reduce computational cost. Thus, this experiment uses a simple diffuse-glossy importance model for multi-lobe BRDFs. For our SG light tree, we store $\boldsymbol{\nu}$, λ, $\boldsymbol{\mu}$, σ_s^2, r, and Φ in each node without data compression (i.e., 10 floating point values). This cluster representation is more compact than the previous cluster representation [Conty and Kulla 2018] (12 floating point values). Although we may be able to further reduce the data size by quantizing the cluster parameters, this optimization is left for future work. We calculate the cluster parameters on an AMD Ryzen™ 9 7950X CPU without parallelization. Even with this naïve implementation, the computation time (Table 2) is negligibly small and faster than the previous method thanks to the simplicity of Eqs. 2, 3, 4, and 5.

Direct Illumination. Fig. 7 shows the equal-time comparison for direct illumination using hierarchical light sampling. While our method has a smaller number of path samples per pixel (spp) than CK2018+LY2020 due to computational overhead, it produces lower error than the previous methods, especially for glossy reflections at grazing angles. Fig. 8 shows plots of error convergence for each scene. Although our method reduces the error, there are still large oscillations due to firefly noise for RMSPE. For error convergence, please see MAPEs, which are less sensitive to such outliers.

Table 3. Computation time for direct illumination (3840×2160 pixels, 1 spp).

	CK2018	CK2018+LY2020	LXY2019	Ours
Hangarship	16.2 ms	16.7 ms	47.6 ms	23.1 ms
Museum	19.7 ms	20.5 ms	59.5 ms	29.7 ms
Toyshop	21.5 ms	22.0 ms	65.6 ms	29.9 ms
Bistro	22.6 ms	23.4 ms	62.9 ms	34.2 ms

Table 4. Computation time for path tracing (3840×2160 pixels, 1 spp).

	CK2018	CK2018+LY2020	LXY2019	Ours
Hangarship	99 ms	101 ms	219 ms	120 ms
Museum	124 ms	131 ms	306 ms	160 ms
Toyshop	123 ms	126 ms	309 ms	149 ms
Bistro	124 ms	126 ms	250 ms	159 ms

Path Tracing. In path tracing, noise on glossy surfaces can be reduced by combining with BRDF sampling using MIS. Fig. 10 shows the equal-time comparison for this practical scenario. Even using MIS, our method produces less error than the previous methods for glossy surfaces. For the Bistro scene where diffuse surfaces are dominant, our method is comparable to CK2018+LY2020 in RMSPE, while our method slightly outperforms it in MAPE. Fig. 11 shows plots of error convergence for each scene. Unlike direct illumination without BRDF sampling, there are fewer oscillations in RMSPE. For glossy scenes, our method improves the convergence speed in both RMSPE and MAPE, even though our light cluster representation is more compact than the previous methods. In the supplementary document, we show additional experimental results for other scenes.

Anisotropy. Fig. 9 shows path tracing with MIS for a highly anisotropic microfacet BRDF. In this scene, CK2018 and CK2018+LY2020 produce noticeable noise because these methods ignore BRDFs. LXY2019 takes glossy BRDFs into account, but it is still inefficient for such anisotropic materials due to the isotropic approximation. Unlike these previous methods, our importance approximation accurately represents anisotropic scattering effects thanks to NDF filtering. As a result, our method efficiently reduces noise on glossy highlights, especially for highly anisotropic microfacet BRDFs.

Performance. Tables 3 and 4 show the rendering times per path sample for direct illumination and path tracing, respectively. Our method has an overhead to calculate SG lighting using mathematical functions (e.g., exp function) for each BRDF lobe. Therefore, it is more expensive than ignoring BRDFs (CK2018+LY2020). On the other hand, our method is faster than Liu et al.'s MIS (LXY2019) which performs two different tree traversals. Our method reduces the MC variance by using only one tree traversal. Thus, our method improves the cost-effectiveness and convergence speed. For the computation times of sampling routines only, please refer to the supplementary document.

7 Limitations

Light Cluster Representation. Although our light cluster can represent all-frequency incoming radiance, it is an isotropic single

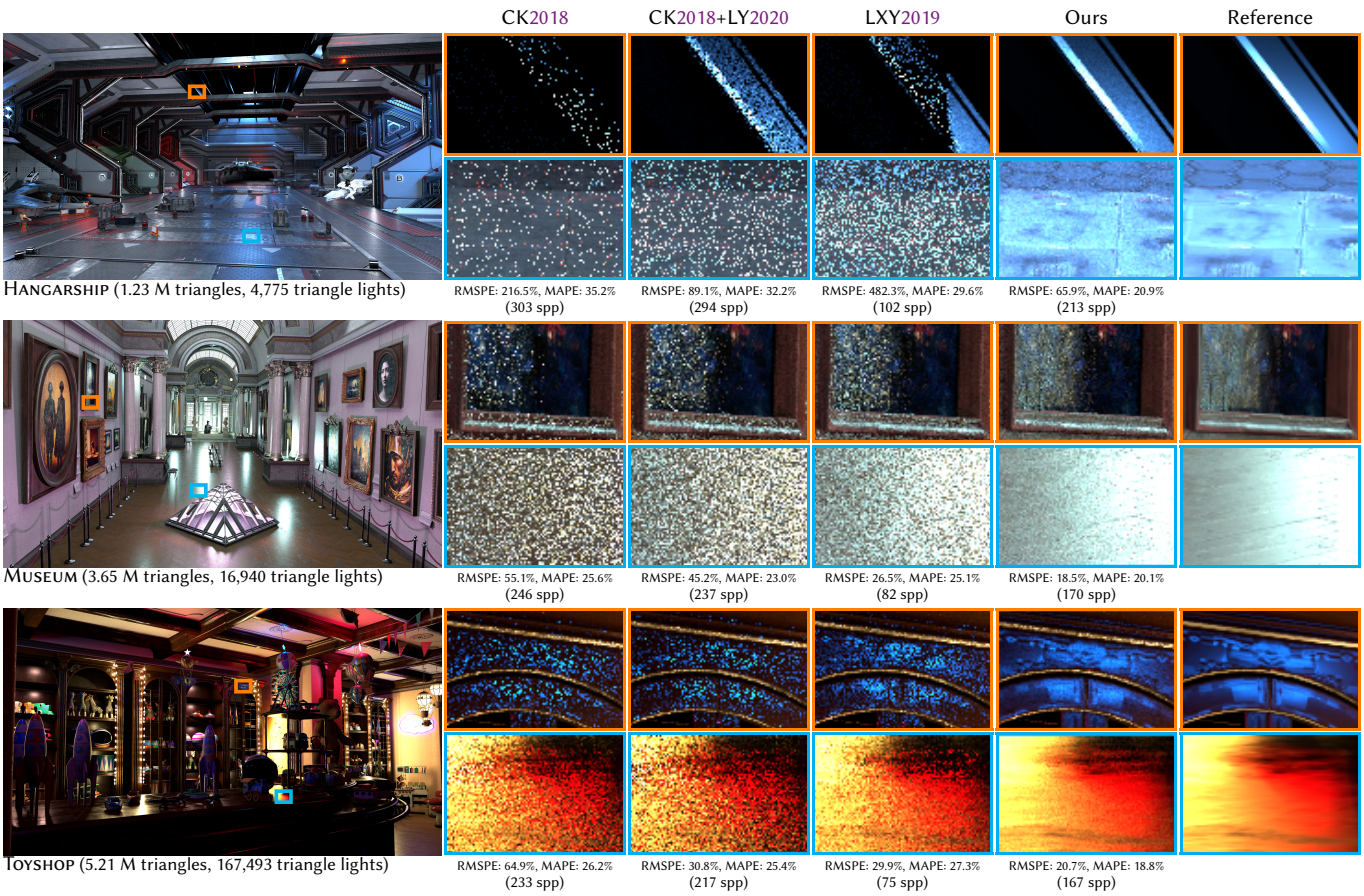


Fig. 7. Equal-time (5 s) comparison of our method and previous methods for direct illumination. BRDF sampling is not used in this experiment. The previous methods produce noticeable Monte Carlo noise on glossy surfaces, especially for highlights at grazing angles. Our method significantly reduces this noise.

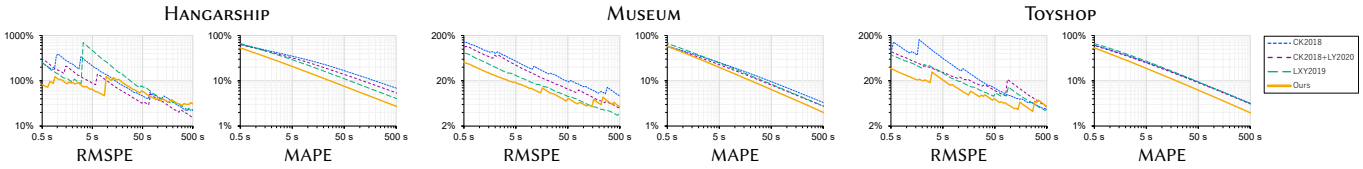


Fig. 8. Plots of RMSPE and MAPE for Fig. 7 (i.e., direct illumination without BRDF sampling). The oscillations in RMSPEs are due to firefly noise on glossy surfaces. The error convergence is visible in the MAPE metric, which is less sensitive to fireflies. Our method improves the convergence speed in MAPE.

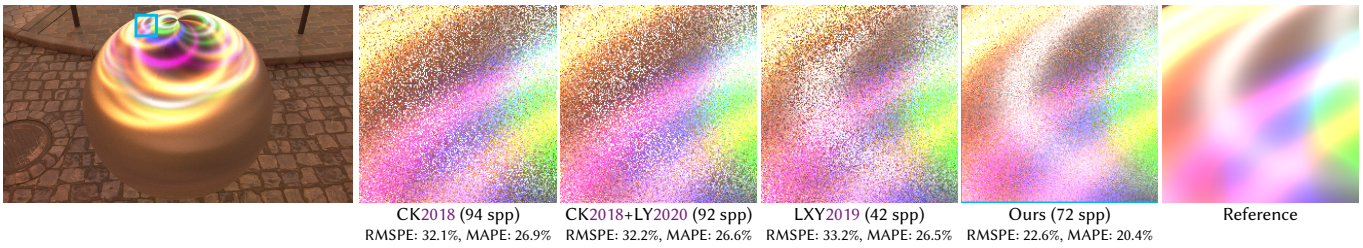


Fig. 9. Equal-time (10 s) comparison of path tracing for an anisotropic glossy sphere placed in the Bistro scene. Highlights are stretched and curved by the anisotropic microfacet BRDF. CK2018 and +LY2020 produce noticeable noise in the highlights because these methods ignore BRDFs. Even with LXY2019, it is still inefficient for this scene due to the isotropic approximation. On the other hand, our method significantly reduces the noise for these anisotropic highlights.

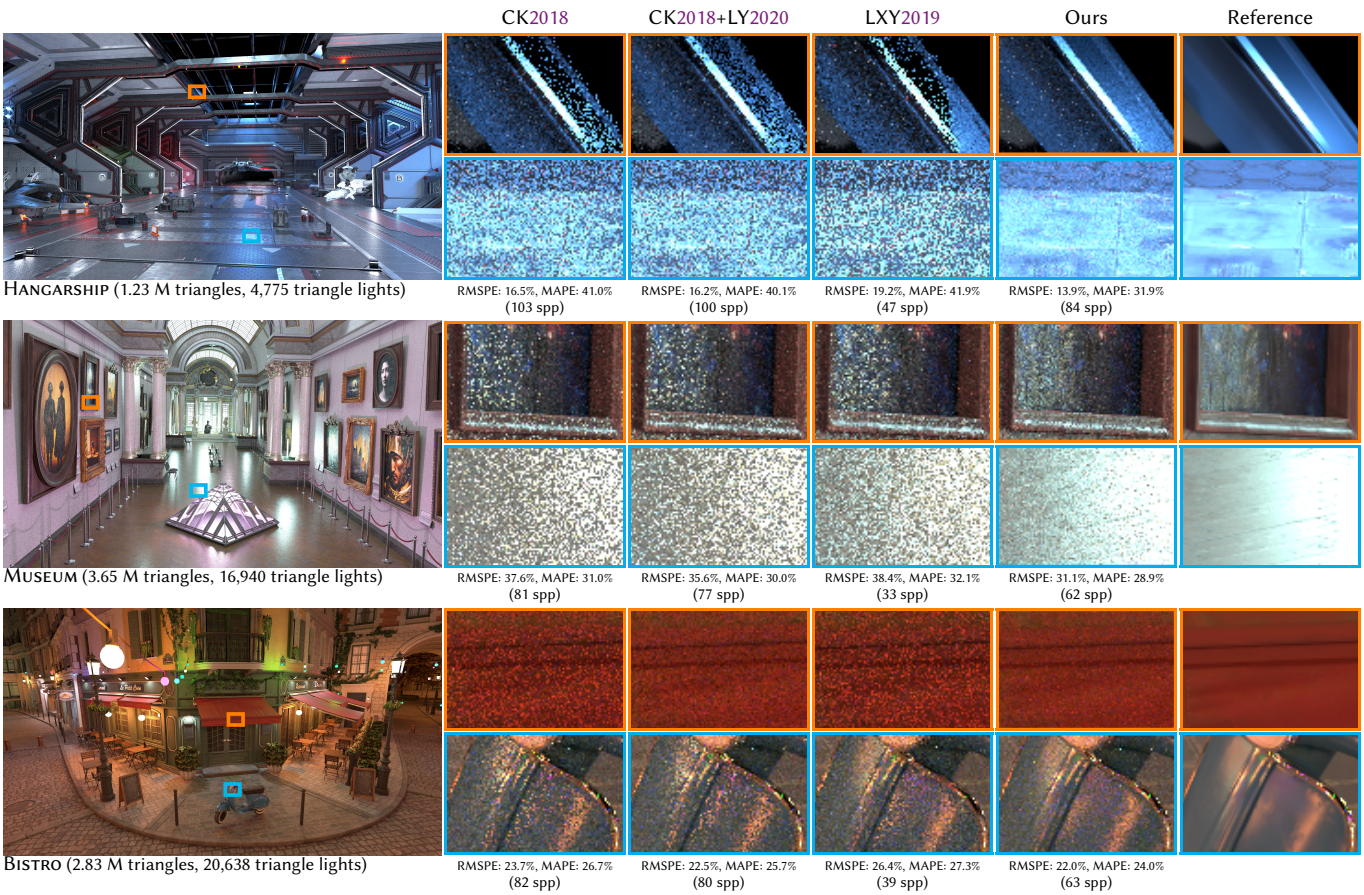


Fig. 10. Equal-time (10 s) comparison of our method and previous methods for path tracing with MIS. MIS improves the sampling quality on glossy surfaces, especially for CK2018 and CK2018+LY2020. Even with MIS, our method is more efficient than the previous methods for the Hangarship and Museum scenes. In the Bistro scene, our method produces a quality comparable to CK2018+LY2020 on diffuse surfaces, while reducing noise on moderately glossy surfaces, as shown in the closeups.

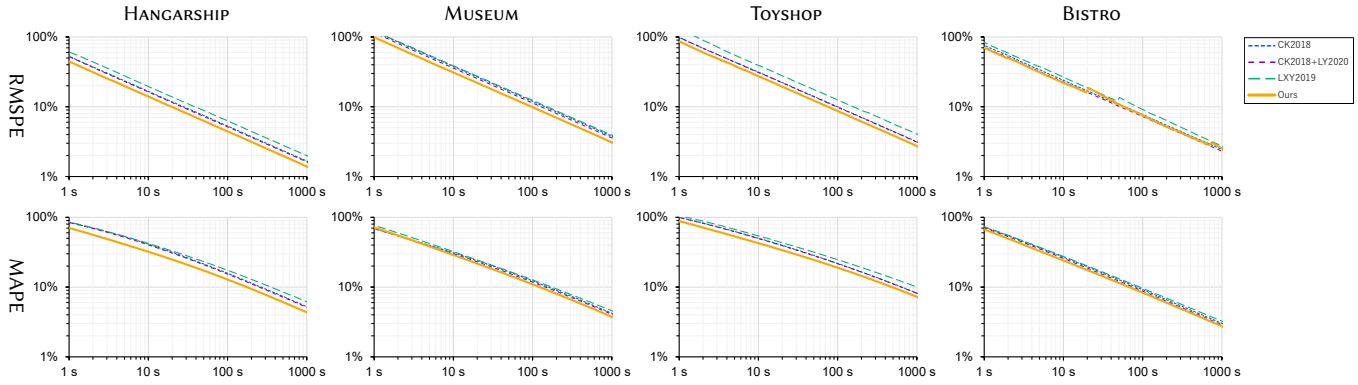


Fig. 11. Plots of RMSPE (upper row) and MAPE (lower row) for Figs. 1 and 10 (i.e., path tracing with MIS). The oscillations for the Bistro scene are due to firefly noise for caustics. Our method improves the error convergence speed for glossy scenes even using MIS with BRDF sampling. For the diffuse-dominant scene (Bistro), while our convergence speed is comparable to CK2018+LY2020 in RMSPE, it is slightly faster than the previous methods in MAPE.

lobe. It cannot represent complex multi-lobe incoming radiance and anisotropic distribution of lights (e.g., narrow light primitives).

Shadow Ray Visibility. While our method approximates the product of the incoming radiance and the BRDF for the importance, it ignores the shadow ray visibility as in previous work. Therefore, the method does not reduce the MC variance for shadows.

Computational Overhead. Our importance approximation has computational overhead to calculate SG lighting. In our current implementation, this SG lighting uses expensive mathematical functions and also increases register pressure. There is room for improvement to reduce the overhead by using simpler and faster implementations of mathematical functions.

Multi-lobe BRDFs. The computational cost of SG lighting is proportional to the number of BRDF lobes. To reduce the cost in our experiments, we merged two glossy lobes of Standard Surface into a single lobe. This approximation is lightweight, but the approximation error may not be negligible if the roughness parameters between the two glossy lobes are significantly different. In such cases, SG lighting should be calculated for every BRDF lobe.

8 Discussion and Future Work

Generalized Spherical Warping. Our covariance transformation (Eq. 9) is a generalized form of the previous *spherical warping* between an NDF and a reflection lobe [Wang et al. 2009; Xu et al. 2013]. Xu et al. [2013]'s Hessian matrix for the warped ASG convolution is equivalent to $(\mathbf{J}^{-1}\bar{\Sigma}_D(\mathbf{J}^{-1})^\top)^{-1} = (\mathbf{J}^{-1}\Sigma_D(\mathbf{J}^{-1})^\top + \frac{1}{\kappa}\mathbf{E})^{-1}$ using the Jacobian matrix $\mathbf{J}$ at the perfect specular reflection vector. Unlike the previous work, we derived $\mathbf{J}$ at an arbitrary light direction. Therefore, we can use a more accurate reflection lobe axis [Lagarde 2014] than the previous warping. Our method can also support transmission by using the Jacobian matrix $\mathbf{J}$ at a refraction vector. In the future, we would like to investigate the improvement of the reflection lobe axis and the efficiency of SG lighting for transmission.

High-frequency Radiant Intensity. Our SG light tree takes into account all-frequency radiant intensity. Therefore, it may be more suitable for lights with high-frequency directional distribution, such as spotlights and IES lights, than the previous bound-based light tree. Quality evaluation for such lights is also left for future work.

Point Lights. Although MIS with BRDF sampling has typically been used for glossy surfaces under area lights, BRDF sampling cannot sample point lights. Therefore, our method can be effective for such lights. We would like to evaluate the efficiency for glossy scenes with point (or extremely small) lights in the future.

Bidirectional Path Tracing (BDPT). Our light sampling method can also be used for path connections in light vertex cache BDPT [Davidovič et al. 2014]. Since our method takes glossy BRDFs into account, it can improve the quality of glossy-to-glossy and specular-diffuse-glossy paths, which are often problematic in BDPT. To build a tree for light vertices generated in each iteration of BDPT, we should use a fast tree construction method [Apetrei 2014] instead of the high-quality SAOH construction. In the future, we would like to extend our method for BDPT by using a fast tree construction.

9 Conclusions

We have presented a hierarchical product importance sampling method for many lights based on novel SG lighting approximations. Unlike previous high-quality SG lighting approximations, our SG lighting satisfies the unbiased sampling constraint while improving the approximation quality. For anisotropic microfacet BRDFs, we extended an NDF filtering method for SG lighting. Thanks to this NDF filtering, our light sampling method significantly reduces the MC variance for glossy highlights, especially at grazing angles. Although our method has an overhead to calculate SG lighting for each BRDF lobe, experimental results show that it is cost-effective for high-frequency or anisotropic BRDFs, even when combined with BRDF sampling using MIS. By using our importance, we can now render high-quality images without complex algorithms to generate multiple light samples per tree-traversal query. We have demonstrated the efficiency of using one light sample per query in the experimental results.

While we have presented accurate SG lighting for diffuse BRDFs and anisotropic microfacet BRDFs, we roughly represent a light cluster in each node by an isotropic distribution. Therefore, there is room for improvement in the cluster representation. For future work, we would like to extend the cluster representation to an anisotropic distribution to further reduce the approximation error.

Acknowledgments

We would like to thank ArtcoreStudios for the Museum scene [2022] and the Amazon Lumberyard team for the Bistro scene [2017]. We would also like to thank the anonymous reviewers for their insightful comments and constructive suggestions. We also appreciate the members of Advanced Rendering Research Group for their support.

References

Ciprian Apetrei. 2014. Fast and Simple Agglomerative LBVH Construction. In *CGVC '14*. https://doi.org/10.2312/cgvc.20141206

ArtcoreStudios. 2022. Museum Environment Kit. https://www.unrealengine.com/marketplace/en-US/product/museum-environment-kit

Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, and Suvrit Sra. 2005. Clustering on the Unit Hypersphere Using von Mises-Fisher Distributions. *J. Mach. Learn. Res.* 6 (2005), 1345–1382.

Petr Beckmann and André Spizzichino. 1963. *Scattering of Electromagnetic Waves from Rough Surfaces.* MacMillan.

Christopher Bingham. 1974. An Antipodally Symmetric Distribution on the Sphere. *Ann. Statist.* 2, 6 (1974), 1201–1225. https://doi.org/10.1214/aos/1176342874

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 (2020), 17 pages. https://doi.org/10.1145/3386569.3392481

Petrik Clarberg, Wojciech Jarosz, Tomas Akenine-Möller, and Henrik Wann Jensen. 2005. Wavelet Importance Sampling: Efficiently Evaluating Products of Complex Functions. *ACM Trans. Graph.* 24, 3 (2005), 1166–1175. https://doi.org/10.1145/1073204.1073328

Estevez Alejandro Conty and Christopher Kulla. 2018. Importance Sampling of Many Lights with Adaptive Tree Splitting. *Proc. ACM Comput. Graph. Interact. Tech.* 1, 2 (2018), 25:1–25:17. https://doi.org/10.1145/3233305

Robert L. Cook and Kenneth E. Torrance. 1982. A Reflectance Model for Computer Graphics. *ACM Trans. Graph.* 1, 1 (1982), 7–24. https://doi.org/10.1145/357290.357293

Roc R. Currius, Dan Dolonius, Ulf Assarsson, and Erik Sintorn. 2020. Spherical Gaussian Light-field Textures for Fast Precomputed Global Illumination. *Comput. Graph. Forum* 39, 2 (2020), 133–146. https://doi.org/10.1111/cgf.13918

Carsten Dachsbacher, Jaroslav Křivánek, Miloš Hašan, Adam Arbree, Bruce Walter, and Jan Novák. 2014. Scalable Realistic Rendering with Many-Light Methods. *Comput. Graph. Forum* 33, 1 (2014), 88–104. https://doi.org/10.1111/cgf.12256

Tomáš Davidovič, Jaroslav Křivánek, Miloš Hašan, and Philipp Slusallek. 2014. Progressive Light Transport Simulation on the GPU: Survey and Improvements. *ACM Trans. Graph.* 33, 3 (2014), 29:1–29:19. https://doi.org/10.1145/2602144

Jonathan Dupuy, Eric Heitz, and Laurent Belcour. 2017. A spherical cap preserving parameterization for spherical distributions. *ACM Trans. Graph.* 36, 4, Article 139 (2017), 12 pages. https://doi.org/10.1145/3072959.3073694

Ronald Aylmer Fisher. 1953. Dispersion on a sphere. *Proc. R. Soc. Lond. Ser. A* 217, 1130 (1953), 295–305. https://doi.org/10.1098/rspa.1953.0064

Iliyan Georgiev, Jamie Portsmouth, Zap Andersson, Adrien Herubel, Alan King, Shinji Ogaki, and Frédéric Servant. 2019. Autodesk Standard Surface. https://autodesk.github.io/standard-surface/

Eric Heitz. 2014. Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs. *JCGT* 3, 2 (2014), 48–107. http://jcgt.org/published/0003/02/03/

Eric Heitz, Jonathan Dupuy, Cyril Crassin, and Carsten Dachsbacher. 2015. The SGGX Microflake Distribution. *ACM Trans. Graph.* 34, 4, Article 48 (2015), 11 pages. https://doi.org/10.1145/2766988

Stephen Hill and Dan Baker. 2012. Rock-Solid Shading: Image Stability Without Sacrificing Detail. In *SIGGRAPH '12 Course: Advances in Real-Time Rendering in 3D Graphics and Games*.

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

Yuchi Huo, Shihao Jin, Tao Liu, Wei Hua, Rui Wang, and Hujun Bao. 2020. Spherical Gaussian-based Lightcuts for Glossy Interreflections. *Comput. Graph. Forum* 39, 6 (2020), 192–203. https://doi.org/10.1111/cgf.14011

Anton S. Kaplanyan, Stephen Hill, Anjul Patney, and Aaron Lefohn. 2016. Filtering Distributions of Normals for Shading Antialiasing. In *HPG '16*. 151–162. https://doi.org/10.2312/hpg.20161201

Brian Karis. 2013. Real Shading in Unreal Engine 4. In *SIGGRAPH '13 Course: Physically based Shading in Theory and Practice*. Article 22, 8 pages. https://doi.org/10.1145/2504435.2504457

Alexander Keller. 1997. Instant Radiosity. In *SIGGRAPH '97*. 49–56. https://doi.org/10.1145/258734.258769

Sébastien Lagarde. 2014. Moving Frostbite to Physically Based Rendering. In *SIGGRAPH '14 Course: Physically Based Shading in Theory and Practice*. 23:1–23:8. https://doi.org/10.1145/2614028.2615431

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 (2022), 23 pages. https://doi.org/10.1145/3528223.3530158

Daqi Lin and Cem Yuksel. 2020. Real-Time Stochastic Lightcuts. *Proc. ACM Comput. Graph. Interact. Tech.* 3, 1, Article 5 (2020), 18 pages. https://doi.org/10.1145/3384543

Yifan Liu, Kun Xu, and Ling-Qi Yan. 2019. Adaptive BRDF-Oriented Multiple Importance Sampling of Many Lights. *Comput. Graph. Forum* 38, 4 (2019), 123–133. https://doi.org/10.1111/cgf.13776

Amazon Lumberyard. 2017. Amazon Lumberyard Bistro, Open Research Content Archive (ORCA). http://developer.nvidia.com/orca/amazon-lumberyard-bistro

Michael D. McCool and Peter K. Harwood. 1997. Probability trees. In *GI '97*. 37–46. https://doi.org/10.20380/GI1997.05

Julian Meder and Beat Brüderlin. 2018. Hemispherical Gaussians for Accurate Light Integration. In *ICCVG' 18*. 3–15. https://doi.org/10.1007/978-3-030-00692-1_1

Daniel Meister, Paritosh Kulkarni, Aaryaman Vasishta, and Takahiro Harada. 2024. HIPRT: A Ray Tracing Framework in HIP. *Proc. ACM Comput. Graph. Interact. Tech.* 7, 3, Article 44 (2024), 18 pages. https://doi.org/10.1145/3675378

Pierre Moreau, Matt Pharr, and Petrik Clarberg. 2019. Dynamic Many-Light Sampling for Real-Time Ray Tracing. In *HPG '19*. 21–26. https://doi.org/10.2312/hpg.20191191

Kosuke Nabata, Kei Iwasaki, Yoshinori Dobashi, and Tomoyuki Nishita. 2016. An Error Estimation Framework for Many-Light Rendering. *Comput. Graph. Forum* 35, 7 (2016), 431–439. https://doi.org/10.1111/cgf.13040

David Neubelt and Matt Pettineo. 2015. Advanced Lighting R&D at Ready At Dawn Studios. In *SIGGRAPH '15 Course: Physically Based Shading in Theory and Practice*. Article 22, 8 pages. https://doi.org/10.1145/2776880.2787670

Matt Pettineo and Stephen Hill. 2016. SG Series Part 3: Diffuse Lighting From an SG Light Source. https://therealmjp.github.io/posts/sg-series-part-3-diffuse-lighting-from-an-sg-light-source/

Justin F. Talbot. 2005. *Importance Resampling for Global Illumination*. Master's thesis. Brigham Young U.

Yusuke Tokuyoshi. 2015. Virtual Spherical Gaussian Lights for Real-time Glossy Indirect Illumination. *Comput. Graph. Forum* 34, 7 (2015), 89–98. https://doi.org/10.1111/cgf.12748

Yusuke Tokuyoshi. 2022. Accurate Diffuse Lighting from Spherical Gaussian Lights. In *SIGGRAPH '22 Posters*. Article 35, 2 pages. https://doi.org/10.1145/3532719.3543209

Yusuke Tokuyoshi. 2024. Fast Indirect Illumination Using Two Virtual Spherical Gaussian Lights. https://github.com/yusuketokuyoshi/VSGL

Yusuke Tokuyoshi and Anton S. Kaplanyan. 2021. Stable Geometric Specular Antialiasing with Projected-Space NDF Filtering. *JCGT* 10, 2 (2021), 31–58. http://jcgt.org/published/0010/02/02/

T. S. Trowbridge and Karl P. Reitz. 1975. Average Irregularity Representation of a Rough Surface for Ray Reflection. *J. Opt. Soc. Am.* 65, 5 (1975), 531–536. https://doi.org/10.1364/JOSA.65.000531

Yu-Ting Tsai and Zen-Chung Shih. 2006. All-Frequency Precomputed Radiance Transfer Using Spherical Radial Basis Functions and Clustered Tensor Approximation. *ACM Trans. Graph.* 25, 3 (2006), 967–976. https://doi.org/10.1145/1141911.1141981

Eric Veach and Leonidas J. Guibas. 1995. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *SIGGRAPH '95*. 419–428. https://doi.org/10.1145/218380.218498

Petr Vévoda, Ivo Kondapaneni, and Jaroslav Křivánek. 2018. Bayesian online regression for adaptive direct illumination sampling. *ACM Trans. Graph.* 37, 4, Article 125 (2018), 12 pages. https://doi.org/10.1145/3197517.3201340

Bruce Walter, Adam Arbree, Kavita Bala, and Donald P. Greenberg. 2006. Multidimensional Lightcuts. *ACM Trans. Graph.* 25, 3 (2006), 1081–1088. https://doi.org/10.1145/1141911.1141997

Bruce Walter, Sebastian Fernandez, Adam Arbree, Kavita Bala, Michael Donikian, and Donald P. Greenberg. 2005. Lightcuts: A Scalable Approach to Illumination. *ACM Trans. Graph.* 24, 3 (2005), 1098–1107. https://doi.org/10.1145/1073204.1073318

Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet Models for Refraction Through Rough Surfaces. In *EGSR '07*. 195–206. https://doi.org/10.2312/EGWR/EGSR07/195-206

Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. 2009. All-Frequency Rendering of Dynamic, Spatially-Varying Reflectance. *ACM Trans. Graph.* 28, 5 (2009), 133:1–133:10. https://doi.org/10.1145/1618452.1618479

Yu-Chen Wang, Yu-Ting Wu, Tzu-Mao Li, and Yung-Yu Chuang. 2021. Learning to cluster for rendering with many lights. *ACM Trans. Graph.* 40, 6, Article 277 (2021), 10 pages. https://doi.org/10.1145/3478513.3480561

Kun Xu, Yan-Pei Cao, Li-Qian Ma, Zhao Dong, Rui Wang, and Shi-Min Hu. 2014. A Practical Algorithm for Rendering Interreflections with All-frequency BRDFs. *ACM Trans. Graph.* 33, 1 (2014), 10:1–10:16. https://doi.org/10.1145/2533687

Kun Xu, Wei-Lun Sun, Zhao Dong, Dan-Yong Zhao, Run-Dong Wu, and Shi-Min Hu. 2013. Anisotropic Spherical Gaussians. *ACM Trans. Graph.* 32, 6 (2013), 209:1–209:11. https://doi.org/10.1145/2508363.2508386

Cem Yuksel. 2021. Stochastic Lightcuts for Sampling Many Lights. *IEEE Trans. Vis. and Comput. Graph.* 27, 10 (2021), 4049–4059. https://doi.org/10.1109/TVCG.2020.3001271