Eurographics Symposium on Rendering 2026
I. Gkioulekas and A. Jarabo
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 45 (2026), Number 4

One-more-vertex Next-Event Estimation with Hierarchical Geometry Sampling

Jorge Garcia-Pueyo*[1], Nestor Monzon*[1], Adrian Jarabo[2] and Adolfo Muñoz[1]

[1]Universidad de Zaragoza, I3A, Spain
[2]Independent researcher, Zaragoza, Spain
*Equal contribution

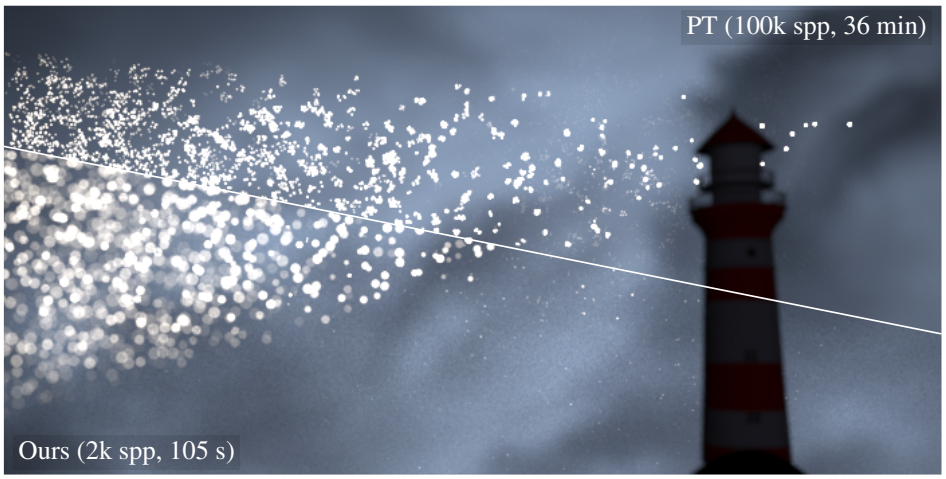


Figure 1: *Our method accelerates rendering of challenging scenes with highly-contributing sparse geometry, such as in this scene with falling snowflakes illuminated from the* Lighthouse*'s spotlight, and rendered from a thin-lens camera. Path tracing (top right) struggles at finding the small contributing flakes even at very large sample counts. In contrast, our method (bottom left) finds three-vertex paths, by sampling vertices in the geometry based on the throughput, which allows to drastically reduce variance in a fraction of time.*

Abstract

Robust next-event estimation (NEE) remains a challenge in scenes characterized by sparse or small-scale geometry where indirect illumination is the primary transport mechanism. In these scenes, traditional path construction, which relies on local directional sampling, often fails to find intersections with the sparse geometry, and standard NEE also struggles as it typically connects vertices directly to emitters, failing when those connections are occluded or require intermediate bounces. We propose a novel approach that constructs paths via direct geometry sampling. Instead of relying on stochastic ray casting, we repurpose the scene's bounding volume hierarchy (BVH) as a hierarchical sampling structure. By performing a stochastic top-down traversal, we transform the selection of the next path vertex into a hierarchical problem. To prioritize high-throughput connections, the traversal is guided by a proxy contribution function evaluated at each internal node. This function leverages aggregated statistics of the geometry contained in the BVH nodes to efficiently estimate contribution during traversal. We demonstrate orders of magnitude improvements in complex scenarios such as indirect illumination from sparse geometry or rendering discrete scattering media.

CCS Concepts

• ***Computing methodologies*** → ***Rendering;***

1. Introduction

Monte Carlo rendering techniques typically compute light transport within a scene by stochastically sampling paths that connect light sources and sensors. The efficiency of these methods relies on the ability to sample paths with high throughput, which is usually done via importance sampling some angular direction at each vertex. A fundamental assumption in these techniques is that the scene geometry is sufficiently dense or large-scale such that sampled directions have a reasonable probability of finding the relevant surfaces. However, this assumption fails in scenes characterized by sparse or small-scale geometry, which are relatively common in the real world: Discrete media such as falling snow or suspended dust, trees or foliage, highly-detailed specular geometry... In such environments, the projected solid angle of relevant geometry from the light sources or eye-path vertices tends to be very small, and therefore, the probability of stochastically reaching this geometry is very low, leading to excessive variance.

Standard next-event estimation (NEE) also struggles in these cases, as it is generally limited to direct connections between vertices and emitters, failing when those connections are occluded or require intermediate scattering events. Even global techniques like path guiding [MGN17], that rely on learned global radiance to sample local directions, suffer in these environments for two reasons: On one hand these path-guiding methods rely on an initial exploration phase to compute radiance distributions; if the initial paths cannot randomly find the sparse geometry, the learning process lacks the bootstrap data required to converge. Furthermore, the angular and spatial resolutions of the guiding structures are also often limited, so even if the sampled direction is approximately correct, it might happen that the sampled paths do not intersect the geometry.

In this paper, we tackle the sampling of paths in these challenging setups. In particular, we focus on sampling an additional intermediate vertex to connect two points at the end of the eye and light subpaths, respectively, so that sampling is approximately proportional to the path throughput. We propose hierarchical geometry sampling, a novel approach where instead of randomly sampling paths in the hope of finding surfaces, we leverage the scene's bounding volume hierarchy (BVH) to sample vertices directly on the geometry. This hierarchical sampling is guided by a practical importance function that allows to explicitly find and connect high-throughput vertices that would be difficult to find via directional sampling. This transforms the computation of the path's next vertex into a hierarchical selection problem. Our practical importance function is based on pre-computed aggregated statistics of the geometry that approximate the expected light behavior in the scene.

We use this throughput-based geometry sample to, instead of deterministically connecting a sensor path vertex and a light source through direct next-event estimation, connect them through an additional vertex importance sampled from the geometry of the scene. We call our approach *one-more-vertex next-event estimation*. We demonstrate our technique in scenes with very complex geometry and light paths, that are very difficult to sample even with path guiding given their high-frequency nature in the directional domain, including discrete media. Despite the overhead introduced by our technique, we get dramatic speed ups in these types of scenes. Moreover, by setting the eye vertex in the sensor, we demonstrate our technique for rendering volumetric discrete media with suspended particles, which allows us to render complex distributed effects (Figure 1) orders of magnitude faster than path tracing. We will publish our code for the research community in our project website: `https://graphics.unizar.es/projects/OneMoreVertex26/`.

2. Related work

Next-event estimation. Standard next-event estimation (NEE) connects path vertices directly to an emitter, but fails when the connection must traverse intermediate specular interfaces or scattering surfaces. Extended-NEE techniques address this by inserting additional vertices along the connection subpath. Walter et al. [WZHB09] and Holzschuch [Hol15] found additional vertices over refractive surfaces in the context of computing single scattering in bounded media. Koerner et al. [KNK*16] placed an intermediate vertex inside the medium to directly satisfy refraction laws during subsurface scattering with refractive boundaries. More generally, manifold-based NEE [HDF15; GH24; ZGJ20] use manifold walks constraints to close-to-delta surfaces to find specular-chain connection to the light. In this context, Loubet et al. [LZHJ20] approaches manifold NEE similar to us, using a hierarchical structure traversed at render time. While their approach focuses on specular NEE and relies on a dedicated pre-rendering tracing pass to bound per-triangle caustic footprints, our method derives node statistics directly from the scene geometry and materials.

In the context of bidirectional methods, Su et al. [SLY*24] build length-two subpaths with a proxy-based importance function, optimizing the final path construction in bidirectional path tracing. In the context of transient rendering, Jarabo et al. [JMM*14] and Pediredla et al. [PVG19] added an extra vertex for ellipsoidal NEE connections to gain explicit control on the path length, in media and surfaces respectively.

Extra vertices for next-event estimation are common in participating media, where the position of these vertices have more degrees of freedom than in surfaces. This idea has been widely used in the context of neutron transport [Kal63; KC77]. In graphics, Kulla and Fajardo [KF11] sampled along the view ray based on the squared-distance with respect to the light source. Georgiev et al. [GKH*13] jointly sampled low-order volumetric scattering subpaths, accounting simultaneously for phase functions and inverse-square geometric terms along two-bounce subpaths, and Weber et al. [WHD17] extended this to arbitrarily many intermediate vertices in dense, forward-scattering media. Closer to us, Hanika et al. [HWD22] jointly importance sampled the phase function and the two inverse-square distance terms of an additional scattering event in thin, forward-scattering volumes. More recently, Schüssler et al. [SHD24] generated connection subpaths of arbitrary vertex count by first sampling phase functions and then jointly importance sampling all edge distances via analytic marginalization over a global scale factor.

Our method shares the extended-NEE philosophy of inserting intermediate vertices between subpaths, but targets sparse-geometry scenes where we importance-sample the intermediate vertex directly on scene geometry via hierarchical traversal.

Rendering sparse geometry. Rendering sparse small geometry, such as glints in surfaces or particles in discrete media, is complex, due to the inherent difficulty of hitting contributing primitives [ZZX*22]. In the context of glinty surfaces, Yan et al. [YHJ*14; YHMR16] fitted directional and spatio-directional distributions for finding contributing normals from high-resolution normal maps. Follow up work accelerated this approach using tensor decomposition [DLW*22] or regularization [FWW*22]. Instead of explicit normal maps, Jakob et al. [JHY*14] used a procedural approach for generating the discrete microfacet surface, which was later extended to iridescence [GCGP18], transparent surfaces [ZWWH25], and to allow fast global illumination [WWH18]. In the context of discrete media, several works [MWM07; MPH*15; MPG*16] accelerate high-order scattering in dense particulate media by using homogeneized continuous statistical bulk optical properties, while for sparser media Guo et al. [GHC*22] preserve the medium discreteness by intersecting point-based scatterers using beam tracing. In contrast, we tackle explicit geometry and sample it based on the estimated contribution of the full three-vertex path.

Auxiliary structures for sampling. Several works have explored auxiliary data structures that improve sampling. Deng et al. [DHC*21] represented light transport as a graph whose edges connect surface clusters, using graph-based operations to discover high-energy connections. Closer to our approach, Tokuyoshi and Harada [TH19] leveraged the BVH to perform hierarchical Russian roulette for vertex connections in bidirectional methods, culling low-contributing connections. Bitterli et al. [BNJ15] proposed to use explicit portals to account for visibility in environment map sampling. The challenge of sampling geometric features that are nearly impossible to find through hemispherical sampling also arises in differentiable rendering, where gradient computation requires integrating over boundary (silhouette) edges of the scene geometry [LADL18; ZMY*20]. Our work addresses a related problem by sampling the explicit geometry, but finding the most contributing vertices. In the context of many-lights evaluation and sampling, the seminal work by Walter et al. [WFA*05] introduced a hierarchical approach that clusters lights into a tree and uses a conservative error bound to prune nodes. Recent techniques extend this to stochastic sampling using BVH traversal [CK18; LY20], two-level hierarchies [MPC19] or BRDF-aware sampling of many-lights [LXY19]. We follow a similar approach, but focusing on three-vertex paths instead of direct subpath connections via NEE, which requires storing geometry statistics instead of light statistics.

Path guiding. Path guiding techniques learn an approximation of the incident radiance field during rendering and use it to inform direction sampling at each path vertex. Several approaches have been proposed, including fitting mixture models [VKŠ*14; DPÖM22], hierarchical grids [MGN17], neural networks [MMR*19; RMW*25], voxel-based guiding structures [LCHL24], and screen-space parametric mixtures [Der22]. Instead of storing the radiance distribution for sampling, which requires a cache phase to build the distribution function, we directly sample an additional vertex with probability approximately proportional to the path throughput by using an auxiliary hierarchical structure.

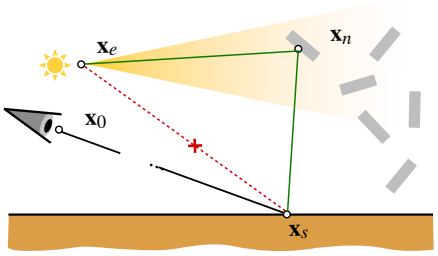


Figure 2: *Standard next-event estimation connects sensor vertex $\mathbf{x}_s$ directly to an emitter vertex $\mathbf{x}_e$. In scenes where $\mathbf{x}_s$ is mostly illuminated through an indirect light from the scene geometry $\mathcal{G}$, that direct connection has low to zero contribution (red dashed line). Our technique tackles this by importance-sampling an intermediate vertex $\mathbf{x}_n$ on the scene geometry $\mathcal{G}$. This allows us to efficiently sample high-throughput indirect paths (green) where the geometry acts as the primary source of reflected illumination for $\mathbf{x}_s$. Note that our approach accounts for the case where $\mathbf{x}_s$ is on the sensor $\mathbf{x}_s = \mathbf{x}_0$.*

3. Problem statement

Integral formulation. We formulate our problem as an integral in area measure over all the three-vertex paths that connect a vertex on the light source $\mathbf{x}_e$ and a vertex on the sensor path $\mathbf{x}_s$, through an intermediate vertex $\mathbf{x}_n$ that lies within the geometry of the scene $\mathcal{G}$, which is our integration domain:

$$L(\mathbf{x}_s \leftrightarrow \mathbf{x}_e) = \int_{\mathcal{G}} f(\mathbf{x}_s \leftrightarrow \mathbf{x}_n \leftrightarrow \mathbf{x}_e)\, dA(\mathbf{x}_n), \tag{1}$$

where $f(\mathbf{x}_s \leftrightarrow \mathbf{x}_n \leftrightarrow \mathbf{x}_e)$ is the measurement contribution function of the three-vertex subpath (Figure 2), defined as

$$\begin{aligned} f(\mathbf{x}_s \leftrightarrow \mathbf{x}_n \leftrightarrow \mathbf{x}_e) = {} & \mathrm{f}_{s\to n} \cdot G_{s\leftrightarrow n} \cdot V_{s\leftrightarrow n} \\ & \cdot \mathrm{f}_{s\to n\to e} \cdot G_{n\leftrightarrow e} \cdot V_{n\leftrightarrow e} \cdot S_{e\to n}, \end{aligned} \tag{2}$$

where $\mathrm{f}_{s\to n\to e}$ is the BSDF at $\mathbf{x}_n$ with incident and outgoing directions $\frac{\mathbf{x}_n-\mathbf{x}_s}{|\mathbf{x}_n-\mathbf{x}_s|}$ and $\frac{\mathbf{x}_e-\mathbf{x}_n}{|\mathbf{x}_e-\mathbf{x}_n|}$ respectively, $S_{e\to n}$ is the radiance emitted from $\mathbf{x}_e$ to $\mathbf{x}_n$, and $\mathrm{f}_{s\to n}$ is a directional importance function at $\mathbf{x}_s$ to $\mathbf{x}_n$, which is the sensor's importance $\mathrm{f}_{s\to n} = W_{s\to n}$ if $\mathbf{x}_s$ is on the sensor, or the scattering function at $\mathbf{x}_s$, $\mathrm{f}_{s\to n} = \mathrm{f}_{s-1\to s\to n}$ otherwise. Finally, $G_{i\leftrightarrow j}$ and $V_{i\leftrightarrow j}$ are the geometric $G_{i\leftrightarrow j} = \frac{\cos\theta_{i,j} \cdot \cos\theta_{j,i}}{|\mathbf{x}_i-\mathbf{x}_j|^2}$, and visibility terms respectively, where $\cos\theta_{i,j}$ is the cosine of the angle between the surface normal at $\mathbf{x}_i$ and the direction $\frac{\mathbf{x}_j-\mathbf{x}_i}{|\mathbf{x}_j-\mathbf{x}_i|}$.

Problem. We aim to solve Equation (1) via a Monte Carlo estimator with low variance. For that, our goal is to importance sample the intermediate vertex $\mathbf{x}_n$ with a probability distribution function (PDF) $p(\mathbf{x}_n|\mathbf{x}_s,\mathbf{x}_e)$ that is as proportional as possible to the contribution $f(\mathbf{x}_s \leftrightarrow \mathbf{x}_n \leftrightarrow \mathbf{x}_e)$ as defined by Equation (2). Formally, we can define our north-star PDF as

$$p(\mathbf{x}_n|\mathbf{x}_s,\mathbf{x}_e) = \frac{f(\mathbf{x}_s \leftrightarrow \mathbf{x}_n \leftrightarrow \mathbf{x}_e)}{\int_{\mathcal{G}} f(\mathbf{x}_s \leftrightarrow \mathbf{x}_n \leftrightarrow \mathbf{x}_e)\, dA(\mathbf{x}_n)}, \tag{3}$$

which cannot be computed in practice, since the denominator is precisely the function we are trying to solve. Also, $\mathbf{x}_s$ and $\mathbf{x}_e$ vary

across sampled paths, so the PDF cannot be practically precomputed. Nevertheless, both vertices are known prior to sampling $\mathbf{x}_n$: $\mathbf{x}_s$ is known from the sensor path and $\mathbf{x}_e$ is obtained by first selecting a light source uniformly and then drawing a point uniformly over its surface area (Section 5).

Overview. In this work we assume that the scene geometry is made of surface primitives small enough so that points $\mathbf{x}_n$ inside a primitive $\mathcal{P} \in \mathcal{G}$ have similar contribution $f(\mathbf{x}_s \leftrightarrow \mathbf{x}_n \leftrightarrow \mathbf{x}_e)$. This is common in modern 3D scenes, where geometries are heavily tessellated. Therefore, the most dominant factor of $p(\mathbf{x}_n|\mathbf{x}_s,\mathbf{x}_e)$ is the selection of the primitive. However, stochastically computing the PDF of each primitive is itself impractical, with cost $O(N)$. To make the problem tractable, we devise a hierarchical structure for all the primitives of the scene (Figure 3(a)), in which each node contains aggregated information of all contained primitives (Figure 3(b)). As we traverse the tree, we use the aggregated information to estimate a probability distribution per-node as a function of $\mathbf{x}_s$ and $\mathbf{x}_e$, which we use to randomly select which node we will continue evaluating in the hierarchy recursively (Figure 3(c)). This allows us to efficiently compute a reasonably accurate probability distribution per primitive in render-time with logarithmic cost.

4. Hierarchical One-more-vertex Sampling

4.1. Hierarchical sampling

In order to compute our PDF for Equation (1), we first decompose Equation (3) as the product of two terms

$$p(\mathbf{x}_n|\mathbf{x}_s,\mathbf{x}_e) = p(\mathcal{P}|\mathbf{x}_s,\mathbf{x}_e) \cdot p(\mathbf{x}_n|\mathcal{P},\mathbf{x}_s,\mathbf{x}_e), \tag{4}$$

where $\mathcal{P} \in \mathcal{G}$ is a primitive so that $\mathbf{x}_n \in \mathcal{P}$, and $p(\mathcal{P}|\mathbf{x}_s,\mathbf{x}_e)$ and $p(\mathbf{x}_n|\mathcal{P},\mathbf{x}_s,\mathbf{x}_e)$ are the probability of sampling the primitive $\mathcal{P}$ and the conditional probability of sampling $\mathbf{x}_n$ in $\mathcal{P}$, defined as

$$p(\mathcal{P}|\mathbf{x}_s,\mathbf{x}_e) = \frac{\int_{\mathcal{P}} f(\mathbf{x}_s \leftrightarrow \mathbf{x}_n \leftrightarrow \mathbf{x}_e)\, dA(\mathbf{x}_n)}{\sum_{\mathcal{P}' \in \mathcal{G}} \int_{\mathcal{P}'} f(\mathbf{x}_s \leftrightarrow \mathbf{x}_n \leftrightarrow \mathbf{x}_e)\, dA(\mathbf{x}_n)} \tag{5}$$

$$p(\mathbf{x}_n|\mathcal{P},\mathbf{x}_s,\mathbf{x}_e) = \frac{f(\mathbf{x}_s \leftrightarrow \mathbf{x}_n \leftrightarrow \mathbf{x}_e)}{\int_{\mathcal{P}} f(\mathbf{x}_s \leftrightarrow \mathbf{x}_n \leftrightarrow \mathbf{x}_e)\, dA(\mathbf{x}_n)}, \tag{6}$$

where $\bigcup_{\mathcal{P}' \in \mathcal{G}} = \mathcal{G}$, and thus $\int_{\mathcal{G}} f(\mathbf{x}_s \leftrightarrow \mathbf{x}_n \leftrightarrow \mathbf{x}_e)\, dA(\mathbf{x}_n) = \sum_{\mathcal{P}' \in \mathcal{G}} \int_{\mathcal{P}'} f(\mathbf{x}_s \leftrightarrow \mathbf{x}_n \leftrightarrow \mathbf{x}_e)\, dA(\mathbf{x}_n)$.

Sampling the primitive $\mathcal{P}$ directly from Equation (5) requires computing a tabulated PDF for each primitive, which is impractical since it is dependent of each pair of arbitrary vertices $\mathbf{x}_s, \mathbf{x}_e$. Instead, inspired by scalable many-lights techniques [WFA*05; CK18; LY20], we opt for a hierarchical approach, where we group the individual primitives in aggregated nodes $\mathcal{N}$. The top node $\mathcal{N}^0$ contains all geometry in the scene, and as we move down the structure each node contains smaller subsets. This way, the geometry set $\mathcal{G}_i^d$ in a node $\mathcal{N}_i^d$ at depth d, is subdivided in the node's children $\mathcal{N}_j^{d+1}$ as $\bigcup_{j \in \mathcal{N}_i^d} \mathcal{G}_j^{d+1} = \mathcal{G}_i^d$. Each node has an *importance weight* $w(\mathcal{N}_i^d, \mathbf{x}_s, \mathbf{x}_e)$ defined as

$$w(\mathcal{N}_i^d, \mathbf{x}_s, \mathbf{x}_e) = \sum_{\mathcal{P}' \in \mathcal{G}_i^d} \int_{\mathcal{P}'} f(\mathbf{x}_s \leftrightarrow \mathbf{x}_n \leftrightarrow \mathbf{x}_e)\, dA(\mathbf{x}_n). \tag{7}$$

With that hierarchical structure, we pose sampling $\mathcal{P}$ as a stochastic traversal of the structure (Figure 3), so that when we are at $\mathcal{N}_i^d$, we decide which of their children $\mathcal{N}_j^{d+1}$ we should recurse with probability (note that the probability is also dependent on $\mathbf{x}_s$ and $\mathbf{x}_e$)

$$p(\mathcal{N}_j^{d+1} \mid \mathcal{N}_i^d) = \frac{w\left(\mathcal{N}_j^{d+1}, \mathbf{x}_s, \mathbf{x}_e\right)}{\sum_{k \in \mathcal{N}_i^d} w\left(\mathcal{N}_k^{d+1}, \mathbf{x}_s, \mathbf{x}_e\right)} = \frac{w\left(\mathcal{N}_j^{d+1}, \mathbf{x}_s, \mathbf{x}_e\right)}{w\left(\mathcal{N}_i^{d}, \mathbf{x}_s, \mathbf{x}_e\right)}, \tag{8}$$

which is simply a discrete PDF. As a consequence of our sampling scheme, the probability of sampling $\mathbf{x}_n$ is the product of all child-selection probabilities along the root-to-leaf traversal path $\bar{\mathcal{N}} = \{\mathcal{N}_0, \ldots, \mathcal{N}_{\text{leaf}}\}$ until we get a primitive $\mathcal{P}$, and the probability of sampling point $\mathbf{x}_n$ within the sampled primitive, which leads to a PDF

$$p(\mathbf{x}_n|\mathbf{x}_s,\mathbf{x}_e) = \left(\prod_{i=1}^{|\bar{\mathcal{N}}|} p(\mathcal{N}_i \mid \mathcal{N}_{i-1})\right) \cdot p(\mathbf{x}_n \mid \mathcal{N}_{\text{leaf}}, \mathbf{x}_s, \mathbf{x}_e). \tag{9}$$

By expanding its terms is easy to verify that Equation (9) results in Equation (4). Unfortunately, computing $p(\mathcal{N}_j^{d+1} \mid \mathcal{N}_i^d)$ is far from practical, since it requires iterating over all primitives within each node. In the following, we define the importance weight $w(\mathcal{N}, \mathbf{x}_s, \mathbf{x}_e)$ of each node probabilistically, which eventually allows us to quickly compute it from the statistics of the primitives within the node.

4.2. Statistical node contribution

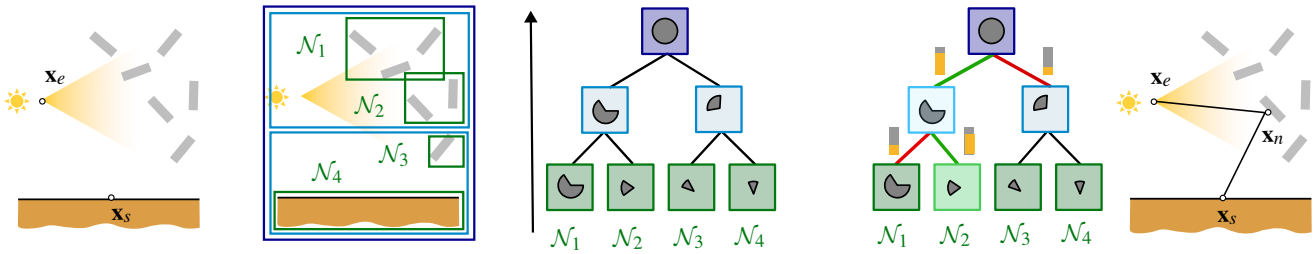
As described above, an exact expression for the node importance weight in Equation (7) is nevertheless untractable, so we aim to obtain a reasonably good approximation to it from a set of compact per-node statistics that can be precalculated.

For that, instead of posing the integral in the manifold defined by $\mathcal{G}_{\mathcal{N}}$, let us rewrite the contribution in terms of probability distributions describing $\mathbf{x}_n$, following:

$$\begin{aligned} w(\mathcal{N}, \mathbf{x}_s, \mathbf{x}_e) = \int_{\mathcal{V}_{\mathcal{N}}} \int_{S} \int_{B_{\mathcal{N}}} & D(\mathbf{x}_n) \cdot D(\mathbf{n}|\mathbf{x}_n) \cdot D(\beta|\mathbf{x}_n) \\ & \cdot \mathrm{f}_{s \to n} \cdot G_{s \leftrightarrow n} \cdot V_{s \leftrightarrow n} \\ & \cdot \mathrm{f}_{s \to n \to e} \cdot G_{n \leftrightarrow e} \cdot V_{n \leftrightarrow e} \cdot S_{e \to n}\, d\beta\, d\mathbf{n}\, d\mathbf{x}_n, \end{aligned} \tag{10}$$

where $D(\mathbf{x}_n)$ is the spatial distribution of surface positions within the node's volume $\mathcal{V}_{\mathcal{N}}$, $D(\mathbf{n}|\mathbf{x}_n)$ is the distribution of geometric normals over the unit sphere S conditioned on position, and $D(\beta|\mathbf{x}_n)$ is the distribution of material parameters over the space of material parameters $B_{\mathcal{N}}$, again conditioned on position. Note that $D(\mathbf{x}_n)$ corresponds to a manifold of the surface within the volume of the node, and $D(\mathbf{n}|\mathbf{x}_n)$ and $D(\beta|\mathbf{x}_n)$ correspond to delta manifolds following $D(\mathbf{n}|\mathbf{x}_n) = \delta(\mathbf{n} - \mathbf{n}_{\mathbf{x}_n})$ and $D(\beta|\mathbf{x}_n) = \delta(\beta - \beta_{\mathbf{x}_n})$, with $\mathbf{n}_{\mathbf{x}_n}$ and $\beta_{\mathbf{x}_n}$ the normal and material parameters at $\mathbf{x}_n$. However, similar to microfacet models or other level-of-detail techniques, we later use approximated distributions to make it practical.

Separated integral form. To make Equation (10) tractable, we expand the geometric terms and rearrange the factors that depend only on the position of $\mathbf{x}_n$, separating them from the ones that depend



a) Scene geometry and hierarchical structure

b) Bottom-up aggregation of primitive's statistics

c) Hierarchical sampling

Figure 3: ***Hierarchical One-more-vertex-Sampling Overview.*** *(a) We wish to importance sample high-throughput three-vertex subpaths between vertices $\mathbf{x}_s$, $\mathbf{x}_e$. Our method is based on a hierarchical structure of the scene primitives that (b) aggregates its directional, spatial, and material statistics (represented by the gray sectors) at each node. The statistics computation is done bottom-up by aggregating the subnodes' statistics. (c) These statistics enable to guide the hierarchical sampling of the intermediate vertex $\mathbf{x}_n$ by evaluating the approximated per-node weight importance $\tilde{w}(\mathcal{N},\mathbf{x}_s,\mathbf{x}_e)$ (represented by the yellow bars) for each node's children and choosing one proportionally to its weight.*

on the material properties and surface normal:

$$w(\mathcal{N},\mathbf{x}_s,\mathbf{x}_e) = \underbrace{\int_{\mathcal{V}_\mathcal{N}} \frac{D(\mathbf{x}_n)\cdot \mathrm{f}_{s\to n}\cdot\cos\theta_{s,n}\cdot V_{s\leftrightarrow n}\cdot V_{n\leftrightarrow e}\cdot\cos\theta_{e,n}\cdot S_{e\to n}}{|\mathbf{x}_s-\mathbf{x}_n|^2|\mathbf{x}_n-\mathbf{x}_e|^2}}_{\text{spatial}} \underbrace{\int_S\int_{B_\mathcal{N}} D(\mathbf{n}|\mathbf{x}_n)\cdot D(\beta|\mathbf{x}_n)\cdot\cos\theta_{n,s}\cdot\cos\theta_{n,e}\cdot \mathrm{f}_{s\to n\to e}\, d\beta d\mathbf{n}\, d\mathbf{x}_n}_{\text{normal and BSDF}}. \tag{11}$$

The outer integral captures the magnitudes that only depend on vertex position $\mathbf{x}_n$, which are the spatial distribution, the inverse-square falloff and the contribution from the external vertices $\mathbf{x}_s$ and $\mathbf{x}_e$ with their corresponding cosines. The two inner integrals correspond to the two material components, which are more challenging to aggregate within a node's properties, since all its components depend on the surface normal and material properties. In the following, we use Equation (11) to derive a practical approximated importance weight of a node, that is computationally cheap, based on aggregated properties and distributions at the node.

4.3. A practical node importance weight

Here we focus on approximating Equation (11) to compute a practical per-node importance weight $w(\mathcal{N},\mathbf{x}_s,\mathbf{x}_e)$ by introducing a set of simplifications and specific distribution functions and material model. We first assume that inside the node the spatial, directional and material statistics are fully decorrelated. We also define an upper bound for visibility, and set $V_{s\leftrightarrow n}=1$ and $V_{n\leftrightarrow e}=1$, which is a strong assumption, but allows avoiding expensive visibility computations. Finally, following the work of Zhou et al. [ZHR*25], we assume that we use a simplified Principled BSDF-like scattering function $\mathrm{f}_{s\to n\to e}$, which allows us to separate between diffuse $\mathrm{f}^{\mathrm{d}}_{s\to n\to e}$ and specular $\mathrm{f}^{\mathrm{s}}_{s\to n\to e}$ components. These assumptions simplify Equation (11) to an approximated weight $\tilde{w}(\mathcal{N},\mathbf{x}_s,\mathbf{x}_e)\approx w(\mathcal{N},\mathbf{x}_s,\mathbf{x}_e)$ with the form

$$\tilde{w}(\mathcal{N},\mathbf{x}_s,\mathbf{x}_e) = \int_{\mathcal{V}_\mathcal{N}} \tilde{w}_{\mathrm{sp}}(\mathbf{x}_n,\mathbf{x}_s,x_e)\cdot\tilde{w}_{\mathrm{dir}}(\mathbf{x}_n,\mathbf{x}_s,x_e)\, d\mathbf{x}_n, \tag{12}$$

where (removed function parameters for clarity)

$$\begin{aligned}
\tilde{w}_{\mathrm{sp}} &= \frac{D(\mathbf{x}_n)\cdot \mathrm{f}_{s\to n}\cdot\cos\theta_{s,n}\cdot\cos\theta_{e,n}\cdot S_{e\to n}}{|\mathbf{x}_s-\mathbf{x}_n|^2|\mathbf{x}_n-\mathbf{x}_e|^2},\\
\tilde{w}_{\mathrm{dir}} &= \tilde{w}^{\mathrm{d}}_{\mathrm{dir}}+\tilde{w}^{\mathrm{s}}_{\mathrm{dir}}\\
&= \int_S\int_{B_\mathcal{N}} D(\beta)\cdot D(\mathbf{n})\cdot\cos\theta_{n,s}\cdot\cos\theta_{n,e}\cdot \mathrm{f}^{\mathrm{d}}_{s\to n\to e}\, d\beta\, d\mathbf{n}\\
&+ \int_S\int_{B_\mathcal{N}} D(\beta)\cdot D(\mathbf{n})\cdot\cos\theta_{n,s}\cdot\cos\theta_{n,e}\cdot \mathrm{f}^{\mathrm{s}}_{s\to n\to e}\, d\beta\, d\mathbf{n}
\end{aligned}$$

where the normal and material terms are independent from the spatial domain.

Spatial representation. We simplify the spatial distribution inside the node by using a 3D Gaussian so $D(\mathbf{x}_n)=\mathrm{Gauss}(\mathbf{x}_n\mid\mu_x,\Sigma_x)$, with mean μ_x and covariance Σ_x, plus the aggregated total surface area $A_\mathcal{N}$ of the primitives inside $\mathcal{N}$.

Directional representation. The distribution of geometric normals within the node is represented as a single-lobe von Mises-Fisher (vMF) distribution $D(\mathbf{n})=\mathrm{vMF}(\mathbf{n}|\mu_n,\kappa_n)$, with mean direction μ_n and concentration parameter κ_n. A high concentration indicates that the primitives share nearly aligned normals; a low concentration indicates a wide angular spread.

Material representation. For the material parameters we opt for a simple approach, and assume that $D(\beta)=\delta(\beta-\beta_\mathcal{N})$ is simply a delta function centered at the weighted average of the material parameters inside the node $\beta_\mathcal{N}=\int_{\mathcal{G}_\mathcal{N}}\beta(\mathbf{x}_n)\,dA(\mathbf{x}_n)$.

Diffuse component. We assume a Lambertian BSDF with form

$$\mathrm{f}^{\mathrm{d}}_{s\to n\to e} = (1-\beta^m)\cdot\frac{\beta^c}{\pi}, \tag{13}$$

where β^c and β^m are the albedo and metallic-ness, respectively. Since $\mathrm{f}^{\mathrm{d}}_{s\to n\to e}$ depends only on β^c and β^m we can separate the

material integral from the normal integral

$$\tilde{w}^{\mathrm{d}}_{\mathrm{dir}} = \int_{B_{\mathbf{x}}} D(\beta)\cdot(1-\beta^m)\cdot\frac{\beta^c}{\pi}d\beta\cdot\int_S D(\mathbf{n})\cdot\cos\theta_{n,s}\cdot\cos\theta_{n,e}\,d\mathbf{n}$$
$$= \rho^{\mathrm{d}}\cdot\int_S D(\mathbf{n})\cdot\cos\theta_{n,s}\cdot\cos\theta_{n,e}\,d\mathbf{n}, \tag{14}$$

where $\rho^{\mathrm{d}} = \int_{B_{\mathbf{x}}} D(\beta)(1-\beta^m)\cdot\frac{\beta^c}{\pi}d\beta \approx (1-\beta^m_{\mathcal{N}})\frac{\beta^c_{\mathcal{N}}}{\pi}$. Since ρ^{d} depends only on precomputed material properties, this expression gets reduced to a cosine-weighted integral of the geometric normal distribution $D(\mathbf{n})$ represented by a vMF. This integral depends only on four scalars parameters: The elevation angles for directions $\mathbf{x}_n\to\mathbf{x}_s$ and $\mathbf{x}_n\to\mathbf{x}_e$ relative to μ_n, θ_s and θ_e respectively, the relative azimuthal difference ϕ between $\mathbf{x}_n\to\mathbf{x}_s$ and $\mathbf{x}_n\to\mathbf{x}_e$, and the vMF concentration κ_n. We precompute the inner integral and fit a polynomial P of degree four, which we measured to be 20% faster than a LUT, with no noticeable increase in error. This leads to a final diffuse importance weight $\tilde{w}^{\mathrm{d}}_{\mathrm{dir}} = \rho^{\mathrm{d}}\cdot P(\theta_s,\theta_e,\phi,\kappa_n)$.

Specular component. The Principled BSDF [BW12] uses a microfacet-based specular term with the form

$$\mathrm{f}^{\mathrm{s}}_{s\to n\to e} = \frac{D_{\mathrm{GGX}}(\omega_h|\alpha,\mathbf{n})\cdot\mathcal{G}(\omega_e,\omega_s|\alpha,\mathbf{n})}{4\cdot\cos\theta_{n,s}\cdot\cos\theta_{n,e}} \tag{15}$$
$$\cdot\left(\beta^m\cdot F(\omega_h,\omega_s;\beta^c)+(1-\beta^m)\cdot F(\omega_h,\omega_s;\beta^s)\right),$$

with $\beta^s\in[0,1]$ the specular value, $\omega_s = \frac{\mathbf{x}_s-\mathbf{x}_n}{|\mathbf{x}_s-\mathbf{x}_n|}$ and $\omega_e = \frac{\mathbf{x}_e-\mathbf{x}_n}{|\mathbf{x}_e-\mathbf{x}_n|}$ the input and output directions and $\omega_h = \frac{\omega_e+\omega_s}{|\omega_e+\omega_s|}$ their half-vector. The terms $D_{\mathrm{GGX}}(\omega_h|\alpha,\mathbf{n})$ and $\mathcal{G}(\omega_e,\omega_s|\alpha)$ are the GGX normal distribution function (NDF) parametrized by its roughness α and the Smith shadowing-masking term, respectively, both defined in the local space of the normal $\mathbf{n}$. $F(\omega_h,\omega_s;r)$ is the Schlick Fresnel approximation [Sch94].

Unlike the diffuse component, the specular component couples the geometric normal $\mathbf{n}$ with the half vector ω_h through the microfacet NDF $D_{\mathrm{GGX}}(\omega_h|\alpha)$ and the material properties. Plugging the specular model into $\tilde{w}^{\mathrm{s}}_{\mathrm{dir}}$, applying the distribution of material properties, and simplifying terms we get

$$\tilde{w}^{\mathrm{s}}_{\mathrm{dir}} = \rho^{\mathrm{d}}_{s\to n\to e}\int_S D(\mathbf{n})\cdot D_{\mathrm{GGX}}(\omega_h|\alpha_{\mathcal{N}},\mathbf{n}) \tag{16}$$
$$\cdot\,\mathcal{G}(\omega_e,\omega_s|\alpha_{\mathcal{N}},\mathbf{n})\cdot H^+_{n,s}\cdot H^+_{n,e}\,d\mathbf{n},$$

where $H^+_{s,n}$ (and similarly $H^+_{n,e}$) is the Heaviside function that evaluates zero if $\mathrm{dot}(\omega_s,\mathbf{n})<0$ and one otherwise, that appears after canceling the clamped cosines of the BSDF, $\alpha_{\mathcal{N}}$ is the node's mean roughness, and the aggregate specular albedo is

$$\rho^{\mathrm{d}}_{s\to n\to e} = \tfrac{1}{4}\left[(1-\mathcal{F}_c(\omega_h,\omega_s))\cdot\bar{R}+\mathcal{F}_c(\omega_h,\omega_s)\right], \tag{17}$$

with $\bar{R} = \beta^m_{\mathcal{N}}\cdot\beta^c_{\mathcal{N}}+(1-\beta^m_{\mathcal{N}})\cdot\beta^s_{\mathcal{N}}$ and $\mathcal{F}_c(\omega_h,\omega_s)$ is Schlick's power-of-five Fresnel approximation. $\beta^s_{\mathcal{N}}$ is the area-weighted aggregated specularity of the node $\mathcal{N}$. $\bar{R}$ depends only on material parameters so we precalculate it per node.

The integral still couples the geometric-normal distribution $D(\mathbf{n})$ with the microfacet NDF $D_{\mathrm{GGX}}(\omega_h|\alpha_{\mathcal{N}},\mathbf{n})$. To find an analytical approximation we introduce three assumptions. First, we remove the Heaviside functions: While this might overestimate the contribution from back-facing normals, our experiments have shown a minor effect on the node's probability function. Second, since the shadowing-masking term varies slowly, we assume it is constant over the integral domain and center it with respect to the distribution mean direction μ_n. Finally, we approximate the GGX microfacet distribution to a vMF lobe with equivalent concentration κ_m derived from the mean roughness $\bar{\alpha}$ [WRG*09].

With all these changes, we approximate $\tilde{w}^{\mathrm{s}}_{\mathrm{dir}}$ as

$$\tilde{w}^{\mathrm{s}}_{\mathrm{dir}} \approx \rho^{\mathrm{d}}_{s\to n\to e}\cdot\mathcal{G}(\omega_e,\omega_s|\alpha,\mu_n)\cdot\int_S D(\mathbf{n})\cdot D_{\mathrm{GGX}}(\omega_h|\alpha_{\mathcal{N}},\mathbf{n})\,d\mathbf{n}. \tag{18}$$

The remaining integral is the convolution between $D(\mathbf{n})$ and $D_{\mathrm{GGX}}(\omega_h|\alpha_{\mathcal{N}},\mathbf{n})$, which when represented as vMF distributions can be approximated as another vMF centered in μ_n and with concentration defined as the harmonic mean of the geometric (κ_n) and material (κ_m) concentrations [IDN12]:

$$\kappa_{\mathrm{conv}} = \frac{\kappa_n\cdot\kappa_m}{\kappa_n+\kappa_m}, \tag{19}$$

which results in a final specular weight $\tilde{w}^{\mathrm{s}}_{\mathrm{dir}}$ defined as

$$\tilde{w}^{\mathrm{s}}_{\mathrm{dir}} \approx \rho^{\mathrm{d}}_{s\to n\to s}\cdot\mathcal{G}(\omega_e,\omega_s|\alpha,\mu_n)\cdot\mathrm{vMF}(\omega_h|\mu_n,\kappa_{\mathrm{conv}}). \tag{20}$$

Final node importance weight. By plugging our approximated $\tilde{w}^{\mathrm{d}}_{\mathrm{dir}}$ and $\tilde{w}^{\mathrm{s}}_{\mathrm{dir}}$ in Equation (12) we have removed the inner integral, but the outer spatial integral still remains. Since this integral does not have a closed-form solution, we approximate it by evaluating the product $\tilde{w}_{\mathrm{sp}}(\mathbf{x}_n,\mathbf{x}_s,x_e)\cdot\tilde{w}_{\mathrm{dir}}(\mathbf{x}_n,\mathbf{x}_s,x_e)$ at set of sample points $\mathbf{x}'_n$. This gives us a final approximated node importance weight

$$\tilde{w}(\mathcal{N},\mathbf{x}_s,\mathbf{x}_e) = \tag{21}$$
$$\frac{1}{C}\sum_{\mathbf{x}'_n} D(\mathbf{x}'_n)\cdot\frac{\mathrm{f}_{s\to n'}\cdot\cos\theta_{s,n'}}{|\mathbf{x}_s-\mathbf{x}'_n|^2}\cdot\frac{\cos\theta_{e,n'}\cdot S_{e\to n'}}{|\mathbf{x}'_n-\mathbf{x}_e|^2}$$
$$\cdot\left(\tilde{w}^{\mathrm{d}}_{\mathrm{dir}}(\mathbf{x}_s,\mathbf{x}'_n,\mathbf{x}_e)+\tilde{w}^{\mathrm{s}}_{\mathrm{dir}}(\mathbf{x}_s,\mathbf{x}'_n,\mathbf{x}_e)\right),$$

with $C = \sum_{\mathbf{x}'_n} D(\mathbf{x}'_n)$ the normalization factor. In practice, we instantiate the spatial distribution as a 3D Gaussian $D(\mathbf{x}'_n) = \mathrm{Gauss}(\mathbf{x}'_n|\mu_x,\Sigma_x)$ (Section 4.3), which admits to be evaluated at a deterministic set of points, as it will be detailed in Section 5. Also, we add a small defensive value to the importance weight of each node; this guarantees that the support of our hierarchical PDF is never zero for a non-zero node. Since $\tilde{w}(\mathcal{N},\mathbf{x}_s,\mathbf{x}_e)$ is a spectral (RGB) quantity, due to the colored albedo and emitter radiance, we convert it to luminance when computing the traversal probabilities (Equation (8)).

5. Implementation

We have implemented our technique in Mitsuba 0.6 [Jak10]. In particular, we have extended two integrators: When $\mathbf{x}_s$ is on the sensor (it is the first vertex of the path, possibly sampled for a thin-lens camera) we extend the particle tracer and use our one-more-vertex next-event estimation on the direct connection from the camera (at $\mathbf{x}_s$) towards the emitter (at $\mathbf{x}_e$) through an interaction on the geometry ($\mathbf{x}_n$). This allows using our technique to sample complex high-detail details, such as glints or discrete media.

For the case where $\mathbf{x}_s$ corresponds to a path vertex, we have

Evaluation-time Statistic	Raw Statistic
Spatial	
Area $A_{\mathcal{N}}$	$A_{\mathcal{N}} = \sum A_i$
Mean $\mu_x = s_\mu / A_{\mathcal{N}}$	$s_\mu = \sum A_i \mathbf{x}_i^c$
Covariance $\Sigma_x = (s_\Sigma / A_{\mathcal{N}}) - \mu_x \mu_x^\top$	$s_\Sigma = \sum A_i \mathbf{M}_i$
Directional	
Direction $\mu_n = s_\mathbf{n} / \|s_\mathbf{n}\|$	$s_\mathbf{n} = \sum A_i \hat{\mathbf{n}}_i$
Concentration $\kappa_n = \mathcal{A}_3^{-1}(\|s_n\| / A_{\mathcal{N}})$	
Material	
Albedo $\beta^c_{\mathcal{N}} = s_c / A_{\mathcal{N}}$	$s_c = \sum_i A_i \beta_i^c$
Metallic-ness $\beta^m_{\mathcal{N}} = s_m / A_{\mathcal{N}}$	$s_m = \sum_i A_i \beta_i^m$
Specularity $\beta^s_{\mathcal{N}} = s_s / A_{\mathcal{N}}$	$s_s = \sum_i A_i \beta_i^s$
Microfacet Concentration $\kappa_m = 2A_{\mathcal{N}} / s_\alpha$	$s_\alpha = \sum A_i \alpha_i^2$
Convolved Concentration $\kappa_{conv} = \kappa_n \kappa_m / (\kappa_n + \kappa_m)$	

Table 1: *Summary of the node statistics used in Equation (21) and its computation from the raw statistics from its children. Due to the associative property of the addition, the raw statistics are additively composable (the addition of the children nodes equals the addition of all primitives in both nodes), which allows for constant-time bottom-up computation. For each primitive $\mathcal{P}$, the spatial terms $\mathbf{x}_i^c$ and $\mathbf{M}_i$ are its centroid and uncentered second moment matrix, and $\hat{\mathbf{n}}_i$ is the first moment of its normals. The function $\mathcal{A}_3(\kappa) = \coth\kappa - 1/\kappa$ maps the normalized area-weighted normal sum to the vMF concentration [Jak12].*

extended the path tracer integrator. Vertices $\mathbf{x}_s$ are generated from sampling the BSDF (or the pixel area) from the previous vertex. Similar to traditional next-event estimation, we apply our technique at every vertex $\mathbf{x}_s$ of the path, generating one indirect connection towards the light source.

In all cases $\mathbf{x}_e$ is sampled by first selecting a light source uniformly and then sampling a point uniformly over its surface area. We apply multiple importance sampling (MIS) between the three-vertex paths $\mathbf{x}_s \leftrightarrow \mathbf{x}_n \leftrightarrow \mathbf{x}_e$ generated with our technique, and the same three-vertex paths generated by sampling the BSDF at $\mathbf{x}_s$, tracing a ray that finds $\mathbf{x}_n$ and then applying a deterministic next-event estimation connection from $\mathbf{x}_n$ to $\mathbf{x}_e$.

Hierarchical structure. We use a binary bounding volume hierarchy (BVH) as the structure performing our stochastic top-down traversal. The bounding volume at each node $\mathcal{N}$ is an axis-aligned bounding-box. For each node, we store the aggregated statistics. Note that to avoid very deep trees, we stop the tree construction at a given depth, and thus leaf nodes store several primitives; for leaf nodes, we store the list of primitives that belong to that particular node.

BVH Construction. We build our BVH using three-dimensional spatial subdivision of the primitives, using a standard surface-area heuristic that minimizes the combined bounding volume of children nodes. We build our BVH top to bottom. Note that, even if in our implementation we create a new BVH, we could directly precalculate the node properties in the standard ray-intersection BVH, using it both for intersections and sampling.

After construction, we compute each node's statistics bottom-up by aggregating the statistics of its subnodes (or primitives, in the case of leaf nodes). We initialize leaf nodes by computing raw statistics for each primitive $\mathcal{P}$ contained in the node, and inner nodes by aggregating the statistics from their children subnodes. These raw statistics are additively composable: Their value at any node equals the sum of its children values, so that their aggregation from the children becomes a simple element-wise addition. However, to allow this additive composition, the raw statistics cannot be used directly, and need to be normalized by the node's total area at evaluation time. A summary of the raw statistics computed at build-time, and how they are normalized during evaluation, can be found on Table 1.

Sampling the node importance spatial domain. During rendering, we evaluate the per-node importance (Equation (21)) at points $\mathbf{x}'_n$ from the 3D Gaussian representation of the node $D(\mathbf{x}'_n) = \text{Gauss}(\mathbf{x}'_n | \mu_x, \Sigma_x)$, computed via the unscented transform [JU04], which yields $2d+1 = 7$ sample points: One on the mean of the distribution μ_x and two points along each principal axis of the Gaussian at $\mathbf{x}'_n = \mu_x \pm \sqrt{\lambda_k}\,\mathbf{v}_k$, with λ_k and $\mathbf{v}_k$ the eigenvalues and eigenvectors of Σ_x. When the node subtends a solid angle below a given threshold as seen from $\mathbf{x}_s$ or $\mathbf{x}_e$, we reduce the evaluation to a single point at the mean μ_x, avoiding redundant computation for distant nodes. In our tests, we define this threshold to 0.1 sr (see Section 6.4).

The evaluation points $\mathbf{x}'_n$ and the parameters μ_x, λ_k, $\mathbf{v}_k$ are precomputed per-node at BVH construction, but the evaluation of the node importance weight $\tilde{w}_{\text{dir}}$ is done at render time since it depends on render-time values $\mathbf{x}_s$ and $\mathbf{x}_e$.

Sampling leaf nodes When reaching a leaf node, we resort to importance sampling a primitive within the node proportional to its surface area within the leaf node. This changes the final probability of the sampled point $\mathbf{x}_n$, so that $p(\mathbf{x}_n | \mathcal{N}_{\text{leaf}}, \mathbf{x}_s, \mathbf{x}_e) = A_{\text{leaf}}^{-1}$. This deviates from our gold-standard PDF (3) slightly, but for small-enough primitives the hierarchical PDF computed by traversing the tree is good enough. For that, we first sample a primitive within a leaf node proportionally to their area, and then sample uniformly the surface of the triangle.

To sample a primitive in a leaf node, we use an alias table [KP79] built during the preprocess, which encodes the discrete probability of each primitive within the node, as described by Pharr et al. [PJH23]. This allows sampling in constant time $O(1)$, versus the commonly used search-based discrete sampling. In practice, the alias table uses eight bytes per primitive (one integer and one float), independently of the depth of the BVH. The memory footprint is usually reasonable to allow $O(1)$ sampling within any leaf node. For example, in our most complex scene (Asteroids, with around eight million triangles), this results in under 62 MB of additional memory.

6. Evaluation

6.1. Accuracy of aggregated importance weight

Here we analyze the accuracy of our approximated per-node aggregated importance weight, described in Equation (21), calculated in constant time from each node's statistics, with a ground truth

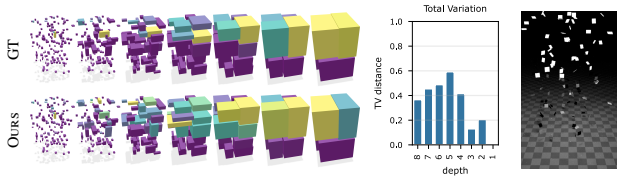


Figure 4: *Visualization of the importance function of Equation (2) (measured for two set query points $\mathbf{x}_s$ and $\mathbf{x}_e$) for different depths of the BVH in a simple scene consisting on a few floating directly lit quads (right).* Left: *visualization of the importance per node of depths 8-2 (1 omitted for space). The top row shows the ground truth function, computed by Monte Carlo integration of a high number of area samples. The bottom row is our approximation, using the aggregated statistics.* Right: *we show the total variation distance of our approximation to the ground truth distribution, computed for each of the depths of the BVH.*

importance weight, computed by Monte Carlo integration of Equation (7) via area samples. We fix the two vertex positions $\mathbf{x}_s$ and $\mathbf{x}_e$ to the same values in both cases. Figure 4 shows the result of this experiment qualitatively and quantitatively by measuring the total variation distance of the two sets of weights at each depth. Our approximation has higher accuracy at greater depths (closest to the leaves), where the nodes contain fewer primitives, so their distributions are simple, and upper tree levels, when the number of primitives per node is high enough that our aggregated statistics capture the general statistics of the scene. However, it is at intermediate depths where our approximation deviates the most. At these depths, the intra-node distributions are harder to fit (being farther from the leaves but not yet low frequency). Still, as we later show, this deviation is not meaningful enough to yield a poor sample distribution in our experiments.

6.2. Ablation study

Our approximated importance weight, described by Equation (21), has several components that take into account different magnitudes that affect the final contribution of the three-vertex path. Its formulation is based on the idea of approximating the ideal importance, Equation (7), as much as possible, while being cheap to compute because it is based on per-node aggregated statistics. In this section, we analyze the relevance of each of these components.

In our first experiment, we render several scenes while deactivating different components of the approximated importance weight. We analyze the convergence of the render with respect to the number of samples per pixel. In this case, we focus on diffuse materials so that the specular component has no effect. The components that we eliminate are represented in the following equation:

$$\tilde{w}(\mathcal{N},\mathbf{x}_s,\mathbf{x}_e) = \frac{1}{C} \overbrace{\sum_{\mathbf{x}'_n}}^{no\_gaussian} \text{Gauss}(\mathbf{x}'_n|\mu_x,\Sigma_x) \cdot \overbrace{\frac{\mathrm{f}_{s\to n'}\cdot\cos\theta_{s,n'}}{|\mathbf{x}_s-\mathbf{x}'_n|^2}}^{no\_bsdf} \cdot \overbrace{\frac{\cos\theta_{e,n'}\cdot S_{e\to n'}}{|\mathbf{x}'_n-\mathbf{x}_e|^2}}^{no\_Le} \cdot \overbrace{\tilde{w}^{\mathrm{d}}_{\mathrm{dir}}(\mathbf{x}_s,\mathbf{x}'_n,\mathbf{x}_e)}^{no\_vmf}, \tag{22}$$

where *no_gaussian* indicates that we only evaluate for a single position (the centroid of the spatial distribution), while *no_bsdf*, *no_Le* and *no_vmf* disable those factors by making them 1. The results of this experiment can be seen in three scenes in Figure 5. Our full model (*ours*) includes all components, and we also include in the ablation the removal of all components except the squared distance to the nodes (*all_ablation*).

As expected, removing all components (*all_ablation*) produces the worst convergence of all the alternatives, while being still better than pure path tracing because it still ensures that an intermediate vertex $\mathbf{x}_n$ is found within the geometry. The experiment also shows that the spatial distribution (*no_gaussian*) is the most relevant component: when omitted the error increases noticeably, indicating that the intra-node geometric variation is essential. Disabling the diffuse component (*no_vmf*) effectively ignores the angular distribution and its effect. It also diminishes the convergence, except on the case of the Veach scene: as the floating facets on top are randomly oriented, the directional distribution of normals is completely uniform, so ignoring it does not provide a practical advantage. The effect of omitting the components from either vertex (*no_Le* and *no_bsdf*) is highly scene-dependent: in some scenes its omission either becomes irrelevant (*no_Le* in AsteroidsFarLight and Veach) or even improves convergence (*no_bsdf* in AsteroidsCloseLight) while performing worse in other setups. In contrast, we find that our full expression for importance weight (*ours*) yields the most robust results across all tested scenarios.

Specular component. Our importance weight also includes a specular term that has been omitted in our previous experiment. Figure 6 shows that in a directly-lit scene with low-roughness quads, the specular approximation helps to guide intermediate samples $\mathbf{x}_n$ towards more important nodes, reducing variance. The specular term is most beneficial for scenes where direct lighting is dominated by sharp highlights; in highly diffuse environments, it provides marginal or negligible benefit. Thus, it could be selectively enabled, though we include it by default for robustness.

6.3. Tree construction and storage cost

We analyze the storage cost of our method in the Asteroids scenes (composed of 8M triangles). Building our hierarchical structure (BVH) takes 2.9s, slightly faster than Mitsuba's kD-tree construction (3.6s). It introduces an storage penalty of around 200MB for a 16-depth BVH, with storage determined by two main factors: Each scene primitive requires an additional 20B for the alias tables and indices necessary for importance sampling and PDF evaluations, while each internal node allocates 296B to store our aggregated spatial, directional, and material statistics (i.e., Gaussian, vMF, surface area, and material properties). Usually, the number of primitives

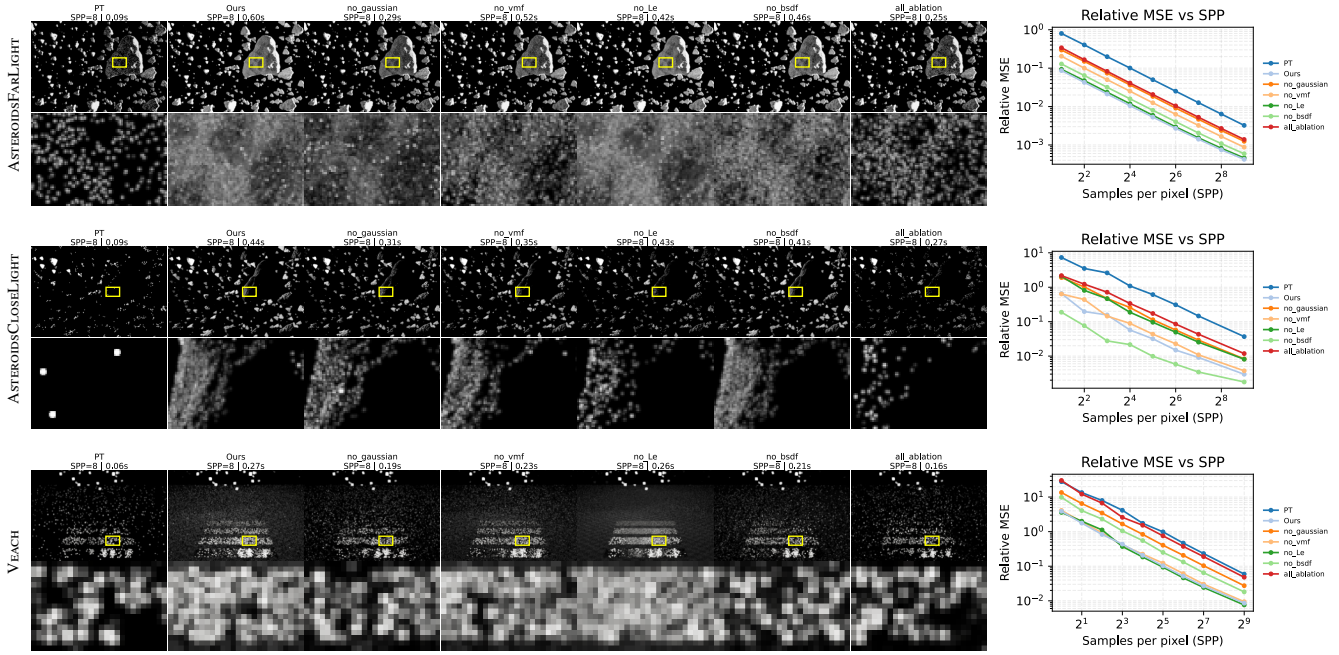


Figure 5: *Ablation study in three different scenes. Some components, such as the spatial distribution (*no_gaussian*) or the VMF integration (*no_vmf*) have a large impact on the correlation of our function to the true PDF (and thus increase variance more when removed). As we can see, in some cases it seems beneficial to ignore some of the factors in our proxy contribution function, such as the intersected BSDF in the AsteroidsCloseLight scene (middle), or the emitter in the Veach scene (bottom). Our full function is however more general, being always one of the two best performing ones.*

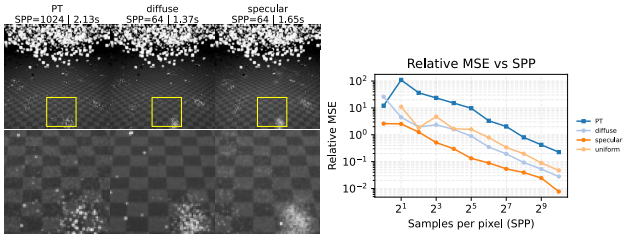


Figure 6: Left: *Roughly equal time (2s) comparison between using only the diffuse (diffuse) or also the specular component (specular) in our proxy contribution function in a scene with specular quads of low roughness ($\alpha = 0.01$).* Right: *convergence plot of the techniques in this scene. The specular component adds some computational overhead, but increases convergence in this type of scenes.*

dominates the node count, with a 16-depth BVH containing at most 131k nodes, and therefore the total memory overhead scales efficiently ($8\,\mathrm{M} \times 20\,\mathrm{B} + 131\,\mathrm{k} \times 296\,\mathrm{B} = 198\,\mathrm{MB}$), making it practical for modern rendering systems.

6.4. Parameter sensitivity

Our sampling technique only depends on two parameters that affect the trade-off between accuracy and efficiency. The first one is the number of primitives per leaf node, which is directly related to the depth of the BVH. In Figure 7 we analyze the convergence w.r.t. the depth of the BVH (between 10 and 23 levels) on the AsteroidsFarLight, which has 7.8 million triangles and would require 23 BVH levels to obtain single-triangle leaf nodes. Smaller leaf nodes (larger depths) show a faster convergence w.r.t. spp, indicating a better accuracy on our importance weight. However, larger depths also involve more time for the traversal of the BVH. The analysis shows that the optimal depth for this scene is around 16, which involves several hundreds of primitives per leaf node.

We also analyze the solid angle threshold for which we only evaluate the importance of a single point (centroid) of the spatial distribution, which also presents a trade-off between accuracy and efficiency. A threshold of 0.1 steradians improves performance with negligible variance increase, which indicates that the spatial variation of far nodes has little effect. Thresholds above one steradian significantly degrade performance, which is consistent with our ablation analysis (Section 6.2).

6.5. Applicability

Our technique targets scenes with sparse geometry, incurring an extra sampling cost amortized by the drastic reduction in variance for these types of scenes. We explore our method's performance with respect to solid angle density of the geometry, as well as the number of triangles, to gain insights into the limits where our method is applicable. In Figure 8, we approximate a hemispherical dome by subdividing a half icosahedron between 10 to 100 times (1k to 100k triangles) and randomly removing triangles to keep a varying *area*

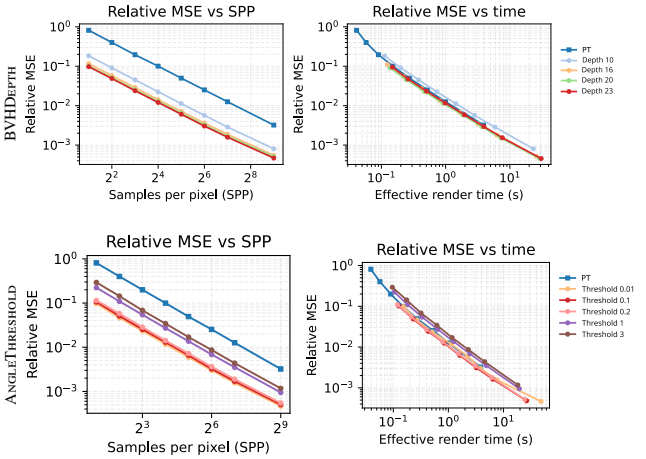


Figure 7: *Variance and time tradeoff of two parameters, measured in the* AsteroidsFarLight *scene.* Top*: maximum depth of the BVH. Increasing it reduces variance, as shown in the top left, but with diminishing returns when the BVH has enough resolution for the geometry of the scene. Adding further levels only increases the cost (top right).* Bottom*: the gaussian solid angle threshold delimits the solid angle to treat the gaussian of the node as a point. A threshold of around* 0.1 *sr (red) improves performance for a similar variance to the baseline (*0 *sr, orange); thresholds above one sr (purple) significantly degrade performance.*

ratio of the covered hemisphere (0.01 to 0.9). We collocate the measurement point and the light source $\mathbf{x_s} = \mathbf{x_l}$ in the center of the hemisphere, where the light source is a wide spotlight that only illuminates the half icosahedron. We measure the irradiance at $\mathbf{x_s}$ both by our method (10 spp) and path tracing at equal sample counts and rendering times.

As the sparsity decreases (*area ratio* increases), the probability of conventional directional sampling hitting a triangle grows. This makes path tracing improve for denser regimes, being the worst in intermediate sparse regimes (around 0.05 to 0.1) due to sporadic intersections. Reversely, in denser environments, our diffuse and spatial statistical approximations deviate more from the target PDF, increasing the error for the same number of samples. This makes our method no longer competitive at equal time beyond an *area ratio* of 0.25. The total number of geometric primitives has a negligible effect on our method's error, as both techniques scale similarly with the number of subdivisions.

7. Results

7.1. Comparison

We evaluate our technique on two variants of the Asteroids scene, which contains approximately eight million diffuse triangles of high albedo, sparsely distributed to create challenging indirect lighting scenarios. We compare our full method against path tracing (PT), bidirectional path tracing (BDPT), and practical path guiding (PPG) [MGN17].

When illuminated by a distant light source (Figure 9), the light's solid angle is effectively uniform across the scene. In this case, standard path tracing fails in the indirectly illuminated parts because using material sampling at the vertex path rarely intersects the sparsely lit geometry, since high-importance directions have small solid angles relative to the full hemisphere. Practical Path Guiding inherits this weakness, relying on path tracing for incoming radiance exploration, and delays the discovery of the few lit surfaces. Bidirectional Path Tracing also struggles as the sparse geometry is also difficult to find from the light paths sampled from the distant light. Our technique converges faster since it is able to both sample an intermediate vertex on the geometry, and guide them towards high-throughput regions without having to explore the full scene. This advantage is clearest at low sample counts.

When the light source is positioned close to the scene, directly illuminating only a subset of asteroids (Figure 10), both Path Tracing and Practical Path Guiding still suffer since the lit geometry is still difficult to find. However, now Bidirectional Path Tracing behaves better since the light emission profile is more concentrated and is able to find the sparse geometry more easily. Even with that, our method remains competitive, as our hierarchical BVH traversal directly targets high-albedo regions without waiting for path-space sampling. This is again especially visible at low sample counts. As sample count increases, bidirectional path tracing catches up and our relative advantage diminishes.

7.2. Discrete media

We evaluate our technique also for direct light, where the vertex $\mathbf{x}_s$ is on the sensor. As stated in Section 5, in this case we resort to a particle tracer integrator. We render sparse subpixel-sized particles positioned over all the scene and directly illuminated by the spotlight of a lighthouse (Lighthouse scene). This scene creates an extreme sparsity problem for path tracing, since each camera ray has a very low probability of intersecting individual particles, making discovery through camera ray tracing prohibitively expensive. In Figure 11 we show an equal time comparison of the path tracer and our method in this scene, rendered with a perspective sensor. The path tracer is unable to find most snowflakes. With a thin-lens camera (Figure 1), the difference becomes dramatic, since the possibility of finding the paths is even smaller and path tracing remains noisy at 100k spp (36 minutes), while our method converges significantly faster (2k spp, 105 s).

8. Discussion

In this paper, we have presented a sampling technique, that we name one-more-vertex next-event estimation. This technique connects a path vertex to a light source through an indirect interaction in a new sampled vertex. The position of this vertex is sampled from the whole geometry of the scene by exploring a hierarchical structure (a BVH). This is particularly beneficial for connections in which the projected solid angle of the geometry towards both vertices (path and light source) is small, which is the case in scenes which are dominated by sparse geometry, and therefore hard to find from BSDF (or emitter) sampling. In those types of scenes, we have shown clear faster convergence than other algorithms. When the path vertex corresponds to the sensor itself, our sampling technique provides efficient glint rendering.

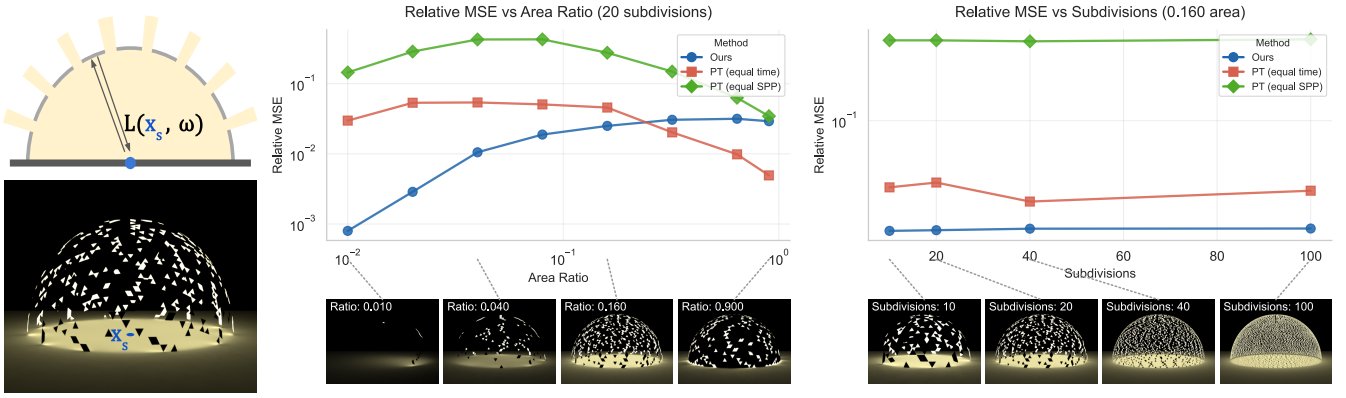


Figure 8: *Analysis of our technique as a function of the sparsity of the scene. We use an hemispherical dome around the light and sensor points* $\mathbf{x}_l = \mathbf{x}_s$*, tessellated with increasing subdivisions, and varying the sparsity as a function of the area ratio by discarding triangles.* Left*: 2D schematic and render of the setup, where measurement point and the light source are collocated at* $\mathbf{x_s}$ *in the center of the hemisphere. We measure the irradiance reaching* $\mathbf{x_s}$ *computed using our method (blue) and path tracing (PT) at equal sample counts (green) and rendering times (orange).* Middle: *Our technique (blue) reduces relative MSE at both equal samples (green) and equal time (orange) in sparse environments, and it is always better than PT in term of sample count, though as the sparseness decreases (area ratio* > 0.25*) the cost of our technique does not pay off, since PT is able to find contributing samples.* Right: *The relative error is independent of the number of triangles, which increases cost in a similar way for standard PT and ours.*

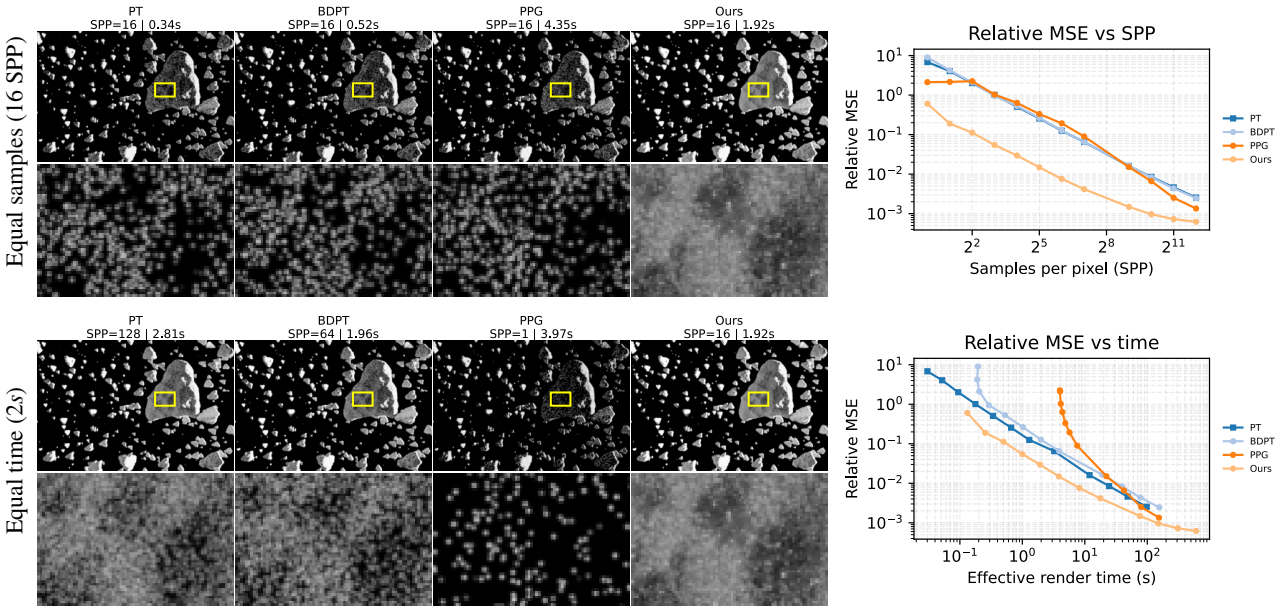


Figure 9: *AsteroidsFarLight comparison. Qualitative results and relMSE convergence for equal samples (top) and roughly equal time (bottom). The convergence of PT and BDPT are nearly identical, since it is equally unlikely for light tracing and material sampling to find a high-importance lit surface. Practical Path Guiding converges quickly in the long term, but still needs many samples to train its guiding structure. Our method finds length-2 connections even at low sample counts (top).*

Performance Overhead Our method incurs a one-time precomputation cost (BVH construction and statistics aggregation, comparable to the construction of the ray-intersection kD-Tree of Mitsuba), as discussed in Section 6.3, and a per-sample sampling cost. The per-sample cost is governed by the stochastic BVH traversal: at each of the D levels, the importance weight $\tilde{w}$ is evaluated for both child nodes, yielding $2D$ weight evaluations per sample. For a BVH over N primitives, the depth is $D = \lceil log_2(N) \rceil$, unless it is bounded (Section 6.4). When MIS is enabled, a second full traversal is required to evaluate the PDF of the BSDF-sampled path, so the total per-sample cost becomes $t(D) = 2 \cdot D \cdot 2 \cdot t_w = 4Dt_w$, where t_w is the time to evaluate a single node's importance weight. In our Asteroids scene

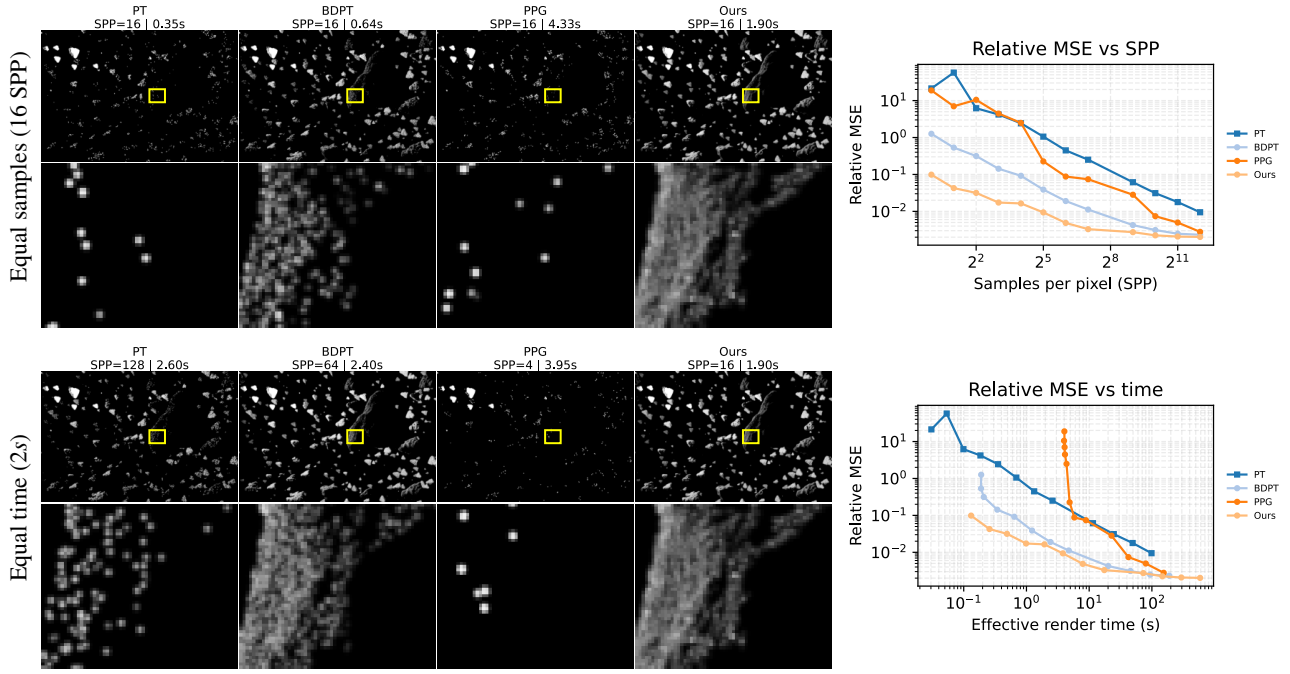


Figure 10: *AsteroidsCloseLight is a variation of the same asteroids scene with the light source in the middle, illuminating only a few of the asteroids. This scene is much more challenging for the path tracer, and Practical Path Guiding takes more time to fit its distribution. Light tracing techniques like BDPT converge quickly, since the paths started at the light source are very likely to find a high-throughput surface. Our technique still converges faster. Rows show qualitative results and relMSE convergence for equal samples (top) and roughly equal time (bottom).*

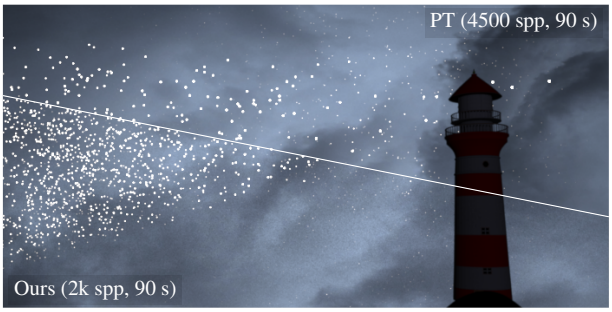
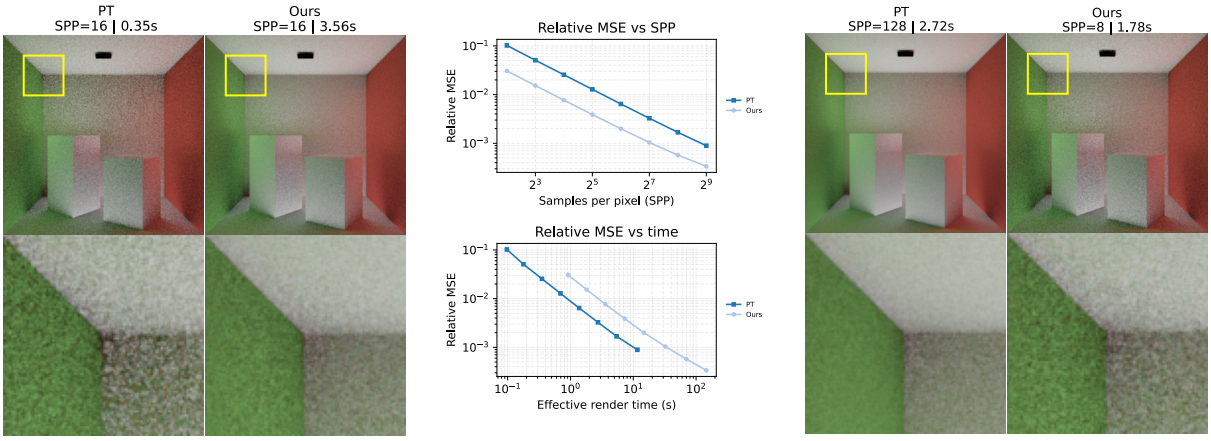


Figure 11: *Lighthouse scene, featuring a strong spotlight illuminating a snow storm. With our particle tracer integrator, our technique effectively guides the light paths to the snowflakes, and connects the camera. With a perspective sensor, this scene is challenging for a path tracer, which is usually unable to find most illuminated snowflakes. Our technique is able to find most of them in equal time.*

($D \approx 20$), this yields a per-sample cost approximately twice that of standard path tracing, as empirically explored in Figure 7. Nonetheless, this overhead is outweighed by the drastic variance reduction in sparse scenes. Performance could be further optimized by reducing the frequency of weight evaluations, for instance by applying heuristics to skip subtrees similar to Lightcuts [WFA*05], or by utilizing cheaper importance weight evaluations through resampling. Notably, this has already been done for many-lights hierarchical sampling [CLH24], a field from which we draw inspiration.

Dense scenes. Our method targets scenarios where conventional path construction struggles to discover the high-throughput paths that come from small solid angle regions related to sparse small geometry. The associated performance overhead is justified, as these paths are difficult to sample otherwise. However, this overhead becomes unjustifiable in dense scenes, which are not the main focus of our approach, where traditional algorithms already perform well. In addition to the analysis in Section 6.5, we show this trade-off for indirect illumination in a Cornell Box (Figure 12). The geometry is not sparse enough to benefit from our guided sampling, and the increased per-sample cost means we perform worse in wall-clock time. In such scenes, geometry is readily discovered through material sampling, and the cost of our method outweighs its benefit. Still, we achieve lower variance per sample than path tracing for indirect light, since we explore indirect paths better (although at increased cost). This suggests a practical hybrid approach: detect geometric sparsity heuristically (e.g., via BVH statistics), and activate our technique selectively. We leave this optimization to future work.



(a) *Equal samples (16 SPP)* **(b)** *Convergence* **(c)** *Equal time (∼3s)*

Figure 12: *Comparison of two-bounce indirect illumination in the Cornell Box, where walls are subdivided into 100 quadrilateral primitives each. We show faster convergence w.r.t. the number of samples per pixel, even if the geometry is very dense. However, the temporal cost of the traversal does not compensate this increase in convergence. Left and Right: Qualitative comparison at equal samples and approximately equal time, respectively. Middle top: Relative MSE with respect to samples per pixel, the 8 spp mark corresponds to the renders to the left. Middle bottom: Relative MSE with respect to render time.*

Specular connections. We have focused our results on scenes where the challenge of the connection comes from the sparsity of the geometry. Nevertheless, there are other types of challenging three-vertex connections, such as those in which the connection occurs on a smooth specular surface. Our expression for importance weight accounts for specular contributions, as we show in Figure 6, so our technique could be applied to those as well. However, in our current implementation, the BVH is constructed from the spatial dimensions, which works optimally for minimizing the spatial spread of each node, but yields larger angular distributions which are suboptimal for detecting very narrow specular peaks at lower tree depths. An interesting avenue for future work would be to explore other hierarchical constructions that also account for the angular dimensions of the distribution of geometry, so that narrow specular connections would be more easily found.

Bidirectional extensions. In our approach, one of the two vertices from which we sample the intermediate one always belongs to a light source. However, our technique could also be applied with a vertex on the light path of a bidirectional technique, such as bidirectional path tracing or virtual point lights, enabling complex bidirectional connections that would allow, for instance, the rendering of glints generated by caustics interacting with sparse geometry.

Alternative importance weight expressions. Our proposed node importance weight, described by Equation (21), is an approximation of the expected contribution within the node, as described by Equation (7). Our methodology would also be practical in other contexts. By changing our target probability distribution to a different domain and devising the adequate per-node statistics to approximate it, we would provide new geometry-based importance sampling routines for challenging connections. This could be useful in the context of transient rendering, where optimal sampling routines along the temporal domain often require searching for vertices in the geometry [PVG19; RGMJ22]. Also, in differentiable rendering, by following our approach, we could target not the most important connections but the connections that provide a larger effect on the derivatives, accelerating convergence in that domain. We hope this inspires future research.

Acknowledgements

Special thanks to the reviewers for their constructive insight. This work has received funding from the European Commission's HORIZON EUROPE Research and Innovation Actions project Sestosenso under GA number 101070310. Jorge Garcia-Pueyo was supported by the FPU23/03132 predoctoral grant and Nestor Monzon by a Gobierno de Aragon predoctoral grant (2023-2027). Polyscope [Sha*19] was used for several debug visualizations, such as Figure 4.

References

[BNJ15] Bitterli, Benedikt, Novák, Jan, and Jarosz, Wojciech. "Portal-masked environment map sampling". *Computer Graphics Forum*. Vol. 34. 4. 2015, 13–19 3.

[BW12] Burley, Brent and Walt Disney Animation Studios. "Physically-based shading at Disney". *Proceedings of SIGGRAPH '12*. 2012, 1–7 6.

[CK18] Conty-Estevez, Alejandro and Kulla, Christopher. "Importance Sampling of Many Lights with Adaptive Tree Splitting". *Proc. ACM Comput. Graph. Interact. Tech.* 1.2 (2018) 3, 4.

[CLH24] Conty, Alejandro, Lecocq, Pascal, and Hellmuth, Chris. "A Resampled Tree for Many Lights Rendering". *SIGGRAPH '24 Talks*. ACM, 2024 12.

[Der22] Derevyannykh, Mikhail. "Real-Time Path-Guiding Based on Parametric Mixture Models". *Proceedings of Eurographics '22*. 2022 3.

[DHC*21] Deng, Xi, Hašan, Miloš, Carr, Nathan, et al. "Path graphs: iterative path space filtering". *ACM Trans. Graph.* 40.6 (2021) 3.

[DLW*22] Deng, Hong, Liu, Yang, Wang, Beibei, et al. "Constant-cost spatio-angular prefiltering of glinty appearance using tensor decomposition". *ACM Trans. Graph.* 41.2 (2022), 1–17 3.

[DPÖM22] Dodik, Ana, Papas, Marios, Öztireli, Cengiz, and Müller, Thomas. "Path Guiding Using Spatio-Directional Mixture Models". *Computer Graphics Forum*. Vol. 41. 1. 2022, 172–189 3.

[FWW*22] Fan, Jiahui, Wang, Beibei, Wu, Wenshi, et al. "Efficient specular glints rendering with differentiable regularization". *IEEE Transactions on Visualization and Computer Graphics* 29.6 (2022), 2940–2949 3.

[GCGP18] Guo, Jie, Chen, Yanjun, Guo, Yanwen, and Pan, Jingui. "A Physically-based Appearance Model for Special Effect Pigments". *Computer Graphics Forum*. Vol. 37. 4. 2018, 67–76 3.

[GH24] Granizo-Hidalgo, Ana and Holzschuch, Nicolas. "Computing Manifold Next-Event Estimation without Derivatives using the Nelder-Mead Method". *Proceedings of EGSR '24*. 2024, 1–9 2.

[GHC*22] Guo, Jie, Hu, Bingyang, Chen, Yanjun, et al. "Rendering discrete participating media using geometrical optics approximation". *Computational Visual Media* 8.3 (2022), 425–444 3.

[GKH*13] Georgiev, Iliyan, Křivánek, Jaroslav, Hachisuka, Toshiya, et al. "Joint importance sampling of low-order volumetric scattering". *ACM Trans. Graph.* 32.6 (2013) 2.

[HDF15] Hanika, Johannes, Droske, Marc, and Fascione, Luca. "Manifold Next Event Estimation". *Computer Graphics Forum* 34.4 (2015), 87–97 2.

[Hol15] Holzschuch, N. "Accurate Computation of Single Scattering in Participating Media with Refractive Boundaries". *Computer Graphics Forum* 34.6 (2015), 48–59 2.

[HWD22] Hanika, Johannes, Weidlich, Andrea, and Droske, Marc. "Once-more scattered next event estimation for volume rendering". *Computer Graphics Forum* 41.4 (2022), 17–28 2.

[IDN12] Iwasaki, Kei, Dobashi, Yoshinori, and Nishita, Tomoyuki. "Interactive bi-scale editing of highly glossy materials". *ACM Trans. Graph.* 31.6 (2012) 6.

[Jak10] Jakob, Wenzel. *Mitsuba Renderer 0.6*. 2010. URL: http://www.mitsuba-renderer.org/index_old.html 6.

[Jak12] Jakob, Wenzel. "Numerically stable sampling of the von Mises-Fisher distribution on S^2 (and other tricks)". *Interactive Geometry Lab, ETH Zürich, Tech. Rep* 6 (2012) 7.

[JHY*14] Jakob, Wenzel, Hašan, Miloš, Yan, Ling-Qi, et al. "Discrete stochastic microfacet models". *ACM Trans. Graph.* 33.4 (2014), 1–10 3.

[JMM*14] Jarabo, Adrian, Marco, Julio, Munoz, Adolfo, et al. "A framework for transient rendering". *ACM Trans. Graph.* 33.6 (2014), 1–10 2.

[JU04] Julier, S.J. and Uhlmann, J.K. "Unscented filtering and nonlinear estimation". *Proceedings of the IEEE* 92.3 (2004), 401–422 7.

[Kal63] Kalos, M. H. "On the Estimation of Flux at a Point by Monte Carlo". *Nuclear Science and Engineering* 16.1 (1963), 111–117 2.

[KC77] Kalli, H.J. and Cashwell, E.D. "Evaluation of three Monte Carlo estimation schemes for flux at a point". (1977) 2.

[KF11] Kulla, Christopher and Fajardo, Marcos. "Importance sampling of area lights in participating media". *SIGGRAPH '11 Talks*. ACM, 2011 2.

[KNK*16] Koerner, David, Novák, Jan, Kutz, Peter, et al. "Subdivision Next-Event Estimation for Path-Traced Subsurface Scattering". *Proceedings of EGSR '16*. The Eurographics Association, 2016 2.

[KP79] Kronmal, Richard A. and Peterson, Arthur V. "On the Alias Method for Generating Random Variables From a Discrete Distribution". *The American Statistician* 33 (1979), 214–218 7.

[LADL18] Li, Tzu-Mao, Aittala, Miika, Durand, Frédo, and Lehtinen, Jaakko. "Differentiable Monte Carlo ray tracing through edge sampling". *ACM Trans. Graph.* 37.6 (2018) 3.

[LCHL24] Lu, Haolin, Chang, Wesley, Hedstrom, Trevor, and Li, Tzu-Mao. "Real-Time Path Guiding Using Bounding Voxel Sampling". *ACM Trans. Graph.* 43.4 (2024) 3.

[LXY19] Liu, Yifan, Xu, Kun, and Yan, Ling-Qi. "Adaptive BRDF-Oriented Multiple Importance Sampling of Many Lights". *Computer Graphics Forum* 38.4 (2019), 123–133 3.

[LY20] Lin, Daqi and Yuksel, Cem. "Real-Time Stochastic Lightcuts". *Proc. ACM Comput. Graph. Interact. Tech.* 3.1 (2020) 3, 4.

[LZHJ20] Loubet, Guillaume, Zeltner, Tizian, Holzschuch, Nicolas, and Jakob, Wenzel. "Slope-space integrals for specular next event estimation". *ACM Trans. Graph.* 39.6 (2020) 2.

[MGN17] Müller, Thomas, Gross, Markus, and Novák, Jan. "Practical Path Guiding for Efficient Light-Transport Simulation". *Computer Graphics Forum* 36.4 (2017), 91–100 2, 3, 10.

[MMR*19] Müller, Thomas, McWilliams, Brian, Rousselle, Fabrice, et al. "Neural importance sampling". *ACM Trans. Graph.* 38.5 (2019), 1–19 3.

[MPC19] Moreau, P., Pharr, M., and Clarberg, P. "Dynamic many-light sampling for real-time ray tracing". *Proceedings of HPG '19*. Eurographics Association, 2019, 21–26 3.

[MPG*16] Müller, Thomas, Papas, Marios, Gross, Markus, et al. "Efficient Rendering of Heterogeneous Polydisperse Granular Media". *ACM Trans. Graph.* 35.6 (2016), 168:1–168:14 3.

[MPH*15] Meng, Johannes, Papas, Marios, Habel, Ralf, et al. "Multi-Scale Modeling and Rendering of Granular Materials". *ACM Trans. Graph.* 34.4 (2015) 3.

[MWM07] Moon, Jonathan T., Walter, Bruce, and Marschner, Stephen R. "Rendering discrete random media using precomputed scattering solutions". *Proceedings of EGSR '07*. Eurographics Association, 2007, 231–242 3.

[PJH23] Pharr, Matt, Jakob, Wenzel, and Humphreys, Greg. *Physically based rendering: From theory to implementation*. MIT Press, 2023 7.

[PVG19] Pediredla, Adithya, Veeraraghavan, Ashok, and Gkioulekas, Ioannis. "Ellipsoidal path connections for time-gated rendering". *ACM Trans. Graph.* 38.4 (2019) 2, 13.

[RGMJ22] Royo, Diego, García, Jorge, Muñoz, Adolfo, and Jarabo, Adrian. "Non-line-of-sight transient rendering". *Computers & Graphics* 107 (2022), 84–92 13.

[RMW*25] Rath, Alexander, Manzi, Marco, Weiss, Sebastian, et al. "Neural Resampling with Optimized Candidate Allocation". (2025) 3.

[Sch94] Schlick, Christophe. "An Inexpensive BRDF Model for Physically-based Rendering". *Computer Graphics Forum* 13.3 (1994), 233–246 6.

[Sha*19] Sharp, Nicholas et al. *Polyscope*. www.polyscope.run. 2019 13.

[SHD24] Schüßler, Vincent, Hanika, Johannes, and Dachsbacher, Carsten. "Bridge Sampling for Connections via Multiple Scattering Events". *Computer Graphics Forum* 43.4 (2024), e15160 2.

[SLY*24] Su, Fujia, Li, Bingxuan, Yin, Qingyang, et al. "Proxy Tracing: Unbiased Reciprocal Estimation for Optimized Sampling in BDPT". *ACM Trans. Graph.* 43.4 (2024) 2.

[TH19] Tokuyoshi, Yusuke and Harada, Takahiro. "Hierarchical russian roulette for vertex connections". *ACM Trans. Graph.* 38.4 (2019) 3.

[VKŠ*14] Vorba, Jiří, Karlík, Ondřej, Šik, Martin, et al. "On-line learning of parametric mixture models for light transport simulation". *ACM Trans. Graph.* 33.4 (2014) 3.

[WFA*05] Walter, Bruce, Fernandez, Sebastian, Arbree, Adam, et al. "Lightcuts: a scalable approach to illumination". *ACM Trans. Graph.* 24.3 (2005), 1098–1107 3, 4, 12.

[WHD17] Weber, Pascal, Hanika, Johannes, and Dachsbacher, Carsten. "Multiple Vertex Next Event Estimation for Lighting in dense, forward-scattering Media". *Computer Graphics Forum* 36.2 (2017), 21–30 2.

[WRG*09] Wang, Jiaping, Ren, Peiran, Gong, Minmin, et al. "All-frequency rendering of dynamic, spatially-varying reflectance". *ACM Trans. Graph.* 28.5 (2009), 1–10 6.

[WWH18] Wang, Beibei, Wang, Lu, and Holzschuch, Nicolas. "Fast global illumination with discrete stochastic microfacets using a filterable model". *Computer Graphics Forum*. Vol. 37. 7. 2018, 55–64 3.

[WZHB09] Walter, Bruce, Zhao, Shuang, Holzschuch, Nicolas, and Bala, Kavita. "Single scattering in refractive media with triangle mesh boundaries". *ACM Trans. Graph.* 28.3 (2009) 2.

[YHJ*14] Yan, Ling-Qi, Hašan, Miloš, Jakob, Wenzel, et al. "Rendering glints on high-resolution normal-mapped specular surfaces". *ACM Trans. Graph.* 33.4 (2014), 1–9 3.

[YHMR16] Yan, Ling-Qi, Hašan, Miloš, Marschner, Steve, and Ramamoorthi, Ravi. "Position-normal distributions for efficient rendering of specular microstructure". *ACM Trans. Graph.* 35.4 (2016), 1–9 3.

[ZGJ20] Zeltner, Tizian, Georgiev, Iliyan, and Jakob, Wenzel. "Specular Manifold Sampling for Rendering High-Frequency Caustics and Glints". *ACM Trans. Graph.* 39.4 (2020) 2.

[ZHR*25] Zhou, Yang, Huang, Tao, Ramamoorthi, Ravi, et al. "Appearance-Preserving Scene Aggregation for Level-of-Detail Rendering". *ACM Trans. Graph.* 44.1 (2025) 5.

[ZMY*20] Zhang, Cheng, Miller, Bailey, Yan, Kai, et al. "Path-space differentiable rendering". *ACM Trans. Graph.* 39.4 (2020) 3.

[ZWWH25] Zhao, Sizhe, Wang, Beibei, Wang, Lu, and Holzschuch, Nicolas. "A discrete microfacet model for transparent glints rendering". *Computational Visual Media* (2025) 3.

[ZZX*22] Zhu, Junqiu, Zhao, Sizhe, Xu, Yanning, et al. "Recent advances in glinty appearance rendering". *Computational Visual Media* 8.4 (2022), 535–552 3.