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

Illumination-Aware Spatial Subdivision for Path Guiding

F. Zheng[1] C. Peters[1] S. Herholz[2] M. Manzi[3] E. Eisemann[1]

[1]Delft University of Technology, The Netherlands
[2]Blender Institute, The Netherlands
[3]DisneyResearch|Studios, Switzerland

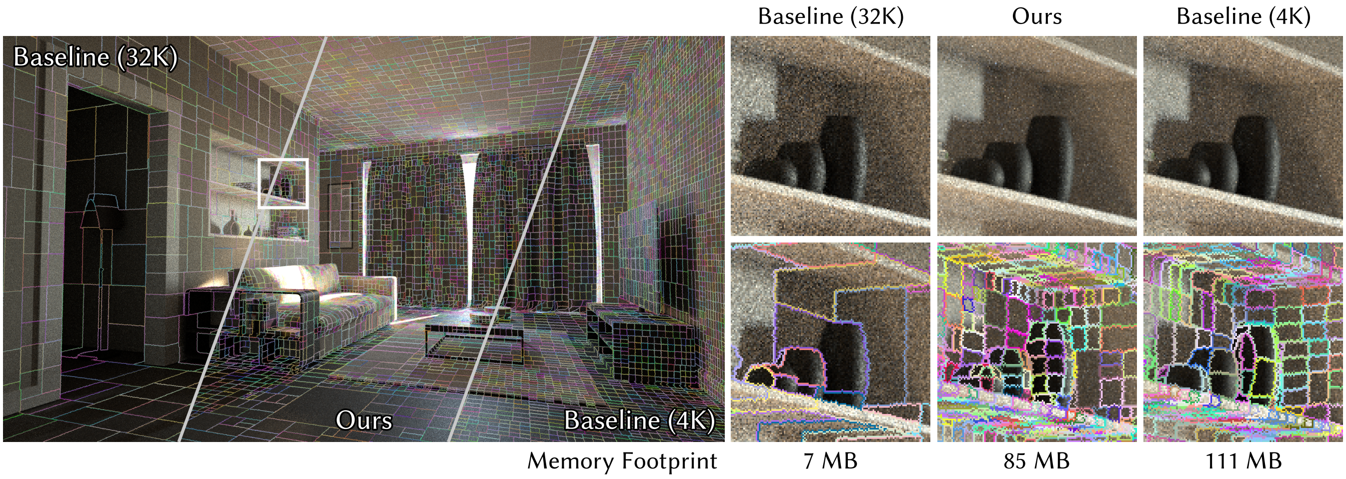


Figure 1: *Compared to traditional sample-count-based subdivision strategies, which lead to a uniform spatial subdivision (**left** and **right**), our illumination-aware strategy (**center**) adapts to changes in the incoming illumination: Baseline* 32K *(splitting on 32K samples, MRAE=0.240), Ours (MRAE=0.164), and Baseline* 4K *(MRAE=0.179). The shown images are* 256 *SPP renderings where the guiding structure is trained during the first* 128 *SPP. Numbers below the insets show the memory footprint of the entire guiding structure generated by the method.*

Abstract

Path tracing is ubiquitous in production rendering and path guiding has established itself as a powerful approach to mitigate its failure cases. Several widely used methods partition the scene using a k-d tree and store a directional guiding distribution per cell for importance sampling. While a lot of research has improved the guiding distributions, the decision when to split the k-d tree still relies on a simple sample count threshold. We propose a method to adapt the k-d tree depending on the variation in the illumination. To this end, we use lookahead cells, i.e. multiple additional levels of k-d tree cells that do not store a guiding distribution. Instead, they store compact characterizations of the light field, which we call signatures. Specifically, we use the mean radiance and radiance-weighted mean direction. We model the uncertainty in these signatures probabilistically to derive split criteria that split k-d tree cells when we are confident that one of their lookahead cells differs substantially. As a result, we make existing guiding methods allocate computational and storage resources more efficiently, using small cells in regions with rapidly varying illumination whilst sharing data for uniformly lit regions.

CCS Concepts

• ***Computing methodologies*** → ***Ray tracing;***

---

1. Introduction

Path tracing is the prevalent approach for physically-based rendering in production. However, there are many light transport phenomena, such as caustics or strong indirect lighting contributions through long paths, that a basic, unidirectional path tracer cannot sample well. Path guiding ameliorates these problems: It progressively learns guiding distributions from samples of a path tracer and uses them for importance sampling. In this way, it improves the sampling of difficult paths without fundamental changes to the architecture of the renderer. Consequently, it has established itself as a widely used method in production rendering [HSRM25].

Many existing methods [MGN17; HZE*19; RHL20; RGH*20; DPÖM22; RGH*22] rely on a spatial subdivision, e.g. using a k-d tree, and train a directional guiding distribution per cell. Prior work has focused on improving the guiding distributions and their training, with little attention to optimal spatial subdivision. The most widely used methods for spatial subdivision for path guiding simply split cells of the k-d tree whenever the number of samples available to train this cell surpasses a user-defined threshold.

As a result, regions with uniform lighting may have excessive subdivision, even though they could share a common guiding distribution. On the other hand, regions with rapid illumination changes may have too few cells, such that the guiding distributions cannot provide a good fit for the entire spatial extent of their cells. In general, smaller cells are beneficial to the quality of path guiding, but beyond a certain scene-dependent point, high-contribution samples in the cells will become too rare to learn good guiding distributions. Then these cells will have much higher variance than neighboring cells, which is an obvious artifact (Fig. 5). Therefore, the choice of the sample count threshold is challenging. Too high values make the guiding ineffective, too low values harm its robustness. Different parts of the same scene may call for different values and the best choice also depends on the rendering resolution.

To overcome these problems, we propose a more sophisticated subdivision approach that takes local changes of illumination into account (Fig. 1). For ease of integration, we stick to the established k-d tree structure, but change the splitting criterion for the cells (Sec. 3). To this end, we use multiple levels of lookahead cells: k-d tree cells that are not used for guiding yet, but store lightweight statistics about available samples (Sec. 3.1). By comparing lookahead cells between different levels, we detect whether the illumination in a smaller cell differs enough to justify a split decision.

Features that are difficult to sample, such as caustics, may cover only a small portion of a cell. To achieve sufficient spatial sensitivity, we use six levels of lookaheads. Therefore, their memory footprint has to be small and the update of their statistics must be efficient. After extensive experiments, we settled on the mean incoming radiance from all directions (Sec. 4), and the radiance-weighted mean direction (Sec. 5). Based on the available samples, we produce noisy estimates of these quantities and explicitly model their uncertainty (Sec. 4.1 and 5.2). On this basis, we derive split criteria that ensure high confidence in the split decisions (Sec. 4.2 and 5.3).

Our results (Sec. 6) show that this strategy reliably adapts the subdivision of the k-d tree to features in the illumination. Regions with rapidly changing illumination are quickly subdivided to a sufficiently high spatial resolution, whereas uniformly lit regions reuse guiding distributions across large areas. The subdivisions are less dependent on the rendering resolution than subdivisions based on sample count thresholds and adapt to characteristics of the integrator, such as whether or not next-event estimation is used. Our results also indicate negligible run time overhead and a moderate memory cost for lookahead cells that is further reduced by having less subdivision in uniformly-lit regions.

2. Background and Related Work

We now provide background on path tracing and discuss related work on path guiding, including the method that we use to define the potential k-d tree splits in our method. Additionally, we discuss the related problem of radiance caching.

2.1. Path Tracing

Path tracing [KFF*15] renders digital scenes realistically by solving the light transport equation (LTE) [Kaj86]:

$$L_o(\mathbf{x},\omega_o) = L_e(\mathbf{x},\omega_o) + \int_{\mathbb{S}^2} f_s(\mathbf{x},\omega_i,\omega_o) L_i(\mathbf{x},\omega_i) |\omega_i \cdot \mathbf{n}| \, d\omega_i \quad (1)$$

where $\mathbb{S}^2$ is the unit sphere, L_o and L_e are outgoing and emitted radiance at a surface point $\mathbf{x}$ into direction $\omega_o \in \mathbb{S}^2$, respectively, and f_s is the bidirectional scattering distribution function (BSDF).

The outgoing radiance is computed using Monte Carlo integration by sampling a random incoming direction and estimating its radiance recursively:

$$\langle L_o(\mathbf{x},\omega_o)\rangle = L_e(\mathbf{x},\omega_o) + \frac{f_s(\mathbf{x},\omega_i,\omega_o)\,\langle L_i(\mathbf{x},\omega_i)\rangle\,|\omega_i \cdot \mathbf{n}|}{p(\omega_i)} \quad (2)$$

where $p(\omega)$ is the probability density function (PDF) of the sampled direction ω_i. Path tracing generates $N \in \mathbb{N}$ paths per pixel in this manner and variance decreases in proportion to $\frac{1}{N}$.

Importance sampling strategies reduce the variance by using densities $p(\omega)$ that approximate part of the integrand in Eq. 1. For example, importance sampling of the visible normal distribution function accounts for the BSDF [Hd14] and light sampling/next-event estimation account for direct illumination from area lights [Pet21]. These strategies focus on aspects of the integrand that are known a priori. Multiple strategies can be combined using multiple importance sampling (MIS) [VG95]. Resampled importance sampling (RIS) [Tal05] samples densities approximately where exact importance sampling is not viable.

2.2. Path Guiding

Path guiding is an *adaptive* importance sampling method that adapts the sampling distributions to data observed during rendering. Typically, the goal is to approximate the distribution of the 7D product (3D from $\mathbf{x}$, 2D from ω_i, and 2D from ω_o) in Eq. 2 from observed data to importance sample those terms when constructing new paths. As learning a 7D function can be challenging, many state-of-the-art methods simplify this problem by learning the 5D incident radiance (3D from $\mathbf{x}$, 2D from ω_i) instead, and combining it with BSDF-based importance sampling using either MIS [VKŠ*14; MGN17; RGH*20] or RIS [DWWH20; RMW*25; HSRM25].

Early works on path guiding [Jen95; SL06; BAJ08] construct directional histograms to guide unidirectional path tracers by gathering information from nearby photons that were distributed during a photon tracing pre-processing pass. Their two limiting factors are the need of a photon pre-pass and the need for expensive photon look-ups for every sample. Lafortune et al. [LW95] circumvent these two limitations by caching the spatio-directional data from a unidirectional path tracer on-the-fly into a 5D tree that is adapted to the sample density. Using resampling, this cache is used to importance sample the full product. However, as it operates on the 5D space directly it suffers from data sparsity and high storage cost. Vorba et al. [VKŠ*14] propose another two-stage method that learns 2D parametric mixture models based on nearby cached samples. The models are evaluated in a lazy fashion only if no nearby models exist already. The regional extent covered by each model adapts to the content of the scene by depending on the angular frequency of the model and geometrical differences.

Recent approaches [MGN17; HZE*19; DWWH20; RGH*20] learn and update a spatio-directional model on the fly during rendering, similar to Lafortune et al. [LW95], but explicitly separate the spatial and directional data-structures. They subdivide the 3D space using a k-d tree with each of its leaf nodes pointing to a local (2D) directional guiding distribution that can be sampled and evaluated. The key idea is to subdivide the 3D space such that each leaf node corresponds to a region of approximately constant incident radiance. Then a single guiding distribution can be used to model incident radiance for the whole region. Other methods still subdivide space, but learn higher dimensional spatially-varying local models per leaf-node [DPÖM22; SHJD22]. Noteworthy is the work from Ruppert et al. [RHL20] that introduces directional guiding distributions that can be conditioned on positions to compensate for parallax changes.

The assumption of (near) constant radiance within subdivision regions underpins many of the existing methods. However, except photon-based learning [Jen95; SL06; BAJ08; VKŠ*14], none of the forward-path guiding methods explicitly drives the spatial subdivision based on the variation in the incident radiance. For instance, Müller et al. [MGN17] drives the subdivision solely based on the number of samples, and splits dimensions in a round robin fashion always in the center of the cell. Ruppert et al. [RHL20] refines this strategy by tracking the spatial mean and variance of non-zero samples observed within that cell, and splits along the dimension of highest variance at the mean location. We build our method upon Intel Open Path Guiding Library [HD22], which implements the same subdivision approach.

In the context of path guiding, prior work has used neural networks to either assist established guiding structures or fully replace them. The work of Zhu et al. [ZXS*21a] improves the learning of directional distributions by using a neural network to construct/denoise quadtree-based representations from a set of radiance samples generated by a path tracer. Their follow-up work [ZXS*21b] further improves the fitting quality by using a neural network to incorporate additional information from photons generated by a photon mapper. First attempts to fully replace the guiding structure use deep neural networks to encode the whole spatio-directional guiding structure of a scene [MMR*19; ZZ19]. To improve performance, recent neural path guiding approaches utilize spatial grid embeddings [MESK22] to encode small neural network representations of local guiding structures, such as neural parametric mixtures [DWL23; HIT*24] or grid-based representations [FHK25; RMW*25]. While these methods show impressive results, they rely on a limited resolution of the hash grid coordinates to spatially subdivide the scene, and the optimal resolution may vary with scene complexity or size. Our method relies on k-d tree subdivision, which is more adaptive than fixed-resolution hash grids because of progressive refinement.

2.3. Radiance Caching

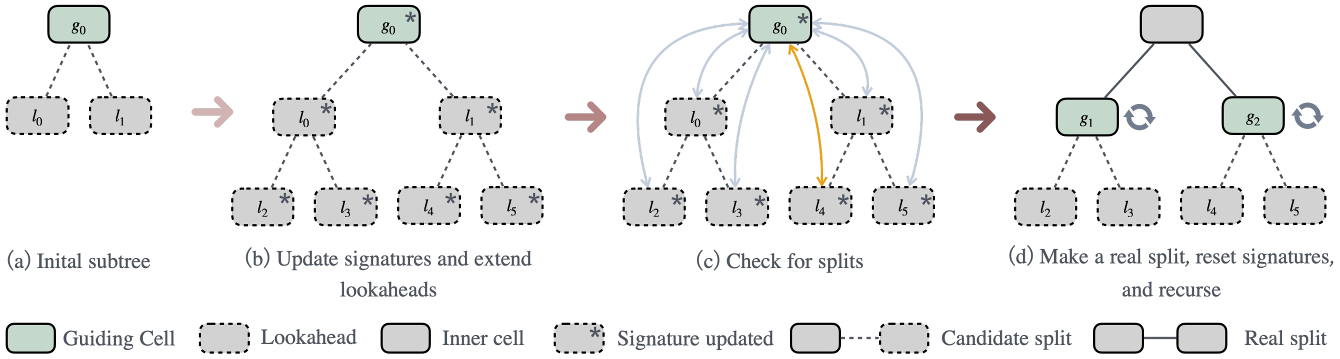
Radiance caching is a typically biased variance reduction technique that caches the scene's irradiance [WRC88; WH08] or radiance [KGPB08] and reuses it directly for estimating indirect illumination. One key problem of radiance caching is how to adaptively place the cache records across the scene to minimize interpolation artifacts. The seminal work by Ward et al. [WRC88; WH08] focuses on indirect diffuse, slowly-changing illumination across the scene. They propose lazy evaluation that reuses existing cache records within a certain radius of the lookup location, and only place a new record when no nearby record is present. The radius is set adaptively based on an estimated irradiance gradient that is derived solely from geometrical information.

Follow-up works [KGW*08; GKB09] extend radiance caching to scenes with glossy surfaces. They improve the adaptive record placement by upper-bounding the record's radius to a quantity that is inversely proportional to the spatial gradient of the outgoing radiance towards a certain direction. This means that higher local radiance variation leads to denser records. Similarly, one component of our proposed method (Sec. 5) detects the spatial change of the mean radiance direction, which we use as one of the characteristics for changing distributions that warrant a spatial split. Huang et al. [HR10] present a real-time rendering technique based on precomputation, where they propose a sparse construction of the light transport matrix. They first sparsely sample points uniformly in the scene, then estimate the region's illumination complexity by computing the local rank of the light transport matrix at the sampled points, and finally send more samples to refine to the regions with high rank.

3. Our Approach

We now give an overview of our method and explain our basic approach. To ensure ease of integration, we stick to the established k-d tree, including the definition of the candidate splits based on the available sample statistics (Sec. 2.2). We focus entirely on the binary decision whether a particular leaf node of the k-d tree used for guiding (i.e. a guiding cell) should be split into two smaller guiding cells or not. That is sufficient to enable the tree to adapt to the spatial variation in illumination and the available samples.

The main question is whether smaller cells can describe the radiance field significantly more faithfully. That can only be answered if we have information about the radiance field in such smaller cells. When important illumination features are misaligned with the split axis or small, a single split might not give a big improvement, while splits deeper down would (Fig. 11). Our splitting strategy has



(a) Inital subtree (b) Update signatures and extend lookaheads (c) Check for splits (d) Make a real split, reset signatures, and recurse

Figure 2: *Our subdivision scheme at a guiding cell g_0 during one training iteration. If the initial subtree (a) does not have enough levels of lookahead cells yet, new levels are created whilst updating signatures (b). We use two levels for this illustration, but more in practice. Then we compare the guiding cell to all of its lookahead cells (c). The lookahead l_4 triggers a split (yellow arrow). As a result, two lookahead cells become guiding cells (d). Then we repeat the procedure recursively for these two guiding cells (a-d).*

to look ahead deep enough in the tree to recognize such potential improvements. That motivates our use of multiple levels of lookahead cells.

3.1. Lookahead Cells

Lookahead cells extend the existing k-d tree with a fixed maximal number of additional levels. In Sec. 6.5, we find that six levels (corresponding to maximally $2+4+\cdots+64=126$ lookahead cells) provide sufficient spatial sensitivity while keeping the cost reasonable. Each cell is a candidate that would eventually be used for guiding, if the tree were split deeply enough. Since the k-d tree itself is built based on the spatial distribution of samples, lookahead cells are only created once their parent contains at least 1000 samples.

The lookahead cells themselves do not carry guiding distributions. The overhead of fitting these distributions is quite substantial, both in terms of storage and computation. Since we wind up having significantly more lookahead cells than guiding cells in the tree, this additional cost in terms of memory and training time would not be justified by better spatial subdivision.

Instead, the lookahead cells store a more compact representation of the radiance field. We call these compact representations *signatures*. Each signature is computed from the path tracer samples collected in the lookahead cell. The computation simply amounts to adding a contribution per lookahead cell. We have experimented with many designs for these signatures, but eventually settled on the two with the smallest memory footprint, since their results were on par with more complicated designs: We use the mean incoming radiance from all directions (Sec. 4) and the radiance-weighted mean direction (Sec. 5). While it is possible that these signatures are identical for two substantially different radiance distributions, these cases are rare enough with our combination of signatures to not cause problems in practice when they are compared in local neighborhoods (Sec. 6.6).

3.2. Splitting

Prior to each training iteration, we compare each lookahead cell to its ancestor that is a guiding cell (for which we also store signatures). We have derived split criteria that compare two signatures to assess whether we can confidently say that their difference is large enough to justify a split decision (Sec. 4.2 and 5.3). If either one of the stored signatures in any of the lookahead cells of a guiding cell indicates that a split should happen (e.g. l_4 in Fig. 2c), we split the guiding cell into two new guiding cells (e.g. g_1, g_2 in Fig. 2d). At first, we only make one split, independently of the level of the lookahead, which triggered the split. The reason is that the new guiding cells may no longer be different enough from that lookahead to justify the split.

As a result of the split, we obtain two new guiding cells. We expect the quality of samples to improve after a split. Therefore, we reset the signatures in all lookahead cells after a split and reinitialize them using the batch of samples from the current training iteration. That improves robustness against high-variance samples in early iterations and we still observe a sufficiently fast pace of subdivision. Sec. 7.2 discusses potential alternatives. We then repeat the whole splitting procedure recursively for the two new guiding cells. Through this recursion, we can theoretically split arbitrarily many levels in a single training iteration. In practice, we observe fast splitting at the beginning, when guiding cells are large, and then it slows down (Sec. 6.2).

4. The Mean Radiance Signature

The simplest way to tell if two cells are illuminated differently is to check whether the overall brightness of the illumination differs. This approach leads to the mean radiance signature, which estimates the integral over incoming radiance from all directions. It has a minimal memory footprint and excellent sensitivity to shadow boundaries and bright caustics. In theory, it is possible that two directional radiance functions only differ by a constant factor, such that the same guiding distributions could be used although the mean radiance differs. In practice, this case is unlikely to occur. The split

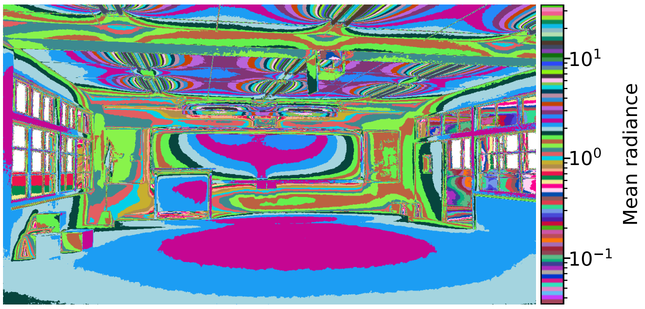


Figure 3: *Per-pixel mean radiance values for the classroom (Fig. 12) binned on a logarithmic scale. Bins are 10% apart. When we split guiding cells as soon as the mean radiance in a lookahead cell differs by at least 5%, we expect to obtain cells whose size matches the spacing of these bands. In most regions, that adapts well to the variability of the illumination, but the floor is a failure case: It is lit uniformly, but different light sources are dominant in different places and the guiding distributions should adapt to that.*

criterion triggers if we are confident that the relative difference in mean radiance between guiding and lookahead cell surpasses a threshold. That results in a logarithmic partitioning of mean radiance values as illustrated in Fig. 3. The drawback of the mean radiance signature is that it has no sensitivity to purely directional changes in the radiance distribution, which is why we introduce an additional signature in Sec. 5.

4.1. Mean Radiance Estimates

We define the mean radiance at a point $\mathbf{x} \in \mathbb{R}^3$ by $\int_{\mathbb{S}^2} L_i(\mathbf{x}, \omega)\, \mathrm{d}\omega \in \mathbb{R}$. Unlike the irradiance, this mean radiance does not depend on the surface normal, which is desirable since guiding learns incoming radiance, not reflected radiance. Our mean radiance signature is also subject to spatial averaging across the locations of training samples in a lookahead cell. Since this mean radiance cannot be known a priori, we have to estimate it based on noisy path tracer samples. Let $N_l \in \mathbb{N}$ denote the number of available training samples in a lookahead cell with cell index $l \in \mathbb{N}_0$. Let $L_0, \ldots, L_{N_l-1} \in \mathbb{R}$ be the corresponding noisy radiance estimates (including the sample contribution weight, e.g. the reciprocal density w.r.t. solid angle). In RGB renderers, we reduce the radiance to a scalar by taking the maximal component. Then our mean radiance estimate is the sample mean

$$\bar{L}_l := \frac{1}{N_l} \sum_{k=0}^{N_l-1} L_k.$$

Of course, this estimate will be noisy and we do not want split decisions to arise from random fluctuations in the Monte Carlo samples. Therefore, it is important to explicitly model the uncertainty in the mean radiance estimate $\bar{L}_l$. To this end, we note that the radiance estimates $L_0, \ldots, L_{N_l-1}$ are nearly independent random variables: Sometimes one path may result in multiple vertices for the same cell. Additionally, samples from earlier training iterations influence the guiding distribution in subsequent iterations. Other than that, the estimates arise from independent random walks.

Then $\bar{L}_l$ is the sum of many nearly independent random numbers. According to the central limit theorem [Kni99, p. 145], its distribution approaches a normal distribution as N_l grows. How many samples are needed to get reasonably close to a normal distribution depends on the distribution of the individual samples. In the presence of extremely rare firefly samples, it may take a long time. Nonetheless, modeling $\bar{L}_l$ as normal-distributed is the most viable approach. To be reasonably confident that this model holds, we enforce a minimal sample count of 1000 samples for both the guiding cell and the lookahead cell before we may decide to split.

The moment estimates up to second order provide sufficient statistics [Kni99, pp. 182, 186] for a normal distribution: That means that they carry all information about the normal distribution that can be extracted from the available sample. Additionally, they can be computed incrementally, since they arise from summation over all samples:

$$b_{j,l} := \sum_{k=0}^{N_l-1} L_k^j, \quad \text{where } j \in \{0,1,2\}.$$

These sufficient statistics serve as input to our split criterion described below. For extremely large sample counts, numeric cancellation in the summation becomes a concern, so we divide the moments $b_{0,l}, b_{1,l}, b_{2,l}$ by two once per 8 million samples.

4.2. Split Criterion for the Mean Radiance

To decide whether a guiding cell with index $g \in \mathbb{N}_0$ should be split, we compare its signature to all of its lookahead cells, using $l \in \mathbb{N}_0$ as index for the lookahead cell. We want to know whether the relative difference of the exact signatures surpasses a threshold $T > 0$, i.e.

$$\frac{|\mathrm{E}(\bar{L}_g) - \mathrm{E}(\bar{L}_l)|}{\mathrm{E}(\bar{L}_g)} > T,$$

where E denotes the expected value, i.e. the noise-free ground truth. Since we do not have access to this ground truth value, we have to make a decision based on the noisy estimates $\bar{L}_g, \bar{L}_l$ and our model of their distribution. We choose a sufficiently small probability of false positive split decisions $\alpha > 0$ (in most of our experiments $\alpha = 10^{-4}$). Then we make a split if

$$\mathrm{P}\left(\frac{|\bar{L}_g - \bar{L}_l|}{\bar{L}_g} > T\right) > 1 - \alpha, \tag{3}$$

where P denotes the probability of the stated event. That means that, under our model of a normal distribution, we make a split if our confidence that the relative difference in the mean radiance surpasses the threshold T is at least $1 - \alpha$.

Appendix A derives a closed-form formula to make the split decision according to Eq. 3 based on the moments $b_{0,l}, b_{1,l}, b_{2,l}, b_{0,g}, b_{1,g}, b_{2,g}$ and the parameters T, α.

5. The Mean Direction Signature

Our splitting based on mean radiance picks up on many changes in illumination, but has a blind spot on evenly lit surfaces where there is still directional variation in the illumination. To complement this strategy, we need another signature that is sensitive to directional

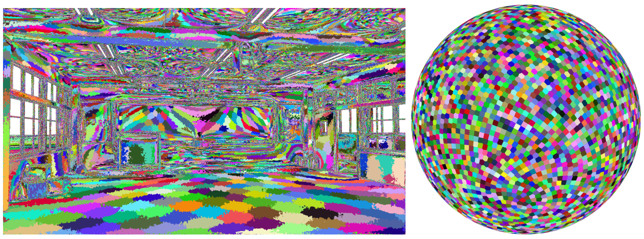


Figure 4: *Per-pixel mean direction vectors for the classroom (Fig. 12) binned as illustrated on the right. The bins are constructed to have roughly 3° diameter. Unlike the mean radiance, the mean direction detects changes for the illumination on the floor (cf. Fig. 3). We expect that cells arising from the split criterion for mean directions match the size of these binned regions.*

changes. After explaining our choice of the mean direction, we proceed similarly as before to model the distribution of mean direction estimates and derive a split criterion from this model.

5.1. Choice of the Mean Direction

Fig. 3 shows a typical failure case of the mean radiance signature. Many man-made scenes are designed to achieve uniform illumination using many distinct light sources. Thus, the directions from which the light is coming vary substantially while the mean radiance does not.

To counteract that, we use the radiance-weighted mean direction as a second signature, i.e. a normalized version of the vector $\int_{\mathbb{S}^2} L_i(\mathbf{x},\omega)\omega\,\mathrm{d}\omega \in \mathbb{R}^3$. As with the mean radiance, we additionally take the mean across sample locations $\mathbf{x} \in \mathbb{R}^3$ inside a cell. This choice has a few advantages: It keeps the total signature memory footprint per lookahead cell small (24 bytes, see Sec. 5.2). It is simple enough to derive an efficient split criterion. It is a canonical choice, since the mean radiance corresponds to the spherical harmonics coefficient in band 0, whereas the mean direction arises from the three coefficients in band 1. It accomplishes the goal of adding directional sensitivity to our signatures, even if its ability to pick up on high-frequency changes in the illumination is limited (Fig. 4).

5.2. Mean Direction Estimates

As in Sec. 4.1, we now consider the N_l samples that fall into cell l with incoming radiance estimates $L_0,\dots,L_{N_l-1}$ and corresponding incoming light directions $\omega_0,\dots,\omega_{N_l-1} \in \mathbb{S}^2$. The additional data that we store per lookahead cell are

$$\bar{\omega}_l := \sum_{k=0}^{N_l-1} L_k\omega_k \in \mathbb{R}^3.$$

Together with the moments $b_{0,l}, b_{1,l}, b_{2,l}$, that amounts to 24 bytes per lookahead cell when using 32-bit floats. Note that additional memory is needed to define the k-d tree based on the spatial distribution of samples [RHL20]. Normalizing $\bar{\omega}_l$ gives us the mean direction estimate $\frac{\bar{\omega}_l}{\|\bar{\omega}_l\|}$.

Like the mean radiance estimate, the mean direction estimate is noisy. We have to model the distribution of our estimates. Though, the central limit theorem is not directly applicable, because we are concerned with a spherical distribution here. In graphics, von Mises-Fisher (vMF) distributions [FLE87, pp. 86-89] are the most established way to model such distributions. We use a maximum-likelihood estimate [BDGS05] to model the ground-truth distribution of radiance-weighted directions in this manner. The vector $r_l := \frac{\bar{\omega}_l}{b_{1,l}}$ provides sufficient statistics for this purpose. Then the mean direction and concentration parameter of the vMF distribution for radiance-weighted directions are

$$\mu_l^\star := \frac{r_l}{\|r_l\|}, \qquad \kappa_l^\star := \frac{\|r_l\|(3-\|r_l\|^2)}{1-\|r_l\|^2}.$$

The corresponding PDF at $\omega \in \mathbb{S}^2$ is

$$p_l^\star(\omega) := \frac{\kappa_l^\star}{4\pi\sinh\kappa_l^\star}\exp(\kappa_l^\star\omega\cdot\mu_l).$$

Under our model, the random variable μ_l, is a weighted sum of N_l realizations of random variables with this distribution. The mean direction estimate $\frac{\bar{\omega}_l}{\|\bar{\omega}_l\|}$ will not be exactly vMF-distributed, even if the individual random variables are, but we model it in this manner. The mean direction is the same, namely $\mu_l := \mu_l^\star$. For the concentration parameter κ_l, we expect that it is greater, since more samples result in less uncertainty. In Appendix B, we derive the following formula for it, which uses a function fit to make evaluation practical:

$$\kappa_l = \sqrt{(y-1)\frac{y^3+1.69934861y^2+5.38753272y+9.85021305}{y^2+0.67453491y+4.31180006}},$$
$$\text{where } y := \frac{b_{1,l}^2}{b_{2,l}}(\kappa_l^\star\coth\kappa_l^\star - 1) + 1. \tag{4}$$

Thus, we now have a model of the uncertainty in the mean direction estimate. Unlike the mean radiance estimate, we do not have a strong justification for this model through the central limit theorem. The central limit theorem justifies treating $\bar{\omega}_l$ as having a trivariate normal distribution. However, normalizing $\bar{\omega}_l$ will not result in a vMF distribution then, most importantly because a vMF distribution is always radially-isotropic. It is possible to derive a model that supports anisotropy here, but it would inflate the storage cost of the signatures and would make the parameter estimation and the split criterion more complicated and less computationally efficient. Therefore, we prefer our vMF model.

5.3. Split Criterion for the Mean Direction

We define our split criterion in an analog fashion to Sec. 4.2. If we had the ground-truth mean direction of guiding cell $g \in \mathbb{N}_0$ and lookahead cell $l \in \mathbb{N}_0$, we would split if the angle between the mean directions exceeds a threshold $\tau > 0$. In the presence of uncertainty, we split if our vMF model of direction estimates indicates that the probability of this event is at least $1-\alpha$, i.e.

$$\mathrm{P}(v_g \cdot v_l < \cos\tau) > 1-\alpha,$$

where v_g, v_l denotes vMF-distributed random variables with mean direction μ_g, μ_l and concentration κ_g, κ_l, respectively.

The most direct way to implement this criterion would be to solve a double integral over the two vMF densities, but this integral appears to be intractable. Appendix C derives that instead of working with two vMF distributions, we can work with one vMF distribution using the effective concentration parameter $\kappa := (\kappa_g^{-1} + \kappa_l^{-1})^{-1}$ while replacing the other distribution by a Dirac-δ. Essentially, κ replaces κ_g, while $\kappa_l \to \infty$. With this modification of the problem statement, the remaining parameter space is small enough to work with a 1D lookup table: The values of τ, α are constants. Thus, for each possible value of κ, there is a single well-defined angle $\beta(\kappa)$ between the estimated mean directions μ_g, μ_l beyond which we should split. Then our split criterion is $\mu_g \cdot \mu_l < \cos\beta(\kappa)$ and $N_g, N_l > 1000$ (we enforce the same minimal sample count as for the mean radiance).

To find $\beta(\kappa)$, we run a preprocessing step that uses numerical cubature to compute $\mathrm{P}(v_g \cdot v_l < \cos\tau)$ and bisection to find the value of $\beta(\kappa)$ where this probability surpasses $1-\alpha$ for each possible value of κ. To ensure that linear interpolation works well on this lookup table, we parameterize it in terms of $\frac{1}{\sqrt{\kappa}}$ using 256 equidistant samples and store $\cos\beta(\kappa)$ for each sample. The largest value for $\frac{1}{\sqrt{\kappa}}$ where a split may occur at all is either the value where $\beta(\kappa) = \pi$ or $\beta(\kappa) = 0$, depending on which of these exists.

6. Results

In this section, we compare our signature-based splitting strategy to sample-count-based strategies on a set of scenes with different challenges. In addition, we evaluate how our subdivision strategy adapts to the changes in the light field, represented by the guiding structure, when next-event estimation is turned on or off and how robust our strategy is to changes in the rendering resolution. To understand the behavior of the parameters of our strategy (e.g. the lookahead depth), we perform multiple ablation studies.

6.1. Implementation and Experimental Setup

We integrated our signature-based spatial subdivision strategy into a fork of OpenPGL [HD22] and built upon their provided PBRT version [PJH23]. When using path guiding, we always use the *parallax-aware von-Mises Fisher mixtures* [RHL20] for the directional guiding distributions and combine them with BSDF sampling using the RIS strategy [Tal05; HSRM25]. Our supplemental results show similar behavior with other directional models such as quadtrees. Furthermore, we disable the stochastic lookup in OpenPGL to highlight the differences between the different subdivision methods and the quality of the per-cell guiding distributions. In all of our experiments, we keep next-event estimation (NEE) turned on unless explicitly stated differently, and our guiding structure learns the MIS-weighted portion of the direct light contribution as proposed by Ruppert et al. [RHL20]. To increase the robustness of the guiding structures, we employ minimal path space regularization with a regularization strength of $\gamma = 0.125$ [WDH*21; HSRM25]. All renderings are performed using a maximum path depth of 20. To avoid additional variance due to early path termination, we use adjoint-driven Russian roulette [VK16] for all renderings. If not mentioned differently, all images are rendered at a resolution of 1920 × 1080 (Head and CornellBox use 1080 × 1080). We measure the rendering quality of different techniques using the mean relative absolute error (MRAE) and the mean relative squared error (relMSE), where we trim the top 0.1% of pixel errors to achieve robust measurements that are not spoiled by individual fireflies. As for the split threshold for the mean radiance, we use $T = 5\%$.The angle threshold for the mean direction is $\tau = 3°$. Most experiments use $\alpha = 10^{-4}$ (Sec. 6.7). All tests were performed on the same machine using an AMD Ryzen 7 7800X3D CPU with 8 cores, 16 threads, and 64 GB of memory. To evaluate the performance and behavior of our subdivision strategy, we use a collection of scenes with different illumination challenges (Fig. 6). Detailed descriptions of the scenes can be found in Appendix D.

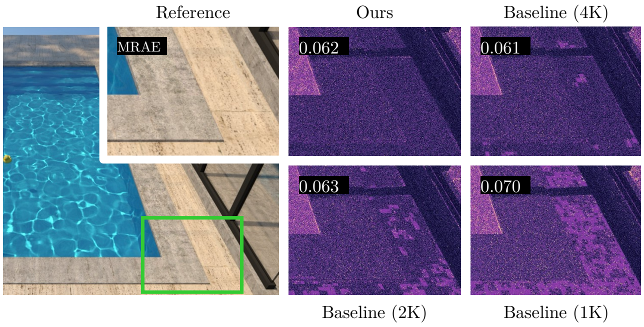


Figure 5: *Block artifacts arise due to degraded guiding distributions when splitting with overly small sample count thresholds in the baseline. Our method generates more stable guiding distributions.*

6.2. Comparison to Fixed Sample Count Thresholds

We evaluate our subdivision strategy by comparing it to sample-count-based strategies (baseline) with maximum sample count thresholds of 32K, 16K, 8K, and 4K. While lower sample count thresholds lead to a more refined guiding field, they may also cause degraded guiding distributions due to insufficient training samples (Fig. 5). This happens at random based on the available samples and causes unpredictable variations in variance between nearby cells.

For each scene, we render 256 samples per pixel (SPP) where the guiding distributions are only trained with the first 128 SPP. We compare the final renderings in Fig. 6. We also provide an interactive, full-resolution comparison of renderings and subdivisions as supplemental material.

In Pool, all methods subdivide the caustic regions aggressively, yielding comparable rendering quality. However, our method produces far fewer splits than the baselines at the floor next to the pool which exhibits simple direct illumination by the sun (Fig. 10). The tree depth visualization reveals an interesting correlation between our subdivision strength and the caustic patterns, indicating that our method allocates complexity where it is needed the most.

In DiningRoom, the relatively uniform indirect lighting on the back wall causes our method to split less often than all baseline methods while maintaining similar rendering quality. High radiance variation at the image frame, the lamps, the table, and on the

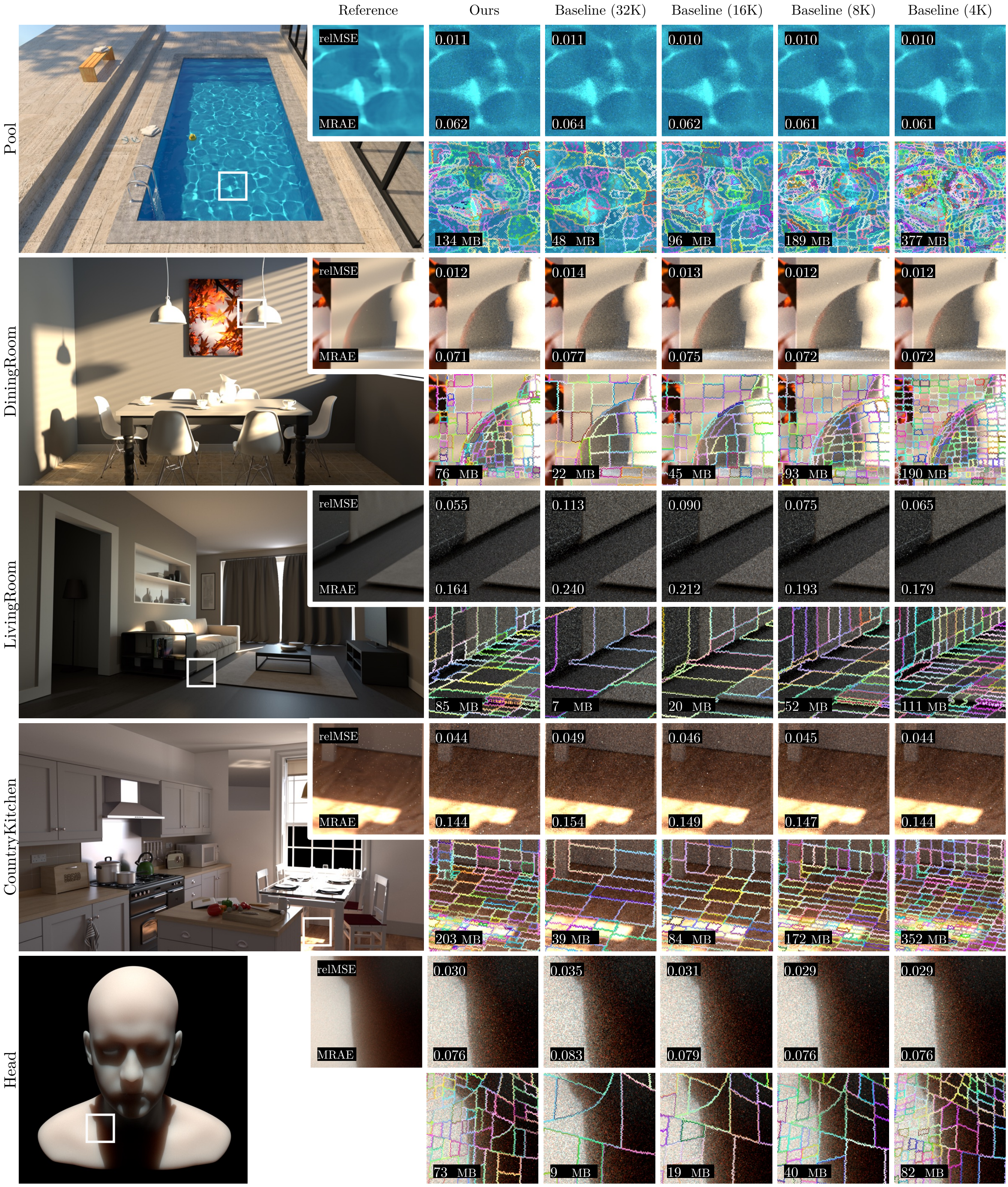


Figure 6: *Comparing our subdivision strategy based on signatures (Ours) against the baseline subdivision strategy using different maximum sample count thresholds:* 32*K,* 16*K,* 8*K and* 4*K. The images show equal-sample comparisons using* 256 *SPP, where the guiding structure is trained during the first* 128 *SPP. The full image errors of the different methods are reported using MRAE and relMSE on the first rows of each scene. The memory footprints of the guiding structure (including the lookaheads and signatures) are shown on the second rows.*

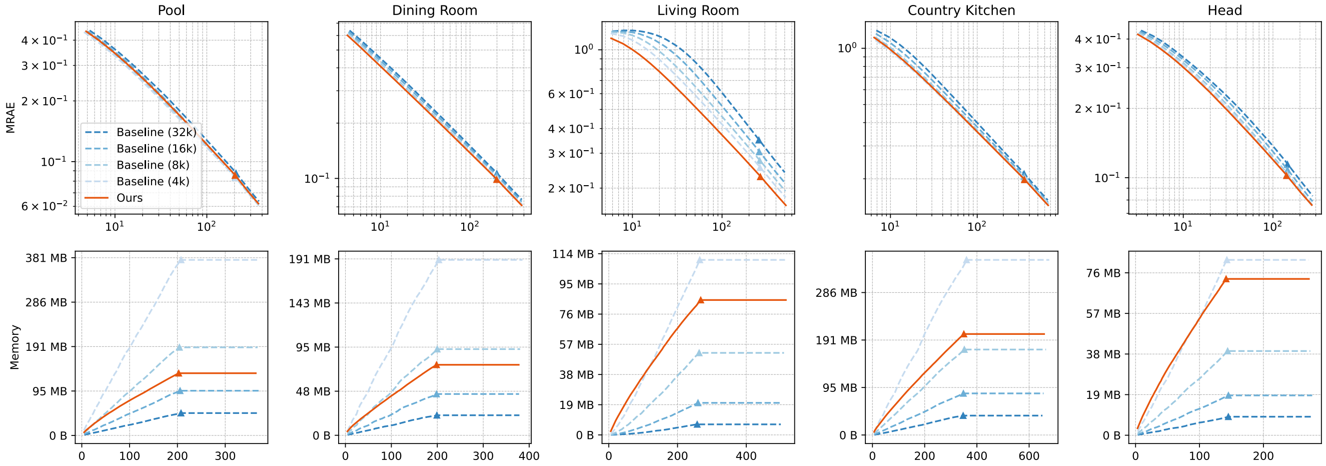


Figure 7: *MRAE and memory cost of the guiding field plotted as a function of time (in seconds) across our test scenes. The triangle markers annotate the time when each method finishes training.*

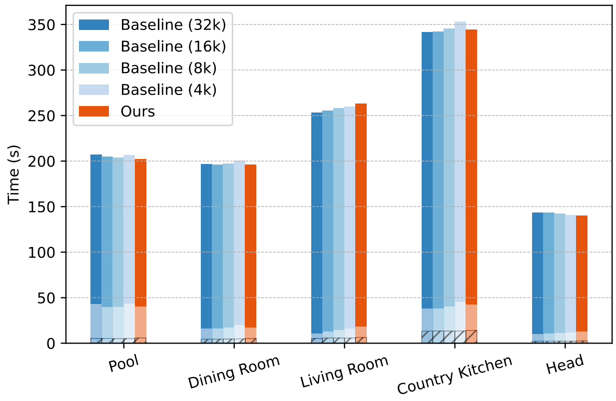


Figure 8: *Training-phase (first* 128 *SPP) timing of different subdivision strategies across our test scenes. The top solid segments of the bars show the time spent on pure rendering. The middle transparent segments indicate the time purely spent on training the mixture models. The bottom hatched ones represent the time purely spent on spatial subdivision, including signature maintenance and split criteria overhead. The* ***guiding structure overhead*** *is the* ***sum*** *of the middle and bottom segments.*

floor is also captured, producing more adaptive splits. As a result, we achieve a slightly lower MRAE than all baselines.

In LivingRoom, our method achieves a significantly lower rendering error than the strongest baseline (4K). Zooming into the inset, we detect and refine geometric details such as the corner of the couch and the folds of the blanket. As illustrated in Fig. 1, our method also better captures high-frequency illumination variation on the wall and in the niche.

We perform similarly to all baseline methods in CountryKitchen, where we adaptively subdivide the challenging caustic on the floor and around the chair legs with abrupt geometric change.

Table 1: *Average guiding structure overhead of different subdivision strategies. The first four columns are baseline methods.*

Method	32K	16K	8K	4K	Ours
Overhead	10.4%	10.3%	10.8%	11.8%	11.5%

Finally, in Head, our method learns the appropriate pace of splitting and achieves similar rendering quality compared to the baseline (4K). Our subdivision adapts to the shadow boundaries at the neck, generating smooth shadow transitions on par with baseline (4K).

To have a deeper understanding of the behavior of all strategies, Fig. 7 provides a quantitative study of rendering error and memory cost. Overall, we consistently outperform baseline (32K) in terms of MRAE at equal time. In DiningRoom, LivingRoom, CountryKitchen, and Head, our method even surpasses all the baseline methods, including the aggressive baseline (4K). From the *Memory* plot, we can see that the baseline methods exhibit a linear growth of memory (proportional to the number of guiding cells) with respect to time during the 128-SPP training phase. Our method, in contrast, performs fast subdivision in the early iterations. This is crucial for low-lighting scenes such as LivingRoom where effective sampling depends heavily on the guiding distribution. By quickly allocating sufficient guiding cells at the beginning, our method accelerates convergence, evidenced by the fast error reduction in the plot of LivingRoom. At later iterations, we observe a gradual slowdown of subdivision, producing fewer total guiding cells than baseline (4K).

It is worth noting that the *guiding structure overheads* (the sum of the time spent on training distributions and spatial subdivision) of different methods are similar (Fig. 8). Table 1 provides an av-

Proceedings published by Eurographics - The European Association for Computer Graphics.

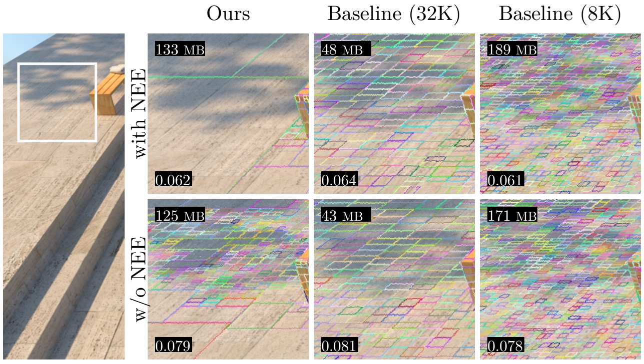


Figure 9: *Showing how our illumination-aware subdivision strategy (**left**) reacts, compared to baselines (**center** and **right**), when the incident radiance distribution stored in the guiding structure adjusts to the use of next-event estimation (NEE): **top:** with NEE and **bottom:** w/o NEE). Rendering uses* 256 *SPP, where we train the guiding field during the first* 128 *SPP. Guiding structure memory costs are reported in the upper image labels. MRAEs of the full images are reported in the lower labels.*

erage insight by dividing the total guiding structure overhead accumulated across all scenes by the total rendering time during the training phase. It shows that the average guiding structure overhead of all methods is around 11%, meaning the increased overhead using our adaptive subdivision (computation and storage of our signatures, as well as checking the split criteria) is negligible. In some cases (POOL and HEAD), our method finishes rendering even earlier than the baseline methods.

6.3. Effect of Next-Event Estimation

One of the main advantages of our proposed subdivision strategy is that it adapts to the features of the incident radiance distribution, whereas the baseline is oblivious to it. We can showcase this by examining how our method reacts to next-event estimation (NEE) being turned on or off: When NEE is disabled, path guiding learn the full incident illumination, including direct illumination. In contrast, when NEE is enabled, path guiding only considers the MIS-weighted portion of the direct illumination. Since our signatures are based on the incident radiance that is passed to the path guiding structure, they will adapt to both situations. The inset in Fig. 9 exemplifies this in POOL. With NEE enabled, only little subdivision occurs since the sun casting the shadow is not learned by guiding as it can be sampled effectively by NEE alone. However, when NEE is disabled, path guiding needs to learn the occlusion patterns of the tree, and therefore the incident radiance used to fit the path guiding structure changes quickly at the shadow boundaries. Our subdivision scheme correctly detects this and triggers many subdivisions in the penumbra regions. In contrast, the baseline subdivision does not adapt at all to these two different situations.

6.4. Adaptivity to Rendering Resolution

The number of splits in the baseline strategy is strongly dependent on the resolution of the rendering, since resolution affects the total number of samples seen per region of space. Our method, however, does not explicitly encode the sample count in our split criteria, except for the minimal sample threshold required by the central limit theorem. In principle, this allows subdivision to be largely independent of the rendering resolution.

To validate this claim, we train both the baseline (using a 32K sample count threshold) and our method with 64 SPP under two resolution settings (720p, 2160p) in the POOL scene. After storing the resulting guiding fields, we restart evaluation renderings at 1080p without further training and compare them side by side in Fig. 10.

The results confirm our expectations. Our subdivision is less sensitive to rendering resolution, as seen in the visualizations of subdivision and tree depth maps. By contrast, the baseline produces a nearly uniform subdivision across the scene, with the tree depth increasing significantly at higher resolutions. In the red insets of Fig. 10, the baseline subdivides the uniformly-lit tiles aggressively. This effect is amplified when the resolution increases. Conversely, our method detects the uniformity of the illumination and stops subdivision consistently for different resolutions.

6.5. Ablation of Maximum Lookahead Depth

In our method, the lookahead depth has to be sufficiently high to achieve spatial sensitivity for high-frequency spatial changes, but an excessive choice causes a high overhead. We compare various lookahead tree depths in Fig. 11: In this test setup, we first run 32 SPP rendering with training. Then, we disable training and continue rendering with the trained guiding field until 64 SPP. To highlight the effect of subdivision, NEE is turned off. We repeat this configuration for lookahead tree depths of 1, 3, 5, 6, 7.

For small depths (1, 3), the lookahead cells are not small enough to capture the variation of lighting. Severe spatial marginalization smoothens out the mean radiance and mean direction, leading to high-frequency changes being ignored (e.g. the caustic regions in POOL). While a depth of 5 avoids most split failures, some are still overlooked on the floor of LIVINGROOM. Using 6 levels or more addresses this problem effectively. We choose 6 as our final parameter.

6.6. Ablation of Split Criteria

We demonstrate the effect of both the mean radiance and mean direction criteria on the subdivision through an ablation in Fig. 12. Like in Sec. 6.5 we show 64 SPP renderings, trained only for 32 SPP and turn off NEE.

We modified the classic CLASSROOM scene to showcase our point by removing all chairs and tables. The floor area is uniformly illuminated by arrays of fluorescent tubes, resulting in very little change in the mean radiance, despite variations in the incoming light direction and the mean direction. Relying solely on the mean radiance would lead to insufficient subdivision in those regions.

Under the water surface of POOL, the mean radiance is crucial to detect changes in brightness between caustic and non-caustic regions. The mean direction criterion, however, struggles in those regions. On the other hand, the mean direction criterion is able to

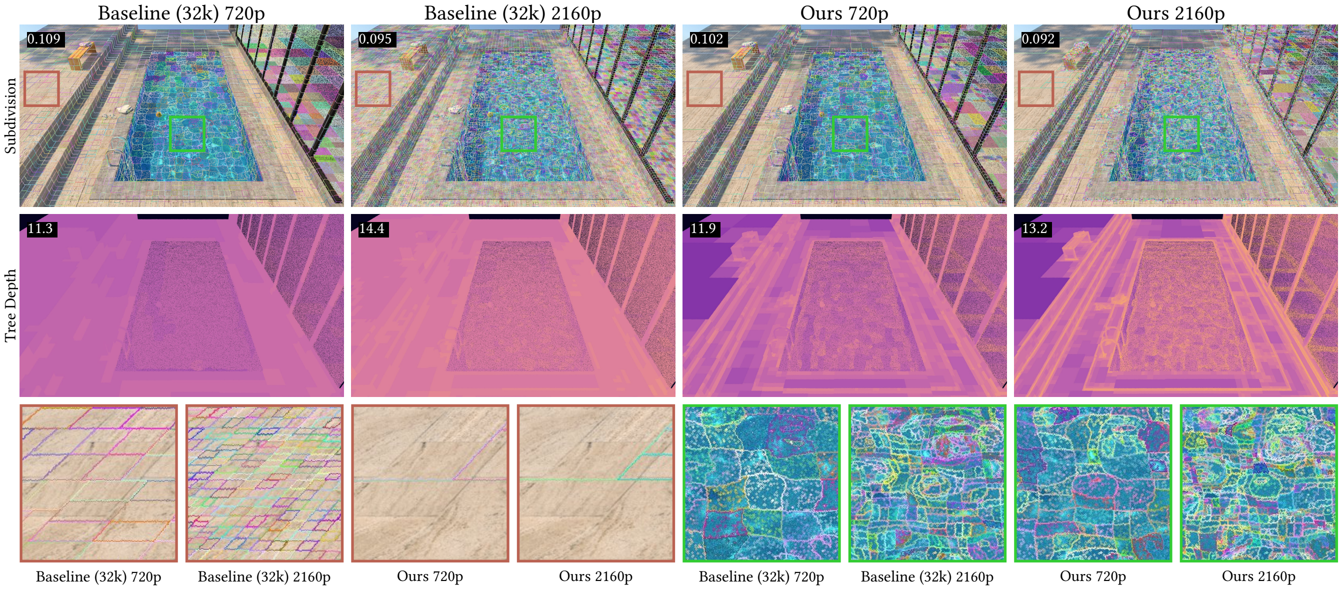


Figure 10: *Equal SPP comparison of using different rendering resolutions during training. The numbers in the "Subdivision" row indicate the MRAE. The "Tree Depth" visualizes the depth of the guiding cell in the k-d tree at the first diffuse hit position, where the numbers are the mean values across the images.*

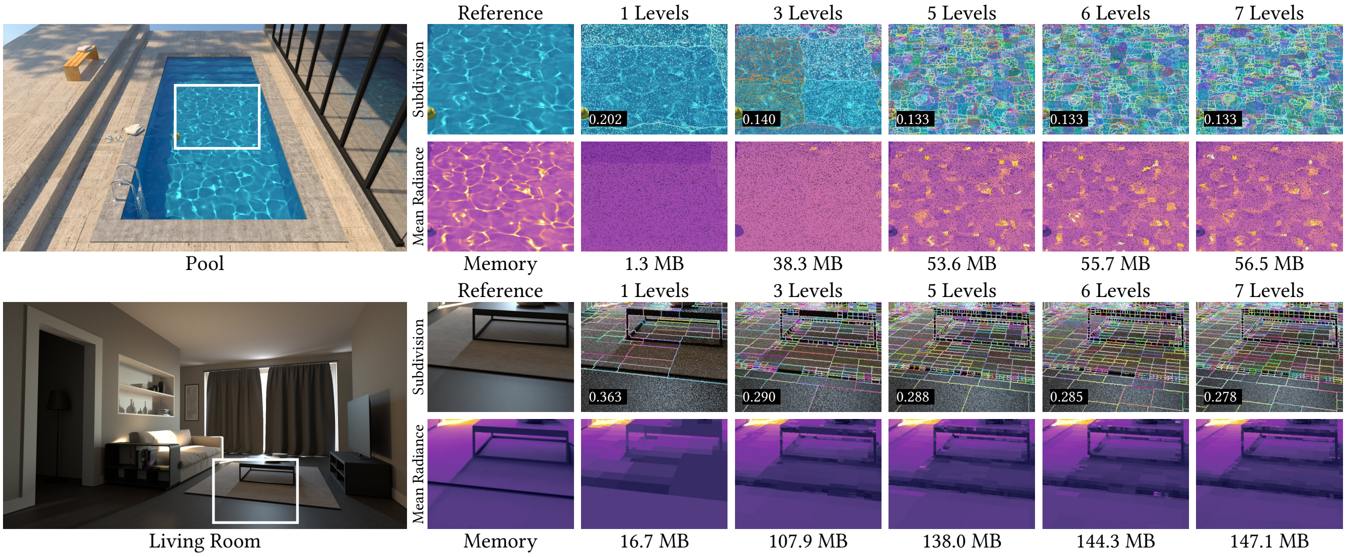


Figure 11: *Exploring the maximum lookahead depth. Numbers in the image labels indicate the MRAE of the full image. The row "Memory" displays the total memory consumption of the entire guiding field. The "Mean Radiance" of the reference crop is rendered using a custom guided path tracing for* 1024 *SPP, then denoised.*

detect abrupt changes in surface normals in shadowed corners, evidenced by the red insets. Rendering errors indicate that our full model is capable of combining the strengths of the two criteria, generating a more finely adapted guiding field than either of them.

6.7. Ablation of the Split Confidence

In this ablation, we analyze the role of the false positive split rate α for both the mean radiance (Sec. 4.2) and the mean direction split criteria (Sec. 5.3). Fig. 13 shows a 64 SPP rendering experiment in CORNELLBOX, a variant of the classic one where the light source is flipped so that most of the scene is lit indirectly. We enable either one of the criteria and sweep its corresponding α from

Proceedings published by Eurographics - The European Association for Computer Graphics.

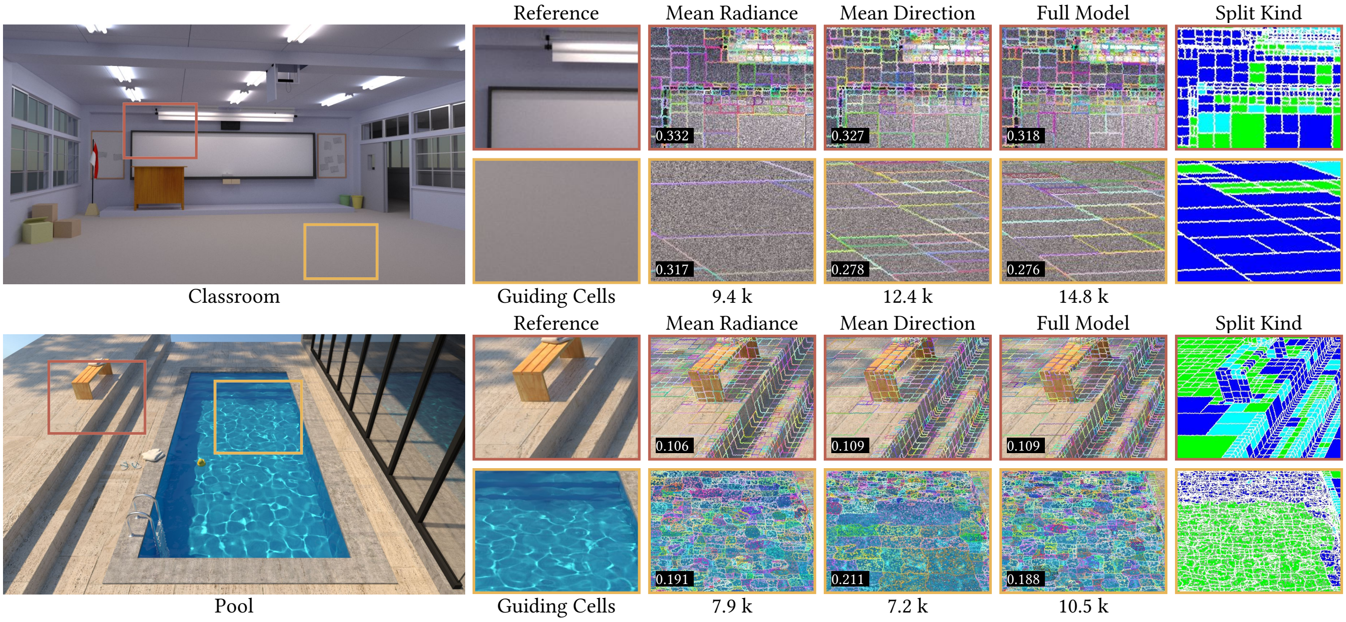


Figure 12: *Exploring the effect of each split criterion. Only the "Full Model" column uses both mean radiance and mean direction criteria. The "Split Kind" column visualizes which split criterion (the most recent split) lead to the creation of this guiding cell. Green indicates the mean radiance criterion. Blue indicates the mean direction criterion. Cyan says both criteria were triggered simultaneously. Numbers in the image labels are the MRAE of the crops. "Guiding Cells" are the cell counts of the entire field.*

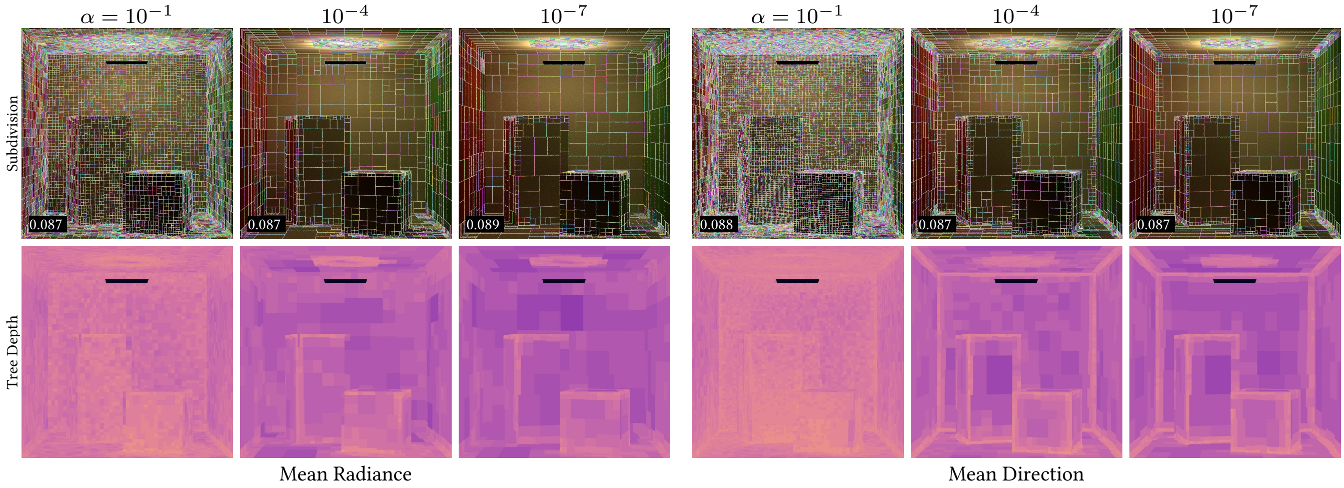


Figure 13: *Exploring parameter values for the false positive rate* α*. The left three columns use solely the mean radiance criterion, whereas the right ones use only the mean direction criterion. Numbers in the labels indicate the MRAE of the full image.*

$\{10^{-1}, 10^{-4}, 10^{-7}\}$. The training lasts through the whole rendering, with NEE turned on. For both criteria, we observe aggressive splits when α is set to a high value such as 10^{-1}, and reduced splits when decreasing it to 10^{-7}.

Though more cells are created, we do not see significant improvement of rendering when using a high value of $\alpha = 10^{-1}$. Additionally, the tree depth maps of $\alpha = 10^{-1}$ have less structure information compared to 10^{-4} and 10^{-7}, indicating increased false positive split decisions. On the other hand, using a more conservative rate such as $\alpha = 10^{-7}$ can slow down the splitting pace too much, leading to higher rendering error.

We choose 10^{-4} as our final parameter for both criteria, as it yields the best performance across our tested scenes, maintaining a good balance between fast and accurate splitting.

7. Limitations and Future Work

In the following, we discuss some of our limitations and potential directions for further improvements.

7.1. Decoupled Subdivision Strategy

Our subdivision strategy solely considers the characteristics of a scene's incident radiance field. However, it does not directly consider how well a model performs for sampling. In theory, it is possible that the guiding distribution of a parent node is not fully trained and requires additional samples to converge (i.e. learn/explore all important incident radiance features). In this case, it would be more advantageous to delay splitting and inherit a higher-quality guiding distribution to the child nodes later on. In early experiments, we explored a splitting strategy with one level of lookahead cells, which store their own guiding distributions. Splits are triggered by a large Kullback-Leibler divergence of the models and the sampled empirical distribution. Unfortunately, the additional memory and fitting overhead made our solution impractical, especially as the need for multiple lookahead levels became apparent, which led us to derive a more lightweight signature-based solution. Nevertheless, finding a solution that directly considers the current sampling quality of the parent and its potential improvements when performing splitting is an interesting direction for future work.

7.2. Split Handling

The way in which we reset signatures when a split happens (Sec. 3.2) may not be optimal. The design is motivated by high-variance samples in the initial rendering stage. However, we notice that sample variance reduces over time. Thus, *not* resetting the signatures on split may accelerate the convergence of split confidence. Exploring a more adaptive handling of the signature on split is an open direction for follow-up works. For instance, Ruppert et al. [RHL20] apply a decay to the strength of the vMF's sufficient statistics when a spatial split happens. Similarly, we could apply a decay to the signature statistics in order to combine the effect of old and new samples.

7.3. Limited Signature Expressiveness

Currently, we use two compact and relatively simple signatures to describe the incident radiance distribution. While the mean radiance signature reliably identifies changes in overall brightness, the mean direction signature is a highly compressed representation of the spherical radiance distribution, which, in rare cases, may not be able to identify changes in the incoming radiance distribution. Sometimes our method thus does not propose splits, even if the distributions differ. Additional signatures and corresponding split criteria that are both compact and more expressive would enhance our method.

7.4. Noisy Subdivision

Our split decision is made by comparing a probability estimate computed from Monte-Carlo samples against a threshold. This results in a Bernoulli-distributed random variable. Therefore, the stochastic nature of path tracing is inherited by the subdivision. Although we have a relatively safe $\alpha = 10^{-4}$, false positive splits still occur in our experiments, generating unnecessary refinements at times. In addition, we notice that more challenging light transport increases the rate at which such noisy splits occur. A potential countermeasure could be to use normality tests to discern whether available samples are sufficient to apply the central limit theorem.

7.5. Automatic Stopping Criteria for Training

We observe that the later the iterations, the more marginally the guiding distributions improve. Therefore, stopping the update of mixture models early during rendering, e.g. at 64 SPP, could increase rendering efficiency for adaptive subdivision methods like ours. Future works could explore automatic stopping criteria to balance between faster rendering and good distribution quality.

7.6. The Validity of Our Parameter Choice

The parameters of our adaptive subdivision scheme (mean radiance threshold T, mean radiance false positive split rate α, mean direction threshold τ, mean direction false positive split rate α, lookahead depth, and minimum samples before a split) are empirically chosen across our test scenes. They might not generalize to other renderers or scenes with very different characteristics. For instance, consider an extremely small shadow in a guiding cell of a uniformly-lit area. More than six levels of lookaheads may be needed to capture the shadow patch in order to trigger the split criterion.

8. Conclusions

While path guiding has seen wide adoption in recent years thanks to sophisticated and robust directional guiding distributions, the spatial subdivision has remained as a potential point of failure. The only reliable way to determine sample count thresholds is trial and error and even then one may find that a single value does not work well for all parts of a scene. Our method fills this gap in prior work by adapting the spatial subdivision to the characteristics of the integrand. It achieves rendering quality comparable to sample-count-based subdivision, if not yielding large or consistent improvements in variance. It accomplishes that with fewer guiding cells, which may make it more suitable in some path guiding implementations. At the same time, it is easy to integrate into existing guiding solutions, has little overhead, and relies on principled statistical models for its split decisions.

References

[BAJ08] Budge, B. C., Anderson, J. C., and Joy, K. I. "Caustic Forecasting: Unbiased Estimation of Caustic Lighting for Global Illumination". *Computer Graphics Forum* 27.7 (2008), 1963–1970. DOI: `10.1111/j.1467-8659.2008.01345.x` 2, 3.

[BDGS05] Banerjee, Arindam, Dhillon, Inderjit S., Ghosh, Joydeep, and Sra, Suvrit. "Clustering on the Unit Hypersphere using von Mises-Fisher Distributions". *Journal of Machine Learning Research* 6.46 (2005). URL: `http://jmlr.org/papers/v6/banerjee05a.html` 6.

[Dow73] Downton, F. "The Estimation of Pr (Y < X) in the Normal Case". *Technometrics* 15.3 (1973). issn: 00401706. doi: 10.1080/00401706.1973.10489081 15.

[DPÖM22] Dodik, Ana, Papas, Marios, Öztireli, Cengiz, and Müller, Thomas. *Path Guiding Using Spatio-Directional Mixture Models*. 2022. doi: 10.1111/cgf.14428 2, 3.

[DWL23] Dong, Honghao, Wang, Guoping, and Li, Sheng. "Neural Parametric Mixtures for Path Guiding". *ACM SIGGRAPH 2023 Conference Proceedings*. SIGGRAPH '23. Los Angeles, CA, USA: Association for Computing Machinery, 2023. isbn: 9798400701597. doi: 10.1145/3588432.3591533. url: https://doi.org/10.1145/3588432.3591533 3.

[DWWH20] Deng, Hong, Wang, Beibei, Wang, Rui, and Holzschuch, Nicolas. "A practical path guiding method for participating media". *Computational Visual Media* 6.1 (Mar. 2020), 37–51. issn: 2096-0662. doi: 10.1007/s41095-020-0160-1. url: https://doi.org/10.1007/s41095-020-0160-1 2, 3.

[FHK25] Figueiredo, Pedro, He, Qihao, and Kalantari, Nima Khademi. "Neural Path Guiding with Distribution Factorization". *Eurographics Symposium on Rendering*. Ed. by Wang, Beibei and Wilkie, Alexander. The Eurographics Association, 2025. isbn: 978-3-03868-292-9. doi: 10.2312/sr.20251178 3.

[FLE87] Fisher, N. I., Lewis, T., and Embleton, B. J. J. *Statistical Analysis of Spherical Data*. Cambridge University Press, 1987. doi: 10.1017/CBO9780511623059 6.

[GKB09] Gassenbauer, Václav, Křivánek, Jaroslav, and Bouatouch, Kadi. "Spatial directional radiance caching". *Proceedings of the Twentieth Eurographics Conference on Rendering*. EGSR'09. Girona, Spain: Eurographics Association, 2009, 1189–1198. doi: 10.1111/j.1467-8659.2009.01496.x. url: https://doi.org/10.1111/j.1467-8659.2009.01496.x 3.

[Hd14] Heitz, E. and d'Eon, E. "Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals". *Computer Graphics Forum* 33.4 (2014), 103–112. doi: 10.1111/cgf.12417 2.

[HD22] Herholz, Sebastian and Dittebrandt, Addis. *Intel® Open Path Guiding Library*. 2022. url: http://www.openpgl.org 3, 7.

[HIT*24] Huang, Jiawei, Iizuka, Akito, Tanaka, Hajime, et al. "Online Neural Path Guiding with Normalized Anisotropic Spherical Gaussians". *ACM Trans. Graph.* 43.3 (Apr. 2024). issn: 0730-0301. doi: 10.1145/3649310. url: https://doi.org/10.1145/3649310 3.

[HR10] Huang, Fu-Chung and Ramamoorthi, Ravi. "Sparsely precomputing the light transport matrix for real-time rendering". *Proceedings of the 21st Eurographics Conference on Rendering*. EGSR'10. Saarbrücken, Germany: Eurographics Association, 2010, 1335–1345. doi: 10.1111/j.1467-8659.2010.01729.x. url: https://doi.org/10.1111/j.1467-8659.2010.01729.x 3.

[HSRM25] Herholz, Sebastian, Sik, Martin, Reichardt, Lea, and Manzi, Marco. "Path Guiding in Production and Recent Advancements". *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Courses*. SIGGRAPH Courses '25. Association for Computing Machinery, 2025. doi: 10.1145/3721241.3733994 2, 7.

[HZE*19] Herholz, Sebastian, Zhao, Yangyang, Elek, Oskar, et al. "Volume Path Guiding Based on Zero-Variance Random Walk Theory". *ACM Trans. Graph.* 38.3 (2019). issn: 0730-0301. doi: 10.1145/3230635 2, 3.

[Jen95] Jensen, Henrik Wann. "Importance Driven Path Tracing using the Photon Map". *Rendering Techniques '95*. Ed. by Hanrahan, Patrick M. and Purgathofer, Werner. Vienna: Springer Vienna, 1995, 326–335. isbn: 978-3-7091-9430-0 2, 3.

[Kaj86] Kajiya, James T. "The rendering equation". *SIGGRAPH Comput. Graph.* 20.4 (Aug. 1986), 143–150. issn: 0097-8930. doi: 10.1145/15886.15902 2.

[KFF*15] Keller, A., Fascione, L., Fajardo, M., et al. "The path tracing revolution in the movie industry". *ACM SIGGRAPH 2015 Courses*. SIGGRAPH '15. Los Angeles, California: Association for Computing Machinery, 2015. isbn: 9781450336345. doi: 10.1145/2776880.2792699. url: https://doi.org/10.1145/2776880.2792699 2.

[KGPB08] Křivánek, Jaroslav, Gautron, Pascal, Pattanaik, Sumanta, and Bouatouch, Kadi. "Radiance caching for efficient global illumination computation". *ACM SIGGRAPH 2008 Classes*. SIGGRAPH '08. Los Angeles, California: Association for Computing Machinery, 2008. isbn: 9781450378451. doi: 10.1145/1401132.1401228. url: https://doi.org/10.1145/1401132.1401228 3.

[KGW*08] Křivánek, Jaroslav, Gautron, Pascal, Ward, Greg, et al. "Practical global illumination with irradiance caching". *ACM SIGGRAPH 2008 Classes*. SIGGRAPH '08. Los Angeles, California: Association for Computing Machinery, 2008. isbn: 9781450378451. doi: 10.1145/1401132.1401213. url: https://doi.org/10.1145/1401132.1401213 3.

[Kni99] Knight, Keith. *Mathematical Statistics*. Texts in Statistical Sciences. Chapman & Hall/CRC, 1999. doi: 10.1201/9780367805319 5.

[LW95] Lafortune, Eric P. and Willems, Yves D. "A 5D Tree to Reduce the Variance of Monte Carlo Ray Tracing". *Rendering Techniques '95*. Ed. by Hanrahan, Patrick M. and Purgathofer, Werner. Vienna: Springer Vienna, 1995, 11–20. isbn: 978-3-7091-9430-0 3.

[MESK22] Müller, Thomas, Evans, Alex, Schied, Christoph, and Keller, Alexander. "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding". *ACM Trans. Graph.* 41.4 (July 2022), 102:1–102:15. doi: 10.1145/3528223.3530127. url: https://doi.org/10.1145/3528223.3530127 3.

[MGN17] Müller, Thomas, Gross, Markus, and Novák, Jan. "Practical Path Guiding for Efficient Light-Transport Simulation". *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering)* 36.4 (June 2017), 91–100. doi: 10.111/cgf.13227 2, 3.

[MMR*19] Müller, Thomas, Mcwilliams, Brian, Rousselle, Fabrice, et al. "Neural Importance Sampling". *ACM Trans. Graph.* 38.5 (Oct. 2019). issn: 0730-0301. doi: 10.1145/3341156 3.

[Pet21] Peters, Christoph. "BRDF importance sampling for polygonal lights". *ACM Trans. Graph.* 40.4 (2021). doi: 10.1145/3450626.3459672 2.

[PJH23] Pharr, Matt, Jakob, Wenzel, and Humphreys, Greg. *Physically based rendering: From theory to implementation*. 4rd. Morgan Kaufmann, 2023 7.

[RGH*20] Rath, Alexander, Grittmann, Pascal, Herholz, Sebastian, et al. "Variance-aware path guiding". *ACM Trans. Graph.* 39.4 (Aug. 2020). issn: 0730-0301. doi: 10.1145/3386569.3392441 2, 3.

[RGH*22] Rath, Alexander, Grittmann, Pascal, Herholz, Sebastian, et al. "EARS: Efficiency-Aware Russian Roulette and Splitting". *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2022)* 41.4 (July 2022). doi: 10.1145/3528223.3530168 2.

[RHL20] Ruppert, Lukas, Herholz, Sebastian, and Lensch, Hendrik P. A. "Robust Fitting of Parallax-Aware Mixtures for Path Guiding". *ACM Trans. Graph.* 39.4 (Aug. 2020). issn: 0730-0301. doi: 10.1145/3386569.3392421 2, 3, 6, 7, 13.

[RMW*25] Rath, Alexander, Manzi, Marco, Weiss, Sebastian, et al. "Neural Resampling with Optimized Candidate Allocation". *Eurographics Symposium on Rendering*. Ed. by Wang, Beibei and Wilkie, Alexander. The Eurographics Association, 2025. isbn: 978-3-03868-292-9. doi: 10.2312/sr.20251181 2, 3.

[SHJD22] SCHÜSSLER, VINCENT, HANIKA, JOHANNES, JUNG, ALISA, and DACHSBACHER, CARSTEN. "Path Guiding with Vertex Triplet Distributions". *Computer Graphics Forum* 41.4 (2022). DOI: `10.1111/cgf.14582` 3.

[SL06] STEINHURST, JOSHUA and LASTRA, ANSELMO. "Global Importance Sampling of Glossy Surfaces Using the Photon Map". *2006 IEEE Symposium on Interactive Ray Tracing*. 2006, 133–138. DOI: `10.1109/RT.2006.280224` 2, 3.

[Slo08] SLOAN, PETER-PIKE. "Stupid Spherical Harmonics (SH) Tricks". *Game Developers Conference 2008*. 2008. URL: `https://www.ppsloan.org/publications/StupidSH36.pdf` 16.

[Tal05] TALBOT, JUSTIN F. *Importance resampling for global illumination*. Brigham Young University, 2005 2, 7.

[VG95] VEACH, ERIC and GUIBAS, LEONIDAS J. "Optimally combining sampling techniques for Monte Carlo rendering". *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '95. Association for Computing Machinery, 1995. DOI: `10.1145/218380.218498` 2.

[VHM*22] VERBIN, DOR, HEDMAN, PETER, MILDENHALL, BEN, et al. "Ref-NeRF: Structured View-Dependent Appearance for Neural Radiance Fields". *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022. DOI: `10.1109/CVPR52688.2022.00541` 16.

[VK16] VORBA, JIŘÍ and KŘIVÁNEK, JAROSLAV. "Adjoint-driven Russian roulette and splitting in light transport simulation". *ACM Transactions on Graphics (Proc. of SIGGRAPH)* 35.4 (July 2016). DOI: `10.1145/2897824.2925912` 7.

[VKŠ*14] VORBA, JIŘÍ, KARLÍK, ONDŘEJ, ŠIK, MARTIN, et al. "Online learning of parametric mixture models for light transport simulation". *ACM Trans. Graph.* 33.4 (2014). DOI: `10.1145/2601097.2601203` 2, 3.

[WDH*21] WEIER, PHILIPPE, DROSKE, MARC, HANIKA, JOHANNES, et al. "Optimised Path Space Regularisation". *Computer Graphics Forum (Proc. of Eurographics Symposium on Rendering)* 40.4 (July 2021), 139–151. DOI: `10.1111/cgf.14347` 7.

[WH08] WARD, GREGORY J. and HECKBERT, PAUL S. "Irradiance gradients". *ACM SIGGRAPH 2008 Classes*. SIGGRAPH '08. Los Angeles, California: Association for Computing Machinery, 2008. ISBN: 9781450378451. DOI: `10.1145/1401132.1401225`. URL: `https://doi.org/10.1145/1401132.1401225` 3.

[WRC88] WARD, GREGORY J., RUBINSTEIN, FRANCIS M., and CLEAR, ROBERT D. "A ray tracing solution for diffuse interreflection". *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '88. New York, NY, USA: Association for Computing Machinery, 1988, 85–92. ISBN: 0897912756. DOI: `10.1145/54852.378490`. URL: `https://doi.org/10.1145/54852.378490` 3.

[ZXS*21a] ZHU, SHILIN, XU, ZEXIANG, SUN, TIANCHENG, et al. "Hierarchical neural reconstruction for path guiding using hybrid path and photon samples". *ACM Trans. Graph.* 40.4 (July 2021). ISSN: 0730-0301. DOI: `10.1145/3450626.3459810`. URL: `https://doi.org/10.1145/3450626.3459810` 3.

[ZXS*21b] ZHU, SHILIN, XU, ZEXIANG, SUN, TIANCHENG, et al. "Photon-Driven Neural Reconstruction for Path Guiding". *ACM Trans. Graph.* 41.1 (Nov. 2021). ISSN: 0730-0301. DOI: `10.1145/3476828` 3.

[ZZ19] ZHENG, QUAN and ZWICKER, MATTHIAS. "Learning to Importance Sample in Primary Sample Space". *Computer Graphics Forum* 38.2 (2019), 169–179. DOI: `10.1111/cgf.13628` 3.

Appendix A: Split Criterion for the Mean Radiance

To estimate the probability in Eq. 3, we first note that the random variables $\bar{L}_g, \bar{L}_l$ are heavily correlated: Every single sample used to compute the mean radiance of the lookahead cell $\bar{L}_l$ is also used for $\bar{L}_g$ since the guiding cell encompasses the lookahead cell. To compensate for this effect, it makes sense to consider a difference cell, consisting of all parts of the guiding cell that are not contained in the lookahead cell. The corresponding moment estimates are simply $b_{j,\Delta} := b_{j,g} - b_{j,l}$ and its sample mean is

$$\bar{L}_\Delta := \frac{b_{1,\Delta}}{b_{0,\Delta}} = \frac{N_g}{N_g - N_l}\bar{L}_g - \frac{N_l}{N_g - N_l}\bar{L}_l.$$

For the random variables $\bar{L}_\Delta, \bar{L}_l$, it is reasonable to assume that they are independent since they are constructed from disjoint sets of samples. Furthermore,

$$\begin{aligned} & \frac{\bar{L}_g - \bar{L}_l}{\bar{L}_g} > T \\ \Leftrightarrow\ & (1-T)\bar{L}_g - \bar{L}_l > 0 \\ \Leftrightarrow\ & (1-T)\frac{(N_g - N_l)\bar{L}_\Delta + N_l\bar{L}_l}{N_g} - \bar{L}_l > 0 \\ \Leftrightarrow\ & \underbrace{(1-T)(N_g - N_l)}_{=:c_{\Delta,+}}\bar{L}_\Delta > \underbrace{(N_g - (1-T)N_l)}_{=:c_{l,+}}\bar{L}_l \end{aligned}$$

and by an analog computation

$$\frac{\bar{L}_g - \bar{L}_l}{\bar{L}_g} < -T \ \Leftrightarrow\ \underbrace{(1+T)(N_l - N_g)}_{=:c_{l,-}}\bar{L}_\Delta > \underbrace{((1+T)N_l - N_g)}_{=:c_{l,-}}\bar{L}_l.$$

Either way, we care about the probability $\mathrm{P}(c_{\Delta,\pm}\bar{L}_\Delta > c_{l,\pm}\bar{L}_l)$, which is the probability that a normal-distributed random variable exceeds an independent normal-distributed random variable.

Downton [Dow73] provide a minimum-variance unbiased estimate for this probability based on a finite sample like the one that we are working with. To apply Downton's formula for large sample counts [Dow73, Eq. 30], we need the sample mean and its sample variance for $k \in \{\Delta, l\}$:

$$\bar{x}_{k,\pm} := c_{k,\pm}\frac{b_{1,k}}{b_{0,k}}, \qquad s_{k,\pm}^2 := c_{k,\pm}^2 \frac{b_{0,k}b_{2,k} - b_{1,k}^2}{b_{0,k}^3}.$$

Then Downton's minimum-variance unbiased estimate of $\mathrm{P}(c_{\Delta,\pm}\bar{L}_\Delta > c_{l,\pm}\bar{L}_l)$ is

$$\Phi\left(\frac{\bar{x}_{\Delta,\pm} - \bar{x}_{l,\pm}}{\sqrt{s_{\Delta,\pm}^2 + s_{l,\pm}^2}}\right), \quad \text{where } \Phi(x) := \frac{1}{2}\left(\operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) + 1\right)$$

is the cumulative distribution function (CDF) of a standard normal random variable. We decide to split if both cells have at least 1000 samples (Sec. 4.1) and

$$\frac{\bar{x}_{\Delta,+} - \bar{x}_{l,+}}{\sqrt{s_{\Delta,+}^2 + s_{l,+}^2}} > \Phi^{-1}(1-\alpha) \ \text{ or } \ \frac{\bar{x}_{\Delta,-} - \bar{x}_{l,-}}{\sqrt{s_{\Delta,-}^2 + s_{l,-}^2}} > \Phi^{-1}(1-\alpha).$$

This criterion is efficient to evaluate and principled.

Appendix B: Concentration Parameter of the Mean Direction

Without loss of generality, we work in a coordinate frame where the mean direction is the z-axis. Then we can quantify the width of a vMF distribution by relating its standard deviation along the x-axis to its mean along the z-axis:

$$\sigma(\kappa_l^\star) := \frac{\sqrt{\int_{\mathbb{S}^2} \omega_\mathrm{x}^2 p_l^\star(\omega)\,\mathrm{d}\omega}}{\int_{\mathbb{S}^2} \omega_\mathrm{z} p_l^\star(\omega)\,\mathrm{d}\omega} = \frac{1}{\sqrt{\kappa_l^\star \coth\kappa_l^\star - 1}}$$

We have solved these integrals using a computer algebra system. For the weighted sum $\bar{\omega}_l$, the standard deviation and mean turn into

$$\frac{\sqrt{\int_{\mathbb{S}^2} \sum_{k=0}^{N_l-1} L_k^2 \omega_\mathrm{x}^2 p_l^\star(\omega)\,\mathrm{d}\omega}}{\int_{\mathbb{S}^2} \sum_{k=0}^{N_l-1} L_k \omega_\mathrm{z} p_l^\star(\omega)\,\mathrm{d}\omega} = \frac{\sqrt{\sum_{k=0}^{N_l-1} L_k^2}}{\sum_{k=0}^{N_l-1} L_k}\sigma(\kappa_l^\star) = \frac{\sqrt{b_{2,l}}}{b_{1,l}}\sigma(\kappa_l^\star).$$

Thus, the concentration parameter κ_l of the vMF distribution that models the spread in the mean direction estimate should satisfy

$$\sigma(\kappa_l) = \frac{\sqrt{b_{2,l}}}{b_{1,l}}\sigma(\kappa_l^\star) \quad \Leftrightarrow \quad \kappa_l \coth\kappa_l = \frac{b_{1,l}^2}{b_{2,l}}\sigma^{-2}(\kappa_l^\star) + 1 = y.$$

To solve this equation, we have constructed the fit in Eq. 4 manually, using non-linear optimization to find the parameters. It achieves a maximal relative error in κ_l of $1.46 \cdot 10^{-4}$ across all possible values of y (that was the objective being minimized).

Appendix C: Effective Concentration Parameter

We now prove that the probability $\mathrm{P}(v_g \cdot v_l < \cos\tau) > 1 - \alpha$ only depends on $\kappa_g^{-1} + \kappa_l^{-1}$, not on κ_g, κ_l individually. Let p_g, p_l denote the density of v_g, v_l, respectively. Furthermore, we let

$$\chi(z) := \begin{cases} 1 & \text{if } z < \cos\tau, \\ 0 & \text{otherwise.} \end{cases}$$

Then the sought-after probability is:

$$\begin{aligned} &\mathrm{P}(v_g \cdot v_l < \cos\tau) = \mathrm{E}(\chi(v_g \cdot v_l)) \\ &= \int \chi(\omega_g \cdot \omega_l) p_g(\omega_g)\,\mathrm{d}\omega_g p_l(\omega_l)\,\mathrm{d}\omega_l \end{aligned} \tag{5}$$

The inner integral can be understood as spherical convolution between a spherical cap of opening angle τ and p_g.

We now represent our functions using spherical harmonics (SH) basis functions $Y_{k,m}(\omega)$ where $k \in \mathbb{N}_0$ is the band index and $m \in \{-k,\dots,k\}$ is the order. We denote the SH coefficients of p_g, p_l by $y_{g,k,m}, y_{l,k,m}$, respectively. Furthermore, we denote the zonal SH coefficients of the spherical cap by $y_{c,k}$. Then the SH coefficients of the signal after the spherical convolution are [Slo08, p. 6]

$$\sqrt{\frac{4\pi}{2k+1}} y_{c,k} y_{g,k,m}.$$

The outer integral in Eq. 5 is an inner product of the output of the spherical convolution and p_l and thus an inner product of the corresponding SH coefficients:

$$\mathrm{P}(v_g \cdot v_l < \cos\tau) = \sum_{k=0}^{\infty}\sum_{m=-k}^{k} \sqrt{\frac{4\pi}{2k+1}} y_{c,k} y_{g,k,m} y_{l,k,m}.$$

Now we apply an approximate formula for the SH coefficients of vMF distributions, which is provably accurate for large κ_g, κ_l [VHM*22]:

$$y_{g,k,m} \approx \exp\left(-\frac{(k+1)k}{2\kappa_g}\right) Y_{k,m}(\mu_g)$$

and analogously for $y_{l,k,m}$. With this approximation, the probability $\mathrm{P}(v_g \cdot v_l < \cos\tau)$ simplifies to

$$\sum_{k=0}^{\infty} \exp\left(-\frac{(k+1)k}{2}\left(\frac{1}{\kappa_g} + \frac{1}{\kappa_l}\right)\right) \sum_{m=-k}^{k} \sqrt{\frac{4\pi}{2k+1}} y_{c,k} Y_{k,m}(\mu_g) Y_{k,m}(\mu_l).$$

It is clear that this only depends on $\kappa_g^{-1} + \kappa_l^{-1}$, not on κ_g, κ_l individually. Using κ, ∞ in place of κ_g, κ_l gives the same value for $\kappa_g^{-1} + \kappa_l^{-1}$.

Appendix D: Test Scenes

Pool: The Pool scene contains simple and challenging light transport. The directly lit area around the pool and the shadows from the tree can already be resolved efficiently with a standard path tracer using NEE. The challenging regions are the caustics inside the pool, in front of and inside the house generated by the glass windows. These caustics require path guiding to be resolved efficiently.

DiningRoom: The DiningRoom scene is an indoor scene with the back wall directly lit by a highly-directional environment light from the right window, indirectly illuminating the rest of the room. It features relatively uniform indirect illumination on the walls and complex geometries such as chairs and teaware.

LivingRoom: The LivingRoom scene is mostly illuminated indirectly by the bright spot on the couch on the left. Direct light only illuminates the scene through the small gap in the curtains. Due to the complicated illumination, the number of paths that sample any light contribution is low, even with NEE, resulting in a small number of samples used to train and update the guiding structure after each rendering iteration. The scene also contains multiple objects, such as the table, the TV, and the bookshelf, which generate occlusions and high variations in the scene's incident radiance.

CountryKitchen: The CountryKitchen scene combines direct illumination through the window from an environment map and indirect illumination from a large caustic on the wooden floor. While direct illumination can be efficiently resolved via NEE, indirect illumination requires path guiding first to direct the paths towards the caustic and then to resolve the caustic afterwards.

Head: The Head scene simulates subsurface scatter via a dense, high-albedo, isotropic volume, where a spot light generates strong self-shadowing effects. Here, path guiding helps to redirect paths entering the volume to quickly find their way back towards the closest, unshadowed surface. Inside the volume, the high density leads to multiple absorption and scattering events that rapidly change the mean radiance and the incident radiance distribution as we further enter the volume. An optimal subdivision adapts to these changes, especially around the shadows and inside the volume.