Practical Product Path Guiding
Using Linearly Transformed Cosines

Stavros Diolatzis[1], Adrien Gruson[2], Wenzel Jakob[3], Derek Nowrouzezahrai[2] and George Drettakis[1]

[1] Université Côte d'Azur, Inria
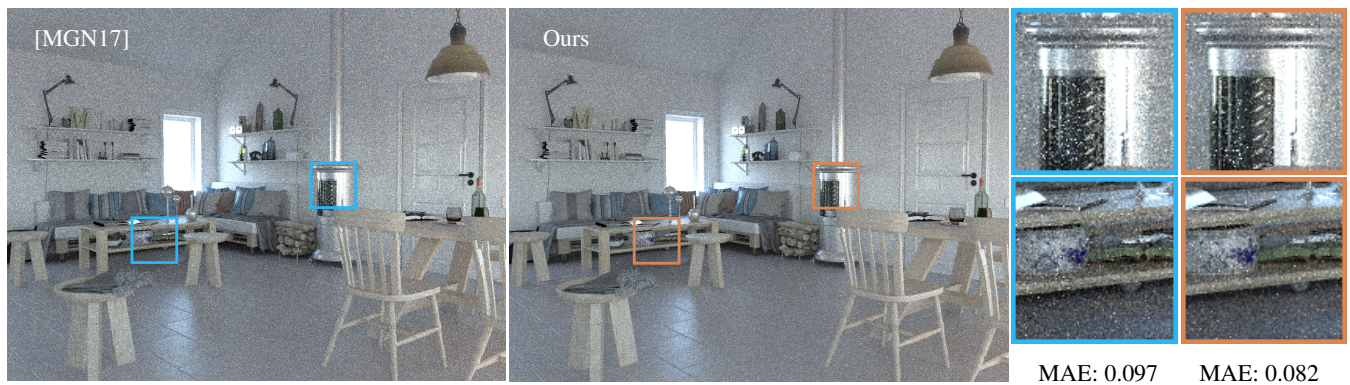[2] McGill University
[3] EPFL

Figure 1: *Equal-time comparisons (2 minutes). Our method reduces noise compared to Müller et al. [MGN17], by efficiently guiding paths based on the product of the BSDF and incoming radiance at each path vertex. We show Mean Absolute Error (MAE).*

Abstract

Path tracing is now the standard method used to generate realistic imagery in many domains, e.g., film, special effects, architecture etc. Path guiding has recently emerged as a powerful strategy to counter the notoriously long computation times required to render such images. We present a practical path guiding algorithm that performs product sampling, i.e., samples proportional to the product of the bidirectional scattering distribution function (BSDF) and incoming radiance. We use a spatial-directional subdivision to represent incoming radiance, and introduce the use of Linearly Transformed Cosines (LTCs) to represent the BSDF during path guiding, thus enabling efficient product sampling. Despite the computational efficiency of LTCs, several optimizations are needed to make our method cost effective. In particular, we show how we can use vectorization, precomputation, as well as strategies to optimize multiple importance sampling and Russian roulette to improve performance. We evaluate our method on several scenes, demonstrating consistent improvement in efficiency compared to previous work, especially in scenes with significant glossy inter-reflection.

Keywords: ray tracing, global illumination, spatial subdivision

1. Introduction

Path tracing is now a standard tool in the film and visual effects industry [KFF*15]. Keller et al. note in their survey that, despite the development of many efficient variants (e.g., [JM12, VG95a, GKDS12, HPJ12]), "simple" path tracing is often still the method of choice in many production settings. This is due to its reliability and lack of parameters to be tweaked, beyond the number of samples

per pixel. In recent years, *path guiding* has emerged as a powerful acceleration strategy that complements path tracing in challenging scenarios, and it is also finding its way into production [VHH*19].

Initial path guiding algorithms involved an expensive learning phase, and the relatively expensive evaluation reduced overall gains [VKŠ*14]. The *practical path guiding* technique of Müller et al. [MGN17] provides a simple yet effective solution using adaptive spatial-directional subdivision to guide paths during rendering based on an estimate of incoming radiance. Product path guiding

additionally takes the BSDF into account during guiding [HEV*16] in a learning-based framework. Product-based sampling increases sampling efficiency, but also involves expensive precomputation inhertited from the method it builds on [VKŠ*14]. We present a new approach that combines the computational benefits of practical path guiding and the sampling efficiency of *product* importance sampling.

Similar to Müller et al. [MGN17], our technique relies on a spatial-directional tree (SD-tree), to represent a continuously updated estimate of incoming radiance. In prior work, this data structure provided estimates of radiance integrals over spherical sets that were required to drive a hierarchical sampling scheme. However, it remains unclear how such an approach could generalize to the product case: direct tabulation of the product of BSDF and incident radiance is clearly infeasible due to the prohibitive increase in dimensionality from 5D ($\mathcal{V} \times \mathcal{S}^2$) to 7D ($\mathcal{V} \times \mathcal{S}^2 \times \mathcal{S}^2$), where $\mathcal{V}$ represents positions and $\mathcal{S}^2$ directions. Hierarchical sampling techniques that approximate the product at each level of the data structure [CJAMJ05] seem more promising but require estimates of BSDF integrals over spherical sets, which are not generally available in closed form. While these integrals could be estimated numerically, the resulting costs would likely diminish the benefits of product guiding. Our technique addresses this problem by providing an efficient approximation of the necessary integrals using Linearly Transformed Cosines (LTCs). Our choice is motivated by the observation that LTCs have been shown to be efficient for similar integrals required in the context of shading for polygonal lights [HDHN16].

Directly using LTCs for practical product sampling requires many integral evaluations during the hierarchical sample warping process, which unfortunately tends to negate the benefits of product guiding. To overcome this problem we introduce two main optimizations, one based on parallelisation and the other on precomputation. In addition, we show how to further improve results by carefully using multiple importance sampling and Russian roulette. We achieve on average 15-20% increase in computation speed for the same quality compared to previous work for our set of test scenes, both with practical guiding [MGN17] and learning-based product sampling [HEV*16].

2. Related Work

Importance sampling is a powerful and general technique for reducing the variance of Monte Carlo estimators by drawing samples from a distribution that approximates the integrand. In rendering, sampling typically targets the factors of the rendering equation [Kaj86], i.e., the incoming radiance L_i, cosine-weighted BSDF $\rho \cos\theta_i$ or some combination of the two:

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_\Omega L_i(x, \omega_i) \rho(x, \omega_o, \omega_i) \cos\theta_i \, d\omega_i, \quad (1)$$

where L_o is outgoing radiance in direction ω_o towards an observer, and the integral is over the hemisphere of directions Ω around the point x, and θ_i is the angle between the normal n at that point and direction ω_i. Different strategies have been developed to achieve

the goal of reducing variance when performing stochastic integration of Eq. 1; we review only the most relevant literature to our method.

Multiple Importance Sampling. Veach and Guibas [VG95b] introduced multiple importance sampling (MIS) to combine multiple sampling techniques (e.g., strategies that approximate different factors of the integrand). Using the balance or power heuristic to combine BSDF sampling with approximations of L_i, such as emitter sampling, is a common technique in rendering. However for some parts of the integral neither emitter nor BSDF sampling strategies are ideal, e.g., glossy surfaces with complex visibility, and in these cases multiple importance sampling is not effective. Recent work has proposed techniques to overcome this limitation, e.g., using optimization [KVG*19, GGSK19]. In our method we approximate and sample the product of the cosine-weighted BSDF and incoming radiance, in the context of a *path guiding* approach; we discuss corresponding previous methods next.

Path Guiding. Although sampling based on the local cosine-weighted BSDF $\rho \cos\theta_i$ has been used since the conception of Monte Carlo rendering, doing so proportionally to the incoming radiance L_i poses numerous challenges. Jensen [Jen95] populated the scene with photons in a preprocessing stage and used density estimation to approximate incoming radiance. More recently, Vorba et al.'s method [VKŠ*14] iteratively learns directional distributions of incoming radiance represented by Gaussian Mixture Models (GMMs) in a training step. These mixture models are used to guide the next iteration of importon/photon tracing while training and finally when rendering. However, the training step is computationally expensive and it can be hard to evaluate how much training is required, i.e., whether the current distributions are sufficiently converged. Reibold et al. [RHJD18] propose the use of an outlier rejection algorithm to determine paths in a scene with high variance and apply guiding using GMMs only to those paths. In this way they use expensive guiding only where necessary.

Another approach to overcome the computational expense was proposed by Müller et al. [MGN17], who use an SD-tree to efficiently approximate the incoming radiance field. The SD-tree consists of a binary tree for the 3D spatial subdivision and a quadtree for 2D directional variation, reminiscent of the 5D structure used for variance reduction with control variates in early Monte Carlo rendering [LW95]. In this method learning is achieved by repeatedly rendering the scene with exponentially more samples. It uses a one sample model [VG95b] with selection probability α to combine sampling based on the BSDF and the SD-tree, resulting in significant quality improvements with low computation overhead. This approach was extended by Müller [Mül19] in several ways: First, an optimization step to choose between BSDF and SD-tree sampling. Second, instead of discarding the previous iteration's images, they are weighted based on an approximated variance and combined in the final image. Finally, both in the spatial and directional domain the recording was improved from nearest neighbor to splatting. We build on the SD-tree approach for our solution, and also use these enhancements. Dahm and Keller [DK17] used a similar data-structure but with a different update policy based on Q-learning. Path guiding has also been expressed in the primary
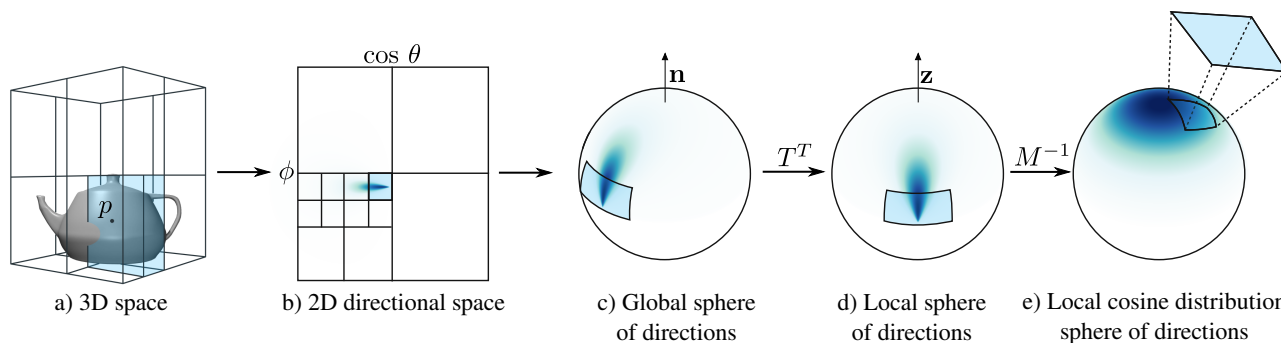
Figure 2: *Given an intersection point p we find the voxel in the a) spatial subdivision that includes it and the corresponding b) directional quadtree. This node spans a region of directions in the c) global sphere of directions with surface normal **n**. Using the transformation T^T we transform this spherical polygon to the d) local sphere of directions with normal the axis **z**. Finally the LTC linear transformation M^{-1} takes the polygon to the e) local cosine distribution sphere of directions.*

sample space [GBBE18]. Recent methods [MMR*19, ZZ19] employ deep neural networks to learn the scene dependent product distribution; however, the overhead of training these networks is still too high to make these solutions competitive in most cases.

Importance sampling based on the *product* of incoming radiance and the BSDF was originally used for direct illumination [CAM08, CJAMJ05]. The product was also used to select cosine lobes used to represent the incident radiance field for indirect illumination computation [BRDC12], or with spherical harmonics to importance sample environment maps [JCJ09]. The GMM approach of Vorba et al. [VKŠ*14] was extended to compute product GMM importance sampling [HEV*16]. The method also represents irradiance with a GMM to compute the product.

We build our approach on both the computational efficiency of practical path guiding [MGN17] that uses an SD-tree to represent incoming radiance, and the sampling efficiency of product path guiding [HEV*16]. The key challenge we address lies in striking the right balance between computational efficiency and sampling quality to improve overall performance. We do this by complementing the SD-tree structure with a LTC-based BSDF representation [HDHN16]. We furthermore introduce several important optimizations to make this approach powerful and effective.

3. Practical Product Path Guiding

We use the SD-tree structure of practical path guiding [MGN17] for our efficient product guiding. This 5D spatial-directional tree is partitioned as shown in Fig. 2. Each node of the spatial subdivision tree (a) contains a quadtree which is stored in 2D directional space (b), parameterized by $\cos\theta$ and ϕ. Each node of the quadtree can be thought of as a spherical polygon in the global sphere of directions with surface normal **n**. In the original method [MGN17] these nodes record the total incident radiance $L_i(x, \omega_i)$ at each iteration of the guiding process. This incident radiance is then used to sample directions in the next iteration. Sampling relies on a hierarchical warping scheme that requires recursive estimation of probabilities while descending in the quadtree of directions [MH97].

The main challenge when sampling the product

$L_i(x, \omega_i)\rho(x, \omega_o, \omega_i)\cos\theta_i$, taking into account both the cosine-weighted BSDF and discretization of L_i, is that the BSDF varies with respect to position and outgoing direction ω_o. This implies that the sampling distribution must be re-generated at every shading point. Tabulation and normalization of the product distribution further require costly evaluation of the product at the finest level of the SD-tree.

We introduce a separability approximation to make this process more practical. In particular, we assume that

$$\int_{\Omega_i} L_i(x, \omega_i)\rho(x, \omega_o, \omega_i)\cos\theta_i\mathrm{d}\omega_i \approx \left[\int_{\Omega_i} L_i(x, \omega_i)\mathrm{d}\omega_i\right] \cdot$$
$$\left[\int_{\Omega_i} \rho(x, \omega_o, \omega_i)\cos\theta_i\mathrm{d}\omega_i\right]$$

within spherical polygons Ω_i. This expression is approximate in particular when Ω_i covers a large solid angle, and it becomes more accurate under refinement. Our method samples this expression hierarchically in a coarse-to-fine manner, requiring many evaluations of spherical integrals over the BSDF, of the form

$$D = \int_{\Omega} \rho(x, \omega_o, \omega_i)\cos\theta_i\mathrm{d}\omega_i, \qquad (2)$$

hence we seek an efficient approximation. Naturally, *too* approximate of an estimate may even increase variance, thus a suitable tradeoff between performance and accuracy is key.

Three possible options to accomplish this could be analytic integration, a numerical solution or a conservative estimation of the integral. Analytic solutions exist only for cosine-like distributions, which would limit us to diffuse materials. Numerical approaches, such as Monte Carlo integration, would be too slow to yield accurate results since they require many samples for each bounce of the path. A conservative estimate such as the one used by Estevez and Lecocq [CEL18] would be problematic for the size of the spherical polygons, up to half a hemisphere, in the first levels of the quadtree. Instead, we use Linearly Transformed Cosines [HDHN16] which enable an analytical solution for more complex distributions, and have been demonstrated to be efficient for the integrals of the form of Eq. (2). We achieve this by transforming from a local sphere of

directions Fig. 2(d) to a local cosine distribution sphere of directions (e).

Although LTCs admit a cheap integration scheme over spherical sets, the recursive nature of our method increases overhead and diminishes gains from product sampling in practice. We first discuss how we perform the product path guiding, and then present optimization strategies that exploit the parallelizable nature of the computations and the precomputation of frequently used factors. Finally we discuss how multiple importance sampling and Russian roulette can be used to further improve results.

3.1. LTC-based Product Sampling

We next discuss LTC fitting for the BSDF, discuss our product sampling approach and the technical specifics required for LTC-based product sampling.

LTC fitting for the BSDF at a shading point. Most realistic materials can be represented with a mixture of diffuse and glossy components, with varying roughness. For the diffuse components we use an LTC with an identity transformation $M = I$. For the glossy components we precompute a table of fitted LTCs over varying roughness α and incoming elevation angle θ, with 128 bins for each. Given a ray intersection point p, the BSDF at that intersection point with roughness α and the outgoing direction elevation angle θ we fetch the corresponding LTC.

The LTC is stored in local space; a linear transformation M that defines the LTC takes points from the local cosine distribution to the local current BSDF distribution (Fig. 2(e) to (d)). Since the quadtrees store incoming radiance in global coordinates we need to apply the transformation T, a rotation of the axis, to take the LTC from local to global coordinates (Fig. 2(d) to (c)). As a result, in our representation, the LTC has a new linear transformation $M' = TM$.

To integrate the BSDF over a spherical polygon using the LTC, we apply the inverse linear transformation $(M')^{-1} = M^{-1}T^T$ to the vertices of the spherical polygon and analytically integrate the cosine distribution over it.

The integration of the BSDF using a fitted LTC uses the closed form expression described by Baum et al. [BRW89]:

$$E(p_1, ..., p_n) = \frac{1}{2\pi} \sum_{i=1}^{n} \cos^{-1}(\langle p_i, p_j \rangle) \langle \frac{p_i \times p_j}{p_i \cdot p_j}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \rangle \quad (3)$$

where p_i are the polygon vertices. This expression assumes that the integration domain lies within the upper hemisphere. Otherwise, it must be clipped over the horizon.

Product sampling. During path generation, given a path vertex p we descend in the binary spatial tree and find the corresponding directional quadtree (Fig. 2(a) to (b)). Similar to Müller et al. [MGN17], we use a hierarchical sample warping scheme proposed by McCool and Harwood [MH97] to generate samples that follow the desired distribution. During traversal, we estimate the probability to move to a child node k of a given internal node, and repeat this recursively until we reach a leaf node. Within the leaf node we sample uniformly.

In Müller et al.'s method each node stores estimated incoming radiance $\hat{L}_i$. At a given level of the quadtree, their sampling algorithm chooses between the four sibling nodes $v_j, j = 1..4$. Each such node k stores the incoming radiance $\hat{L}_i^k$, and the relative value $\frac{\hat{L}_i^k}{\sum_{j=1}^4 \hat{L}_i^j}$ determines the probability to move to it next. Sampled directions thus follow the incoming radiance $L_i(x, \omega_i)$ component of the integrand (Eq. 1). For the other component of the integrand, BSDF sampling generates directions following $\rho(x, \omega_o, \omega_i) \cos\theta_i$ and the two strategies are combined using MIS.

Our algorithm is summarized in Algorithm 1. Our path guiding method directly takes the product of the cosine-weighted BSDF into account. We thus need to compute the probability to descend into part of the quadtree based on the product of the BSDF integral D and the incoming radiance, using the LTCs for fast integration of D.

Technical specifics for LTC-based product sampling. The LTC representation described above allows us to integrate $\rho(x, \omega_o, \omega_i) \cdot \cos\theta_i$ over the four spherical polygons at each quadtree level.

During the hierarchical sampling process, the probability to move to a given child in the quadtree is given by the product of these values $D_j, j = 1..4$ and the stored values $\hat{L}_i^j, j = 1..4$. For child k, the probability is:

$$P_k = \frac{D_k \hat{L}_i^k(x, \omega_i)}{\sum_{j=1}^4 D_j \hat{L}_i^j(x, \omega_i)}. \quad (4)$$

where $\hat{L}_i^k$ is an estimate of the corresponding spherical integral of incoming radiance. This probability is used in the traversal of the sample warping scheme, thus generating samples that follow the full (product) integrand.

Discussion. Since the SD-tree stores the incoming radiance in global spherical coordinates over a spatial subdivision, the normals are averaged over this space. As a result sampled directions can be in the wrong hemisphere, terminating path generation. With product sampling this only happens when the leaf node lies on the horizon and the uniform sample generated within falls below the horizon, which is quite rare. In addition there is a chance that the product of the BSDF and the incoming radiance is 0 over the entire sphere of directions. In such cases we sample according to the BSDF since this means that no incoming radiance has been recorded in the local hemisphere for this surface normal.

3.2. Optimization Strategies

In practice, using LTC fitting during recursive sampling adds computational overhead that tends to negate the benefits of product path guiding. The specific computation required for our new method enables two optimizations: First, we can evaluate per-node integrals in parallel using vectorization. Second, common factors can be precomputed. These two optimizations improve the performance of our approach compared to previous methods, as we show in the results (Sec. 4).

Algorithm 1: Quadtree product sampling

---

if ISNODELEAF() **then**
  | UNIFORMLYSAMPLEWITHINNODE()
end
$\Pi \leftarrow 0$;
for $i \leftarrow 1$ to 4 **do**
  | $v \leftarrow$ GETQUADVERTICES();
  | **for** $j \leftarrow 1$ to 4 **do**
  | | $v_j \leftarrow$ CANONICALTOCARTESIAN(v_j);
  | | $v_j \leftarrow (M')^{-1} * v_j$;
  | | NORMALIZE(v_j);
  | **end**
  | $D_i \leftarrow$ LTCINTEGRATEQUAD(v);
  | $L_i \leftarrow$ GETSTOREDRADIANCESUM(i);
  | $\Pi_i \leftarrow D_i * L_i$;
  | $\Pi \leftarrow \Pi + \Pi_i$;
end
$P_k \leftarrow \frac{\Pi_k}{\Pi}, k = 1, ..., 4$;
CHOOSENODEWITHPROBABILITIES(P)

3.2.1. Parallel Processing

Clipping the polygon against the horizon results in a variable number of vertices (3 to 5) and requires branching code which impacts performance. Hill and Heitz [HH16] propose an approximation of this process by using the vector form factor of the unclipped polygon, i.e., Eq. 3 without the *z* axis dot product:

$$\mathcal{F} = E(p_1, ..., p_n) = \frac{1}{2\pi} \sum_{i=1}^{n} \cos^{-1}(\langle p_i, p_j \rangle) \qquad (5)$$

From F we can compute the angular extent and elevation angle of a sphere that has the same form factor as the unclipped polygon. We use the precomputed ratio of the clipped sphere's form factor to the unclipped one to scale the polygon's form factor accordingly. With this approximation, the BSDF integration requires Eq. 2 to be evaluated four times for each node (once for each pair of vertices), for four child nodes at each level (Algorithm 1). This computation represents the additional overhead compared to the sampling in Müller et al. [MGN17], and is a good candidate for vectorization. We perform these 16 computations at once on an AVX512 enabled CPU. The rest of the process involves fetching the stored $\hat{L}_i^j$ values so no further vectorization was possible there. Note that any optimization to other parts of the method from Müller et al. [MGN17] would also benefit our solution.

3.2.2. Precomputation

Another way to reduce the overhead of the product sampling is by precomputing the diffuse vector form factors $\mathcal{F}$ (see Eq. 5 above). When a material is diffuse the corresponding LTC has $M = I$ so the total inverse linear transformation applied to the quad vertices is given by: $(M')^{-1} = T^T$. T is an orthogonal transformation and as such we can apply it to the resulting vector form factor instead of applying it to the vertices and then doing the computation. Given this observation, we precompute and store the vector form factors $\mathcal{F}$ for the five first levels of a quadtree with a total memory foot-

print of 256KB. For these levels we can avoid the arc cosine, dot and cross product in Equation 2. This increases performance for all diffuse and multiple component materials with a diffuse component.

3.2.3. Discussion

Heitz et al. [HDHN16] mention that LTCs do not approximate the target distribution well in cases of incident grazing angles and high roughness materials. To avoid fireflies due to error in such cases we switch to path guiding without the BSDF LTC integration. Specifically we do this for cases of outgoing directions with local elevation angle $85° \geq \theta \geq 90°$. Moreover, if the material roughness is above a 0.5 threshold, we conservatively treat the material as diffuse when performing the product computation.
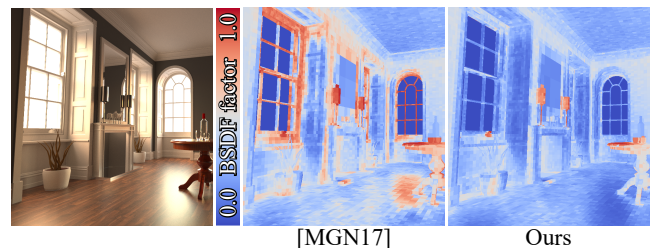

Figure 3: *The proportion of BSDF sampling learned with ADAM for our product sampling and for Müller et al. [MGN17]. Red regions show part of the scene where the guiding is used less than BSDF sampling. Our approach is more robust because it considers the product, thus lowering usage of BSDF sampling.*

Figure 4: *The six scenes used in our tests. From left to right: Bathroom, Living Room, Glossy Kitchen, Pink Kitchen, Attic and Necklace.*
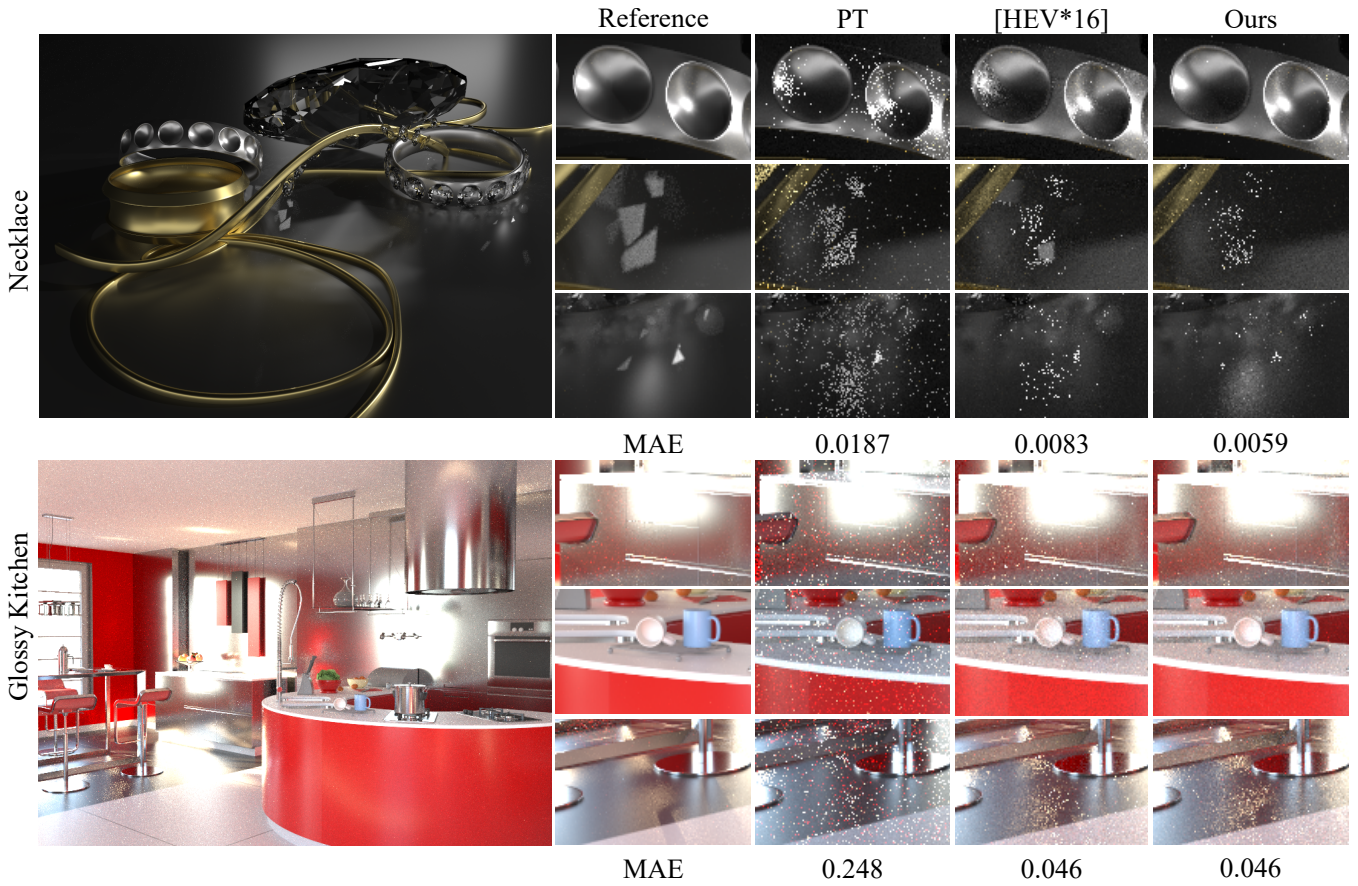
Figure 5: *Equal-time comparison between standard uni-directional path tracing (PT), product GMM [HEV*16] and our product for the Necklace (5 minutes) and Glossy Kitchen (10 minutes). Due to the online nature and more robust radiance representation, our technique can generate images with lower noise. However, the GMM product can be better at capturing fine lighting details due to its high directional resolution. The training time for GMM is 61 and 194 seconds respectively.*

3.3. Optimization of MIS

Our product path guiding depends on the accuracy of the incoming radiance representation and of the LTC integration. In some cases (e.g., very shiny materials and/or grazing angles), these representations may not provide the best result. To overcome this issue, we combine our method with BSDF sampling using multiple importance sampling (MIS) as a defensive sampling strategy [HEV*16, MMR*19].

We use the approach of Müller et al. [Mül19], that learns the probability to either sample the BSDF or path guide based on the performance of each sampling technique. Specifically, we run an optimization step to learn the α value (see below), for the sampling probability $p(\omega_o|x,\omega_i)$ for the outgoing direction ω_o from position x and incoming direction ω_i

$$p(\omega_o|x,\omega_i) = (1-\alpha)p_g(\omega_o|x,\omega_i) + \alpha p_{bsdf}(\omega|x,\omega_i) \quad (6)$$

where p_g is the probability defined by our product guiding, and p_{bsdf} is the BSDF sampling probability. Note that for Müller et al.'s approach, p_g does not consider the incoming direction ω_i during sampling. We use the methodology of Müller et al. [MMR*19] to

find α, i.e., we use the ADAM optimizer with the same parameters and optimize the Kullback-Leibler (KL) divergence as it is more robust to outliers [MMR*19].

It is interesting to visualize the learned α value for our product method compared to the original practical path guiding [Mül19] (Figure 3). Observe that our α values are lower compared to Müller et al.'s, which indicates that guiding reduces the need for BSDF sampling as a defensive strategy. In particular, we rely more on our path guiding strategy on glossy surfaces (i.e., wooden floor) or in regions with normal variations (i.e., thin objects like the windows' frames). This is visible as more blue in the figure, indicating that BSDF sampling is used less in these regions with our approach.

3.4. Russian Roulette

As observed in previous work [HEV*16], product sampling increases average path length. This is because with product sampling we do not generate paths towards the light source if the BSDF value (and thus the path contribution) are low, in contrast to previous methods that only take incoming radiance into account. In these
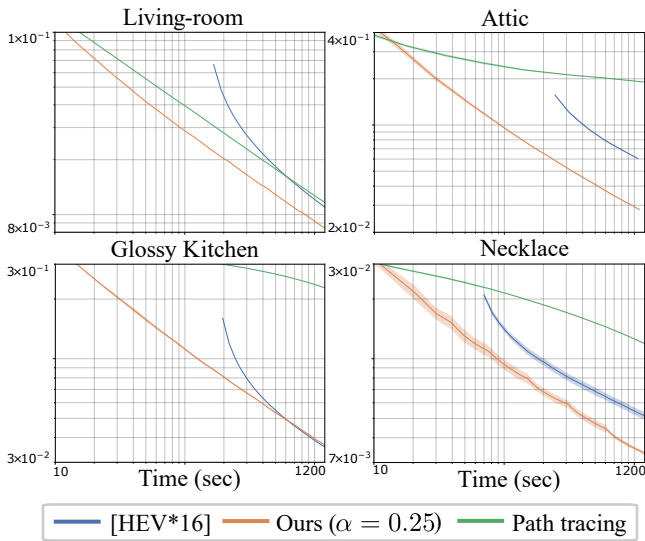
Figure 6: *L_1 convergence graph for 4 scenes between product GMM [HEV*16] and our product without Russian roulette and fixed BSDF sampling probability. We show the average and variance over 5 runs of each technique.*

Table 1: *Sampling cost of generating* 64 *sample per pixel for different scenes. These timings includes all costs, which includes ray intersection and our guiding procedure. Our optimized version reduces the sampling cost by around* 30−35 % *making our technique practical.*

Scene	Naive	Optimized	Optimized AVX
Living-room	26.30	20.92 (79 %)	16.54 (62 %)
Pink Kitchen	53.60	44.22 (82 %)	32.94 (61 %)
Attic	64.04	54.47 (85 %)	41.93 (65 %)
Bathroom	58.28	48.13 (82 %)	36.91 (76 %)
Glossy Kitchen	20.12	17.90 (88 %)	13.69 (68 %)
Necklace	12.79	11.84 (92 %)	8.95 (69 %)

cases, our technique introduces a tradeoff between path length and higher contribution. In practice, we perform Russian roulette for all paths on length two and higher, using Adjoint-driven Russian roulette [VK16] (without splitting) where pixel estimates are directly stored in the spatial binary tree nodes. This approach was originally proposed by Müller et al. [MGN17].

4. Results and Evaluation

We implemented our method in the Mitsuba [Jak10] renderer, and used the Enoki library [Jak19] for parallelization with AVX-512 acceleration. We will provide the source code of our method, including the Enoki optimization, for open research use.

We use L_1 difference for the metrics shown in the main paper. For other metrics, please refer to the additional material. We choose this specific metric as it less prone to overweight fireflies compared to square error metrics such as L_2. Guiding methods may increase variance in important but undiscovered areas. To remove these few remaining fireflies we could use an outlier removal method [ZHD18].

We ran evaluations on a set of six test scenes shown in Fig. 4: Bathroom, Living Room, Glossy Kitchen, Pink Kitchen, Attic and Necklace. Some of these are variants of scenes used in previous work [MGN17, HEV*16]. We were unable to compare with GMM [HEV*16] on Pink Kitchen and Bathroom due to specific issues with materials in these scenes. All reference images are computed with several hours of computation by averaging several independent runs of practical path guiding or by high sample count path tracing. We set maximum path length to 10. To generate the results for comparisons, the authors have kindly provided their own implementations of the corresponding methods. All results are generated

with implementations in Mitsuba [Jak10], with 40 threads on a dual Intel Gold 6148 Skylake at 2.4Ghz, with dual AVX-512 units.

We disabled next-event estimation for all the techniques. Next-event estimation can be an ineffective sampling strategy in scenes with a highly occluded light source if no importance cache is used; this is the case for most of our test scenes. More generally, guiding techniques store the direct illumination directly inside the cache. Moreover, storing the direct and indirect illumination when doing the product guiding has the advantage of taking the BSDF at the shading point into account.

As our technique uses online learning, we combine all the iterations using the inverse variance scheme [Mül19]. We also use a box filter and stochastic filtering when splatting a contribution on the directional and spatial data structure respectively.

We first present statistics illustrating the contribution of each of our optimizations to the efficiency of our technique. We discuss our experiments with an alternative to LTCs. We then compare to the product GMM [HEV*16] approach and to practical path guiding [MGN17]. We also compare to unguided path tracing to illustrate which part of the light transport is difficult to sample.

Product optimization. We summarize the performance results of our technique with our different optimization strategies in Table 1 for various scenes. The naive version uses accurate polygon clipping and no diffuse precomputation (Section 3.2.2). The optimized version uses all optimizations listed in Section 3.2, except for parallel processing. We used the Enoki library [Jak19] to achieve parallel processing using AVX-512 to compute the integrals at each level of the quad-tree in parallel. The scenes are ordered from the most diffuse to the most glossy one. For these results, no Russian roulette was used. We did not observe any noise increase when using the sphere form factor approximation to make our code branchless. Note that using AVX-512 instruction is a crucial optimization to making the technique even more computationally effective.

LTC vs Monte Carlo integration. An alternative to LTCs for the BSDF integration is a Monte Carlo approach, which we experimented with. To avoid repeating the integral estimation process for each node during traversal we created a temporary quadtree which
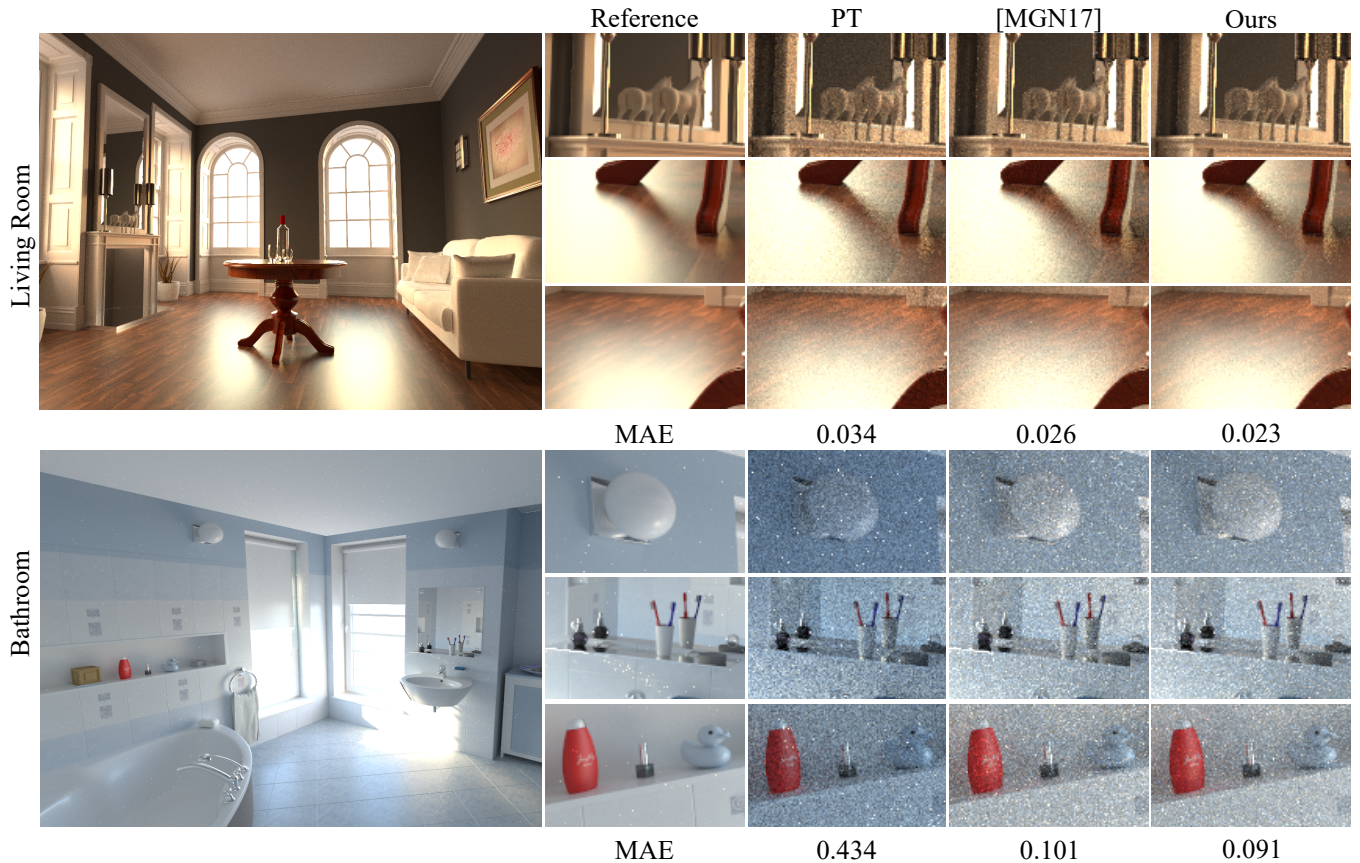
Figure 7: *Equal-time comparison between standard uni-directional path tracing (PT), Müller et al. [MGN17] and our product for the Living Room (2 minutes) and Bathroom (3 minutes) scenes.*

we filled with *N* equal energy samples using BSDF sampling. We added a constant 5% of the total energy to all the nodes to ensure we explored nodes that, due to the low number of samples, haven't received any energy. The two quadtrees, temporary BSDF quadtree and incoming radiance quadtree, shared the same structure and the product of their values determined the traversal. For $N = 64$ and for same quality results the Monte Carlo integration approach was 9 times slower than the LTC approach.

Comparison with product GMM [HEV*16]. To achieve a fair comparison, we do not use Russian roulette since it was not available in the reference implementation of the product GMM method. We also set the BSDF sampling probability to 0.25 for both the techniques. As also noticed by Müller et al. [MGN17] the GMM training implementation does not scale up with a high number of threads, while the other two methods do. To provide a fair comparison we trained radiance GMM with 8 threads and assumed perfect linear scaling of speedup to 40 threads to take training time into account. We do 30 training passes with 2M photons or importons emitted per pass. We use default parameters for the rest of the algorithm.

Figure 5 shows the comparison of Product GMM [HEV*16] and our technique. Due to the online nature which uses all the samples

combining different iterations using inverse variance and a more robust irradiance representation, our algorithm can perform more samples per pixel and achieve lower error. Still GMMs are usually better at capturing fine highlights like caustics, due to a denser cache. However, due to GMM Expectation Maximization instability, GMM techniques can generate artefacts in some regions, like on the sliver ring in Necklace scene. L_1 convergence graph are shown in Figure 6. Overall our method is more efficient, however for the Glossy Kitchen scene performance is similar or better at later iterations.

Comparison with Practical Path Guiding [MGN17]. We compared our product guiding with Müller et al. [MGN17] by enabling all the improvements presented in [Mül19]. Here all the BSDF improvements (Section 3.3) and ADRRS (Section 3.4) are enabled.

We also present graphs of error convergence for the three methods shown in Figure 8 for the six test scenes. We see that our method almost always has lower error, converging faster than the previous solutions. On average, we are 15% faster for the same quality. Figure 7 shows equal-time comparison for two scenes.

Overall, we observed that our method is particularly helpful in two cases: (1) on glossy surfaces where our guiding reduces noise since the effect of the product is more pronounced than elsewhere
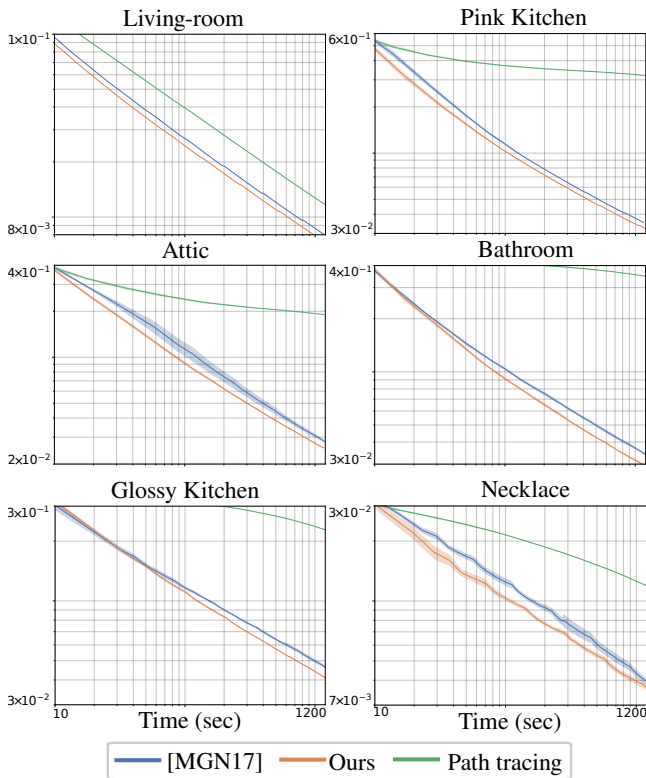
Figure 9: *An artificial failure case for our separability approximation. a) The integral of the product of these two functions is zero but our approximation by a product of integrals b) gives a non zero value.*

cases, e.g., for deterministic sampling techniques such as glass material. However, the treatment of such light interactions, e.g., the decision to reflect or refract could result in paths with high contribution where guiding could be beneficial. It is unclear how to adapt our data structures to effectively guide such sampling decisions, without storing the complete path [RHJD18].

Our separability approximation introduced in Sec. 3 performs adequately in our test scenes but in theory it still has failure cases. We could construct such a failure case with two checkerboard functions illustrated in Fig. 9. A similar scenario could arise with very complex materials with multiple glossy lobes that don't overlap with the incoming radiance in some directions. In that case our method would overestimate the product value and allocate samples in regions of low importance.

Finally, one key element for the efficiency of our approach is the use of LTC to integrate the BSDF contribution over a node of the quadtree. This works well for some materials such as the GGX model we used in our tests, but the current LTC fitting procedure may need to be adapted for other models. In addition, the expense of our product approach is proportional to the number of LTCs needed to integrate for a given BSDF, making the treatment of very complex materials more challenging.

5.2. Future Work

In future work, we would like to further investigate the interaction between MIS and path guiding. For now we are using ADAM to optimize the BSDF selection probability. However, this BSDF selection probability is given for a spatial cell and does not take into account the incoming direction. It will be interesting to investigate if a more elaborate approach providing a finer BSDF selection probability can give better results. Finally, it is not clear how to incorporate such an optimization procedure with recent MIS techniques [GGSK19, KŠV*19].

Recent techniques restrict guiding to regions where necessary by storing complete paths [RHJD18], in contrast to a cache of all paths such as the SD-tree we use. Developing a method that combines the ability of the former to treat very hard paths and the full path expressivity of the latter is an exciting direction for future work.

Figure 8: *L_1 convergence graph for our six scenes, compared to [MGN17] and simple path tracing. We show the average and variance over 5 runs of each technique.*

(2) on diffuse surfaces where we can clamp the irrelevant directions. Overall, even if our computation is more expensive (due to the multiple LTC integrations), we almost always see improvement in the level of noise, compared to previous work.

5. Limitations and Future Work

Our method shows improvement in most of the scenes we tested, with more significant gains for scenes with many glossy/rough materials. It is however not without limitations that we discuss next, followed by directions for future work.

5.1. Limitations

For scenes with moderate to high complexity in geometry and materials, our product guiding is generally advantageous. However, for some simple scenes the overhead of path guiding may not be worthwhile. This is especially true for product guiding that involves a significantly higher overhead than simple strategies such as BSDF sampling, even though the tradeoff needs to be considered for all guiding methods. Despite recent work (e.g., [RHJD18, BJ19]), there is currently no easy way to identify "difficult paths" for which product guiding is guaranteed to be cost effective.

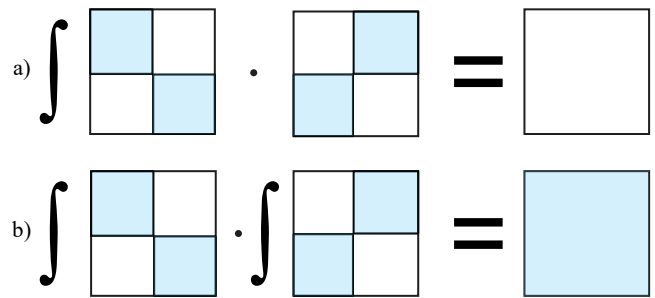In our current approach, guiding is not used in some specific

A possible future research direction would be to build a data structure based in primary sample space for sampling a point on the emitter given a position in space. However, to apply our product approach, it would be necessary to know the light source geometry in terms of polygonal shapes to apply LTC integration.

Finally, both practical and product path guiding could be used in the context of volume rendering, e.g., by adapting LTC integration to support phase functions.

6. Conclusions

We have presented a new product based path guiding technique, that combines the computational efficiency of spatial-directional subdivision and sampling efficacy of product guiding. The key element of our approach is the use of Linearly Transformed Cosines allowing efficient integration of the BSDF during hierarchical importance sampling, i.e., when recursively traversing the quadtree representation of the directional component of the subdivision. To make the approach cost effective, we introduce two main optimizations, using parallelization and precomputation, and also exploit the benefits of MIS and Russian roulette to further improve performance. We have demonstrated how our new approach is beneficial on a set of six test scenes, and we have also presented an analysis of the benefits of each of our optimizations.

Acknowledgments

References

[BJ19] BITTERLI B., JAROSZ W.: Selectively Metropolised Monte Carlo light transport simulation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) 38*, 6 (Nov. 2019). doi:10.1145/3355089.3356578. 9

[BRDC12] BASHFORD-ROGERS T., DEBATTISTA K., CHALMERS A.: A significance cache for accelerating global illumination. *Comput. Graph. Forum 31*, 6 (2012), 1837–1851. doi:10.1111/j.1467-8659.2012.02099.x. 3

[BRW89] BAUM D. R., RUSHMEIER H. E., WINGET J. M.: Improving radiosity solutions through the use of analytically determined form-factors. *ACM Siggraph Computer Graphics 23*, 3 (1989), 325–334. doi:10.1145/74334.74367. 4

[CAM08] CLARBERG P., AKENINE-MÖLLERY T.: Practical product importance sampling for direct illumination. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 681–690. doi:10.1111/j.1467-8659.2008.01166.x. 3

[CEL18] CONTY ESTEVEZ A., LECOCQ P.: Fast product importance sampling of environment maps. In *ACM SIGGRAPH 2018 Talks*. 2018, pp. 1–2. doi:10.1145/3214745.3214760. 3

[CJAMJ05] CLARBERG P., JAROSZ W., AKENINE-MÖLLER T., JENSEN H. W.: Wavelet importance sampling: efficiently evaluating products of complex functions. In *ACM SIGGRAPH 2005 Papers*. 2005, pp. 1166–1175. doi:10.1145/1073204.1073328. 2, 3

[DK17] DAHM K., KELLER A.: Learning light transport the reinforced way, 2017. arXiv:1701.07403. 2

[GBBE18] GUO J., BAUSZAT P., BIKKER J., EISEMANN E.: Primary sample space path guiding. In *Eurographics Symposium on Rendering* (2018), vol. 2018, The Eurographics Association, pp. 73–82. doi:10.2312/sre.20181174. 3

[GGSK19] GRITTMANN P., GEORGIEV I., SLUSALLEK P., KŘIVÁNEK J.: Variance-aware multiple importance sampling. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 1–9. doi:10.1145/3355089.3356515. 2, 9

[GKDS12] GEORGIEV I., KRIVÁNEK J., DAVIDOVIC T., SLUSALLEK P.: Light transport simulation with vertex connection and merging. *ACM Trans. Graph. 31*, 6 (2012), 192–1. doi:10.1145/2366145.2366211. 1

[HDHN16] HEITZ E., DUPUY J., HILL S., NEUBELT D.: Real-time polygonal-light shading with linearly transformed cosines. *ACM Transactions on Graphics (TOG) 35*, 4 (2016), 41. doi:10.1145/2897824.2925895. 2, 3, 5

[HEV*16] HERHOLZ S., ELEK O., VORBA J., LENSCH H., KŘIVÁNEK J.: Product importance sampling for light transport path guiding. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 67–77. doi:10.1111/cgf.12950. 2, 3, 6, 7, 8

[HH16] HILL S., HEITZ E.: Advances in real-time rendering. real-time area lighting: a journey from research to production. In *ACM SIGGRAPH 2016 Courses* (2016), SIGGRAPH 16. 5

[HPJ12] HACHISUKA T., PANTALEONI J., JENSEN H. W.: A path space extension for robust light transport simulation. *ACM Trans. Graph. 31*, 6 (2012). doi:10.1145/2366145.2366210. 1

[Jak10] JAKOB W.: Mitsuba renderer, 2010. URL: http://www.mitsuba-renderer.org. 7

[Jak19] JAKOB W.: Enoki: structured vectorization and differentiation on modern processor architectures, 2019. URL: https://github.com/mitsuba-renderer/enoki. 7

[JCJ09] JAROSZ W., CARR N. A., JENSEN H. W.: Importance sampling spherical harmonics. *Computer Graphics Forum (Proceedings of Eurographics) 28*, 2 (Apr. 2009). doi:10.1111/j.1467-8659.2009.01398.x. 3

[Jen95] JENSEN H. W.: Importance driven path tracing using the photon map. In *Rendering Techniques 95*. Springer, 1995, pp. 326–335. doi:10.1007/978-3-7091-9430-0_31. 2

[JM12] JAKOB W., MARSCHNER S.: Manifold exploration: a markov chain monte carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics (TOG) 31*, 4 (2012), 58. doi:10.1145/2185520.2185554. 1

[Kaj86] KAJIYA J. T.: The rendering equation. In *ACM SIGGRAPH computer graphics* (1986), vol. 20, ACM, pp. 143–150. doi:10.1145/15922.15902. 2

[KFF*15] KELLER A., FASCIONE L., FAJARDO M., GEORGIEV I., CHRISTENSEN P., HANIKA J., EISENACHER C., NICHOLS G.: The path tracing revolution in the movie industry. In *ACM SIGGRAPH 2015 Courses*. 2015, pp. 1–7. doi:10.1145/2776880.2792699. 1

[KŠV*19] KARLÍK O., ŠIK M., VÉVODA P., SKŘIVAN T., KŘIVÁNEK J.: Mis compensation: optimizing sampling techniques in multiple importance sampling. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 1–12. doi:10.1145/3355089.3356565. 9

[KVG*19] KONDAPANENI I., VÉVODA P., GRITTMANN P., SKŘIVAN T., SLUSALLEK P., KŘIVÁNEK J.: Optimal multiple importance sampling. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 1–14. doi:10.1145/3306346.3323009. 2

[LW95]   LAFORTUNE E. P., WILLEMS Y. D.: A 5D tree to reduce the variance of Monte Carlo ray tracing. In *Rendering Techniques* (1995), pp. 11–20. `doi:10.1007/978-3-7091-9430-0_2`. 2

[MGN17]   MÜLLER T., GROSS M., NOVÁK J.: Practical path guiding for efficient light-transport simulation. In *Computer Graphics Forum* (2017), vol. 36, Wiley Online Library, pp. 91–100. `doi:10.1111/cgf.13227`. 1, 2, 3, 4, 5, 7, 8, 9

[MH97]   MCCOOL M. D., HARWOOD P. K.: Probability trees. In *Graphics Interface* (1997), vol. 97, pp. 37–46. 3, 4

[MMR*19]   MÜLLER T., MCWILLIAMS B., ROUSSELLE F., GROSS M., NOVÁK J.: Neural importance sampling. *ACM Trans. Graph. 38*, 5 (2019). `doi:10.1145/3341156`. 3, 6

[Mül19]   MÜLLER T.: "practical path guiding" in production. In *ACM SIGGRAPH Courses: Path Guiding in Production, Chapter 10* (New York, NY, USA, 2019), ACM, pp. 18:35–18:48. `doi:10.1145/3305366.3328091`. 2, 6, 7, 8

[RHJD18]   REIBOLD F., HANIKA J., JUNG A., DACHSBACHER C.: Selective guided sampling with complete light transport paths. *ACM Transactions on Graphics (TOG) 37*, 6 (2018), 1–14. `doi:10.1145/3272127.3275030`. 2, 9

[VG95a]   VEACH E., GUIBAS L.: Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques*. Springer, 1995, pp. 145–167. `doi:10.1007/978-3-642-87825-1_11`. 1

[VG95b]   VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (1995), ACM, pp. 419–428. `doi:10.1145/218380.218498`. 2

[VHH*19]   VORBA J., HANIKA J., HERHOLZ S., MÜLLER T., KŘIVÁNEK J., KELLER A.: Path guiding in production. In *ACM SIGGRAPH 2019 Courses*. 2019, pp. 1–77. `doi:10.1145/3305366.3328091`. 1

[VK16]   VORBA J., KŘIVÁNEK J.: Adjoint-driven russian roulette and splitting in light transport simulation. *ACM Transactions on Graphics (TOG) 35*, 4 (2016), 1–11. `doi:10.1145/2897824.2925912`. 7

[VKŠ*14]   VORBA J., KARLÍK O., ŠIK M., RITSCHEL T., KŘIVÁNEK J.: On-line learning of parametric mixture models for light transport simulation. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 101. `doi:10.1145/2601097.2601203`. 1, 2, 3

[ZHD18]   ZIRR T., HANIKA J., DACHSBACHER C.: Re-weighting firefly samples for improved finite-sample monte carlo estimates. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 410–421. `doi:10.1111/cgf.13335`. 7

[ZZ19]   ZHENG Q., ZWICKER M.: Learning to importance sample in primary sample space. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 169–179. `doi:10.1111/cgf.13628`. 3