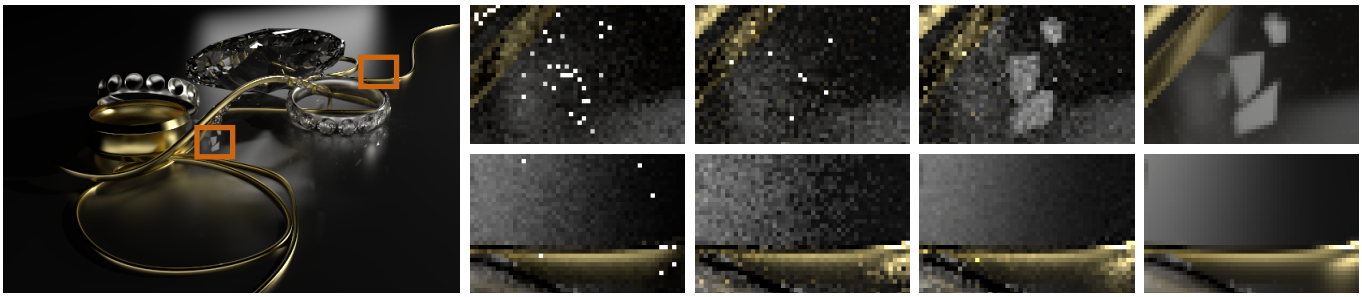
Variance-Aware Path Guiding

ALEXANDER RATH, DFKI and Saarland University, Germany
PASCAL GRITTMANN, Saarland University, Germany
SEBASTIAN HERHOLZ, University of Tübingen, Germany and Intel Corporation
PETR VÉVODA, Charles University, Prague and Chaos Czech a.s., Czech Republic
PHILIPP SLUSALLEK, DFKI and Saarland University, Germany
JAROSLAV KŘIVÁNEK, Charles University, Prague and Chaos Czech a.s., Czech Republic



Necklace

(a) Path tracer
89.9

(b) Radiance-based
0.4 (baseline)

(c) Our target density
0.16 (2.6x)

(d) Reference
relMSE

Fig. 1. The guiding approach of Müller et al. [2017] benefits greatly from our target densities, e.g., for caustics on glossy surfaces, as shown here. Our method consists of a trivial modification applicable to a variety of path guiding algorithms without additional parameters or computational overhead.

Path guiding is a promising tool to improve the performance of path tracing algorithms. However, not much research has investigated what target densities a guiding method should strive to learn for optimal performance. Instead, most previous work pursues the zero-variance goal: The local decisions are guided under the assumption that all other decisions along the random walk will be sampled perfectly. In practice, however, many decisions are poorly guided, or not guided at all. Furthermore, learned distributions are often marginalized, e.g., by neglecting the BSDF. We present a generic procedure to derive theoretically optimal target densities for local path guiding. These densities account for variance in nested estimators, and marginalize provably well over, e.g., the BSDF. We apply our theory in two state-of-the-art rendering applications: a path guiding solution for unidirectional path tracing [Müller et al. 2017] and a guiding method for light source selection for the many lights problem [Vévoda et al. 2018]. In both cases, we observe significant improvements, especially on glossy surfaces. The implementations for both applications consist of trivial modifications to the original code base, without introducing any additional overhead.

Authors' addresses: Alexander Rath, alexander.rath@dfki.de, DFKI, Saarland University, Saarbrücken, Germany; Pascal Grittmann, Saarland University, Saarbrücken, Germany, grittmann@cg.uni-saarland.de; Sebastian Herholz, University of Tübingen, Tübingen, Germany, Intel Corporation, Sebastian.Herholz@intel.com; Petr Vévoda, Charles University, Prague, Chaos Czech a.s., Czech Republic, petrvevoda@seznam.cz; Philipp Slusallek, DFKI, Saarland University, Saarbrücken, Germany, philipp.slusallek@dfki.de; Jaroslav Křivánek, Charles University, Prague, Chaos Czech a.s., Czech Republic.

0730-0301/2020/7-ART1 $15.00
https://doi.org/10.1145/3386569.3392441

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

Additional Key Words and Phrases: global illumination, importance sampling, path guiding

ACM Reference Format:
Alexander Rath, Pascal Grittmann, Sebastian Herholz, Petr Vévoda, Philipp Slusallek, and Jaroslav Křivánek. 2020. Variance-Aware Path Guiding. *ACM Trans. Graph.* 39, 4, Article 1 (July 2020), 12 pages. https://doi.org/10.1145/3386569.3392441

1 INTRODUCTION

The majority of rendering systems today rely on unidirectional path tracers [Burley et al. 2018; Fascione et al. 2018; Georgiev et al. 2018; Keller et al. 2015]. The simplicity, flexibility, and extensibility of the algorithm is what makes it so appealing. The performance, however, depends heavily on the employed importance sampling strategy. Ideally, paths should be sampled proportionally to their pixel contribution. Unfortunately, computing that ideal distribution is a harder problem than rendering the image, because it would require knowledge of the full light transport in the scene. Hence, many implementations construct paths by locally sampling from coarse approximations, like BSDF importance sampling.

Path guiding methods learn better importance sampling densities, either locally or for full paths, based on information gathered from previous rendering iterations [Vorba et al. 2019]. The learned densities are then used to importance sample paths in future iterations.

Learning the optimal sampling density for a complete path is often infeasible, due to the high dimensionality [Müller et al. 2018; Zheng and Zwicker 2019]. Alternatively, it is theoretically possible to construct an optimal path with only local decisions. To achieve

that goal, however, every single local decision needs to be guided perfectly. The optimal local decision for zero-variance sampling is proportional to the product of the BSDF and the incident radiance. Hence, this high-dimensional distribution would need to be either learned, which is expensive, or computed on-the-fly at every intersection point, which is also expensive. In practice, that usually means that zero-variance sampling cannot be achieved.

In this work, we show how to derive optimal target densities when zero-variance sampling cannot be achieved. Our target densities minimize the image error due to the local sampling decisions. In effect, more samples are invested towards directions that cause high variance, to adaptively reduce that variance. Furthermore, our method can learn lower-dimensional densities that marginalize provably well. Thus, the BSDF is accounted for in a provably good manner, without full product sampling, rather than ignoring it completely as is common in previous work.

From a practical perspective, our method achieves visible improvements with trivial changes to the implementation. An example is shown in Fig. 1. Here, our target density greatly improves the caustic on the glossy surface by marginalizing over the BSDF and, at the same time, investing more samples towards the directions that cause most variance in the estimate.

2 PREVIOUS WORK

Path tracing. The path tracing algorithm computes light transport via Monte Carlo integration [Kajiya 1986]. In its simplest and most common form, paths are traced backwards from the sensor through the scene, until they reach an emitter. At each intersection with the scene, a local sampling decision is made to decide in which direction to continue the path. This decision is usually based on importance sampling the BSDF and cosine term [Pharr et al. 2016]. Additionally, next-event estimation is often performed to connect the path vertices directly to points on the light sources.

Bidirectional methods. Complex effects, like strong indirect illumination, are sometimes better handled by also tracing paths starting at the emitters [Veach 1997]. These bidirectional methods either connect the resulting paths via shadow rays [Lafortune and Willems 1993; Veach and Guibas 1995a] or perform density estimation [Georgiev et al. 2012a; Hachisuka et al. 2008, 2012; Jensen 1996]. While bidirectional methods perform well for some hard cases, they tend to be less efficient overall, especially for large scenes with many light sources. It is possible to reduce the overhead by limiting the bidirectional techniques to those effects that benefit from them the most, like caustics [Grittmann et al. 2018]. Still, the additional implementation complexity and strict requirement of physically accurate models often makes bidirectional methods less appealing in practice [Fascione et al. 2018; Georgiev et al. 2018].

Local path guiding. The efficiency of the path tracing algorithm, whether it is bi- or unidirectional, hinges on the local importance sampling of directions to continue the path. Unfortunately, the simple approach of importance sampling only the BSDF and cosine term performs poorly in most realistic scenes (e.g., due to complex visibility). The goal of local path guiding methods is to build better distributions that take an approximation of the incident radiance into account. Existing methods mostly differ in terms of data structures and what samples they use for training. A simple approach is to build histograms based on a photon map [Jensen 1995]. Others use tree structures [Bus and Boubekeur 2017; Lafortune and Willems 1995; Müller et al. 2017; Pegoraro et al. 2008], particle footprints [Hey and Purgathofer 2002], parametric mixture models [Vorba et al. 2014], or even neural networks [Bako et al. 2019; Müller et al. 2017]. Some of these approaches only approximate the incident radiance. Others compute the product with the BSDF on-the-fly, which is expensive if done accurately [Bashford-Rogers et al. 2012; Herholz et al. 2018, 2016; Hey and Purgathofer 2002; Jensen 1995; Lafortune and Willems 1995]. Local guiding methods can also be extended to participating media, where even more random decisions have to be guided [Herholz et al. 2019]. Zero variance sampling can be achieved if every local decision is perfect. Unfortunately, this would require perfect representations, full product sampling, and sampling conditionally on many terms (e.g., position, outgoing direction, and wavelength), which is not always possible due to practical limitations. In this work, we show how to design local target densities that compensate for such practical limitations.

Light selection. Practical scenes often feature a large number of light sources. While the direct illumination from those lights can be learned as a hemispherical distribution, as local path guiding methods would do, it is often more efficient to explicitly sample the lights. Guiding can also be employed for light source selection [Georgiev et al. 2012b; Pantaleoni 2019; Vévoda et al. 2018]. Conceptually, these methods differ from local path guiding only in that they learn discrete distributions and limit themselves to direct illumination. Hence, we also apply our theory in that context.

Global path guiding. An alternative to guiding the local decisions is to learn distributions that sample full paths. This can be achieved, for instance, by sampling paths similar to a set of previously selected guiding paths [Reibold et al. 2018], or by guiding in primary sample space [Guo et al. 2018; Müller et al. 2018; Zheng and Zwicker 2019]. Unfortunately, such global guiding suffers from the curse of dimensionality and is often only practical for relatively short paths.

Metropolis light transport. Global path guiding methods are somewhat similar to Markov chain Monte Carlo (MCMC) methods [Šik and Křivánek 2018; Veach and Guibas 1997]. MCMC methods also attempt to sample full paths according to some target distribution. The difference is that rather than learning a distribution, they mutate the previously sampled paths. Hence, a major problem of MCMC methods is that the paths are highly correlated and the chain can get stuck in local extrema, resulting in visible artefacts and poor temporal coherence. Nevertheless, MCMC methods have been successfully applied in a bidirectional setting, where the correlation artefacts can be avoided by sampling the camera paths with regular Monte Carlo [Šik and Křivánek 2019; Šik et al. 2016].

Target densities. Many successful MCMC methods do not use chains that converge to the integrand. Instead, the chains are often made to converge to a different function: the target function. These functions are usually designed to be easier to explore by the Markov chain [Hachisuka and Jensen 2011], to adaptively sample regions of high error [Gruson et al. 2017], or to perform better in

combination with a regular path tracer [Šik et al. 2016]. The idea of learning anything but (a portion of) the rendering equation integrand has received little attention in existing local path guiding methods. Vévoda et al. [2018] learn how to importance sample light clusters for next event estimation, while accounting for the variance of the unguided sampling within the chosen cluster. We show how to derive provably optimal target densities for applications where zero-variance sampling cannot be achieved. The result of Vévoda et al. can be re-derived as a discrete special case of our theory, without incorporating the BSDF. Pantaleoni and Heitz [2017] show how a given target density can be optimally approximated, an orthogonal improvement that can be trivially applied to our target densities.

3 BACKGROUND

Rendering equation. Light transport is described by the rendering equation [Kajiya 1986]:

$$L_\mathrm{o}(\omega_\mathrm{o}, x) = L_\mathrm{e}(\omega_\mathrm{o}, x) + \int_\Omega L_\mathrm{i}(\omega_\mathrm{i}, x) B(\omega_\mathrm{o}, x, \omega_\mathrm{i}) |\cos\theta_\mathrm{i}| \,\mathrm{d}\omega_\mathrm{i}, \quad (1)$$

where L_o is the outgoing radiance from point x in direction ω_o, which is given by the emission L_e and the reflected radiance. The reflected radiance is computed by integrating over all directions ω_i and computing the incident radiance L_i recursively, as the outgoing radiance from visible points. The BSDF B models how the surface reflects light from ω_i to ω_o.

Monte Carlo integration. The integral in the rendering equation can be estimated by evaluating the integrand for one direction ω_i, sampled at random [Pharr et al. 2016; Veach 1997]:

$$L_\mathrm{o} \approx \langle L_\mathrm{o} \rangle = L_\mathrm{e} + \frac{\langle L_\mathrm{i} \rangle B |\cos\theta_\mathrm{i}|}{p(\omega_\mathrm{i} \mid x, \omega_\mathrm{o})}, \quad (2)$$

where we dropped the arguments for brevity. Dividing by the probability density function (PDF) of the random direction ω_i ensures that the expected value of the estimator is the desired integral: $\mathbb{E}\left[\langle L_\mathrm{o} \rangle\right] = L_\mathrm{o}$. Usually, the L_i term in the integrand needs to be estimated recursively; resulting in a random walk through the scene, sampling one direction ω_i at a time – the path tracing algorithm.

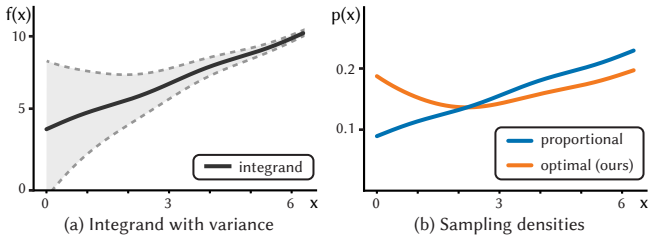
Estimation error. The error in MC integration is governed by the estimator's variance, the deviation of the second moment from the squared ground truth [Veach 1997]:

$$\mathbb{V}\left[\langle L_\mathrm{o} \rangle\right] = \mathbb{E}\left[\langle L_\mathrm{o} \rangle^2\right] - L_\mathrm{o}^2. \quad (3)$$

Since the ground truth is constant, we can limit our analysis on the second moment $\mathbb{E}\left[\langle L_\mathrm{o} \rangle^2\right]$. The variance is zero if the PDF is exactly proportional to the integrand, hence the 'zero-variance' goal of previous guiding methods is to approximate that PDF as closely as possible. This, however, can only succeed if the recursive $\langle L_\mathrm{i} \rangle$ is also sampled perfectly, as we will discuss in the next section.

4 TARGET DENSITIES FOR LOCAL PATH GUIDING

In this section, we present a generic approach to derive optimal target densities. At first, we assume that only a single decision along the random walk can be guided. We start with optimal target densities to guide that one local decision at an exact surface point x. We show how to account for variance in nested estimators and marginalization over the outgoing direction in that setting. Then, we derive the local target density that minimizes the average error in the rendered image. Lastly, we show how to extend the results to the typical case where densities are learned for regions of space S, rather than exact points x.



(a) Integrand with variance (b) Sampling densities

Fig. 2. (a) An integrand (black line) and the variance of its nested estimator (shaded region) for a hypothetical integration problem $\int \langle f(x) \rangle \,\mathrm{d}x$. (b) Instead of importance sampling the ground truth shape of the integrand $f(x)$, our method invests more samples where the variance of the nested $\langle f(x) \rangle$ is high, resulting in provably better performance.

4.1 Adaptive densities: The irradiance integral

A common quantity in rendering is the irradiance $E(x)$ at a point x (e.g., to evaluate diffuse reflection at that point).

$$E(x) = \int_\Omega L_\mathrm{i}(\omega_\mathrm{i}, x) |\cos\theta_\mathrm{i}| \,\mathrm{d}\omega_\mathrm{i}. \quad (4)$$

The corresponding estimator typically is:

$$\langle E(x) \rangle = \frac{\langle L_\mathrm{i}(\omega_\mathrm{i}, x) \rangle |\cos\theta_\mathrm{i}|}{p(\omega_\mathrm{i} \mid x)}. \quad (5)$$

Here, for brevity, we assume that only a single sample for ω_i is taken. The incident radiance is computed via a nested MC estimator $\langle L_\mathrm{i} \rangle$.

Previous work usually sets the target function to the ground truth value of incoming radiance:

$$p(\omega_\mathrm{i} \mid x) \propto L_\mathrm{i}(\omega_\mathrm{i}, x) |\cos\theta_\mathrm{i}| = \mathbb{E}\left[\langle L_\mathrm{i}(\omega_\mathrm{i}, x) \rangle\right] |\cos\theta_\mathrm{i}|. \quad (6)$$

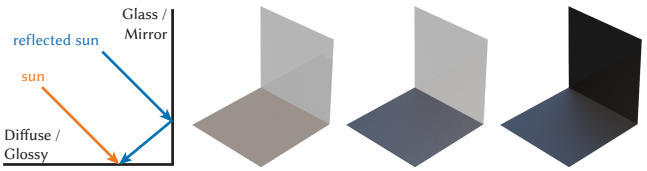
This, however, neglects variance in the nested $\langle L_\mathrm{i} \rangle$ estimation. Consider the illustrated example in Fig. 2. The shaded region around the black line visualizes the variance of the nested estimator. In the extreme case plotted here, the variance is highest where the integrand is lowest. Hence, sampling proportionally to the ground truth value performs poorly: The region of highest variance would receive the fewest samples.

We can find a better suited target density by minimizing the variance of the irradiance estimator,

$$\mathbb{V}\left[\langle E(x) \rangle\right] = \mathbb{E}\left[\langle E(x) \rangle^2\right] - E^2(x). \quad (7)$$

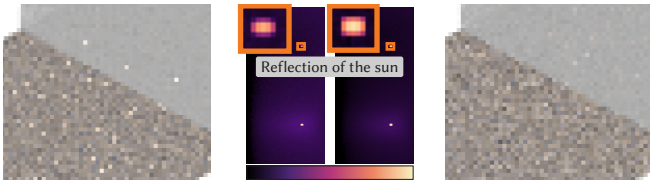
The free variable is the PDF $p(\omega_\mathrm{i} \mid x)$: For path guiding, we would like to find the best such PDF and approximate it based on training samples. Looking at (7), we can see that only the first term, $\mathbb{E}\left[\langle E(x) \rangle^2\right]$, the second moment, depends on the PDF. The squared ground truth value $E^2(x)$ is constant. The second moment is a convex functional in the PDF, so the minimizer can be found via Lagrange multipliers:

$$p_E(\omega_\mathrm{i} \mid x) = \underset{p(\omega_\mathrm{i} \mid x)}{\arg\min}\, \mathbb{E}\left[\langle E(x) \rangle^2\right] + \lambda \left(\int p(\omega_\mathrm{i}' \mid x) \,\mathrm{d}\omega_\mathrm{i}' - 1 \right), \quad (8)$$



(a) Cross-section (b) Diffuse / Glass (c) Glossy / Glass (d) Glossy / Mirror

Fig. 3. We evaluate the theory on simple test scenes: two perpendicular quads illuminated by a strong sun and uniform sky. The materials are varied for the three different stages of our theory (b-d).



(a) Radiance-based (b) Densities (c) Our target function

Fig. 4. Previous work (a) samples the reflection of the sun proportional to its radiance. Our method (c) compensates for the unguided 'reflect vs refract' decision on the glass with a higher density towards the reflected sun (b).

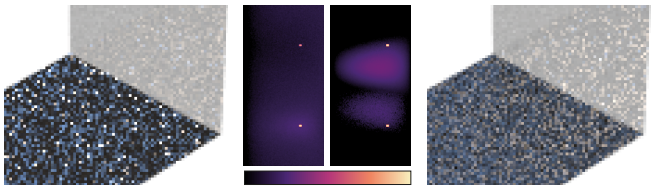
where λ is the Lagrange multiplier and ensures that $p(\omega_\mathrm{i} \mid x)$ integrates to one, i.e., is a valid PDF. The full derivation can be found in the supplemental document. The resulting target density is:

$$p_E(\omega_\mathrm{i} \mid x) \propto \sqrt{\mathbb{E}\left[\langle L_\mathrm{i}(\omega_\mathrm{i}, x)\rangle^2\right]}|\cos\theta_\mathrm{i}|. \tag{9}$$

That is, sampling should be proportional to the square root of the second moment of the nested radiance estimator, multiplied by the cosine term. If the nested estimator has no variance, the second moment is equal to the squared ground truth: $\mathbb{E}\left[\langle L_\mathrm{i}\rangle^2\right] = L_\mathrm{i}^2$. Then, our target function is equal to the one used in previous work (6). Otherwise, the nested estimator's variance is accounted for.

To test our target density, we apply it to a simple rendering problem: We guide the irradiance estimation on a perfectly diffuse, planar surface. The scene layout is illustrated in Fig. 3. In all cases, two perpendicular quads are illuminated by an environment map with a small, strong sun and a uniform sky. Since the direction of the direct illumination is invariant with the position, we build a single, high resolution histogram with a large number of samples, to closely approximate our target density. Then, a small number of samples is taken from the approximated target density.

In the first example, depicted in Fig. 4, the floor is diffuse, the wall made of perfectly smooth glass. Hence, the light on the diffuse surface comes mostly from two directions: directly from the sun, and from the sun's reflection in the glass. The latter requires an additional sampling decision of whether to reflect or refract on the glass. In practice, guiding on Dirac delta surfaces (e.g., glass or mirrors) is usually infeasible, since it requires an exact representation of the local, potentially high-frequent 4D light field. Therefore, the decision on the glass is not guided, relying on Fresnel term importance sampling instead. Unfortunately, that results in a large number of rays being refracted through the glass, never finding the bright sun.



(a) Radiance-based (b) Densities (c) Our target function

Fig. 5. A glossy floor is illuminated by the sun and the sun's reflection in a glass pane. Rather than ignoring the BSDF, we use a provably good marginalized distribution.

Our method invests more samples towards the sun's reflection in the glass, compensating for the nested variance.

4.2 Marginalized product sampling

The reflected light off a glossy surface is given by the rendering equation integral (1). For that case, the zero-variance distribution would be proportional to the product of incident radiance, BSDF, and cosine. Such a product distribution can be computed on-the-fly, given a distribution proportional to the BSDF and one for irradiance, represented in a suitable model (e.g., parametric mixtures) [Herholz et al. 2016]. Then, we would only need to learn our irradiance target function (9). Unfortunately, computing suitable representations for all types of BSDF models and their variations is not always feasible [Herholz et al. 2018, 2016].

We could still try to learn the zero-variance density $p(\omega_\mathrm{i} \mid x, \omega_\mathrm{o})$. That, however, would be a 7D density, possibly containing high frequencies. The accuracy of the fit, the required number of samples, and the overhead of the implementation can be greatly reduced by simplifying to a 5D density, conditional only on the position x and marginalized over the outgoing direction: $p(\omega_\mathrm{i} \mid x)$. Previous work has performed that simplification by ignoring the BSDF term [Vorba et al. 2019]. We, instead, derive an optimal marginalized target density.

The goal is to guide an estimator for the reflected radiance, with a PDF independent of the outgoing direction ω_o:

$$\langle L_\mathrm{o}(x, \omega_\mathrm{o})\rangle = \frac{B(\omega_\mathrm{i}, x, \omega_\mathrm{o})\langle L_\mathrm{i}(\omega_\mathrm{i}, x)\rangle|\cos\theta_\mathrm{i}|}{p(\omega_\mathrm{i} \mid x)}. \tag{10}$$

To find a suitable target distribution, we first need to define our optimization goal. One option is to minimize the expected error under a given distribution of outgoing directions ω_o:

$$p_{L_\mathrm{o}}(\omega_\mathrm{i} \mid x) = \underset{p(\omega_\mathrm{i}|x)}{\arg\min}\, \mathbb{E}_{\omega_\mathrm{o}}\left[\mathbb{E}\left[\langle L_\mathrm{o}(x, \omega_\mathrm{o})\rangle^2\right]\right] + \lambda(\dots). \tag{11}$$

Following the same steps as before, we arrive at the target density:

$$p_{L_\mathrm{o}}(\omega_\mathrm{i} \mid x) \propto \sqrt{\mathbb{E}\left[B^2(\omega_\mathrm{i}, x, \omega_\mathrm{o})\langle L_\mathrm{i}(\omega_\mathrm{i}, x)\rangle^2\right]}|\cos\theta_\mathrm{i}|. \tag{12}$$

The key differences to the irradiance target density (9) are that we average over all outgoing directions and include the squared BSDF.

Figure 5 shows the same simple scene as before, with the floor made glossy. Ignoring the BSDF, as done by previous work results in the exact same density as if the floor was diffuse. Our density (12), instead, allocates a significant amount of samples to the glossy reflection of the sky. Note, however, that the error in the reflection

increases. To reduce the overall error, our density trades a slight increase of noise in the glass for a significant improvement on the glossy surface.

While this target density is optimal regarding our chosen optimization goal (11), it is not optimal in a global sense. On the one hand, some glossy effects will still be better handled by BSDF importance sampling, e.g., almost specular reflections. On the other hand, the target density above optimizes for the *most frequent* directions ω_o, which might not be the *most important* ones. In Section 5 we discuss how to adapt the target density in an MIS combination, to avoid oversampling glossy effects that are better handled by BSDF importance sampling. But first, the next section will derive a density that accounts for the importance of different outgoing directions.

4.3 Minimizing the image error

In our discussion so far, we have neglected the image contribution of the local estimator. The target densities should, ideally, minimize the variance of every pixel. In the following, we first derive the optimal local target density if we were able to learn one density per pixel. Then, we show how to extend the derivation to arrive at a marginalized density, shared for all pixels.

Pixel contribution. To form our optimal local target densities, we need to consider the contribution of a point x to some pixel px. To compute that contribution, one has to consider every path $\bar{x}$ that leads from the pixel to the point x. The contribution is then the outgoing radiance at x multiplied by the throughput of the path $\bar{x}$, integrated over all such paths:

$$C_{\text{px}}(x) = \int_{P_x} f_{\text{px}}(\bar{x}) \langle L_o(x) \rangle \, \mathrm{d}\bar{x}. \tag{13}$$

Here, $\bar{x}$ is a path starting in the pixel and eventually arriving at the point x. We denote the space of all such paths as P_x. The contribution of the path to the pixel, $f_{\text{px}}(\bar{x})$, is the product of the sensor response and the path throughput. This notation allows us to minimize the image error by minimizing the error due to each individual point x.

Minimizing the pixel error. Our goal is to minimize the pixel variance due to local random sampling at the point x in the scene. Consider a pixel estimator that starts by sampling a path $\bar{x}$, starting in a pixel px and eventually arriving at point x (13). The second moment of that estimator is:

$$\mathbb{E}\left[\langle C_{\text{px}}(x) \rangle^2\right] = \underbrace{\int_{P_x} \frac{f_{\text{px}}^2(\bar{x})}{p(\bar{x})} \, \mathrm{d}\bar{x}}_{\text{pixel contribution}} \underbrace{\int_{\Omega} \frac{B^2 \cos^2 \theta_i}{p(\omega_i \mid x)} \mathbb{E}\left[\langle L_i \rangle^2\right] \, \mathrm{d}\omega_i}_{\text{local estimator}}, \tag{14}$$

where $p(\bar{x})$ is the joint probability of the random walk that leads to the point x. Minimizing this second moment yields:

$$p_{\text{px}}(\omega_i \mid x, \text{px}) \propto \sqrt{\int_{P_x} \frac{f_{\text{px}}^2(\bar{x})}{p(\bar{x})} B^2 \cos^2 \theta_i \, \mathbb{E}\left[\langle L_i \rangle^2\right] \, \mathrm{d}\bar{x}}. \tag{15}$$

Note the dependency on the pixel: To use this target density, we would have to learn one density for every pixel in the image. Next, we show how to marginalize over the pixels instead.

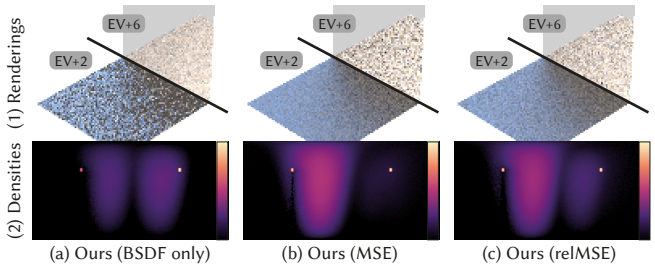


Fig. 6. Reflection of a glossy quad in a dark mirror (c.f., Fig. 3). The exposure value (EV) is unevenly adjusted over the image. (a) Considering only the frequency of ω_o assigns equal weight to both glossy lobes. (b) Minimizing the MSE favors the (originally) brighter pixels, i.e., the directly visible portion. (c) Minimizing the relMSE balances the noise between bright and dark regions, which is usually more desirable.

Minimizing the MSE of the image. It is usually infeasible to learn one density per pixel. One alternative is to minimize the mean variance over all pixels of the image. That is, instead of minimizing (14) for an individual pixel, we minimize the average over all pixels. Hence, the resulting local target density, derived as before, also contains the summation over all pixels:

$$p_{\text{MSE}}(\omega_i \mid x) \propto \sqrt{\sum_{\text{px}} \int_{P_x} \frac{f_{\text{px}}^2(\bar{x})}{p(\bar{x})} B^2 \cos^2 \theta_i \, \mathbb{E}\left[\langle L_i \rangle^2\right] \, \mathrm{d}\bar{x}}. \tag{16}$$

The result is an estimator that renders images with the lowest possible mean-squared error (MSE). However, the MSE is not always the best error metric for an image, because it scales quadratically with the pixel luminance. The target density would neglect darker pixels in favor of brighter ones, as the comparison in Fig. 6 shows. When replacing the glass pane in our simple example by a dark mirror, the glossy floor is visible from two sets of pixels: the ones that see it directly and the ones that see the dark reflection. Minimizing the MSE over the complete image focuses on the brighter pixels and neglects the reflection almost completely.

Minimizing the relMSE of the image. Instead of minimizing the MSE, we propose to minimize the relative MSE (relMSE): the MSE divided by the squared ground truth value of the pixel. The relMSE is independent of the pixel luminance, hence minimizing it achieves a more balanced level of noise. The only modification to our optimization is a division by a constant, the ground truth value I_{px}, which propagates into the target density, yielding:

$$p_{\text{relMSE}}(\omega_i \mid x) \propto \sqrt{\sum_{\text{px}} \int_{P_x} \frac{f_{\text{px}}^2(\bar{x})}{\tilde{I}_{\text{px}}^2 \, p(\bar{x})} B^2 \cos^2 \theta_i \, \mathbb{E}\left[\langle L_i \rangle^2\right] \, \mathrm{d}\bar{x}}. \tag{17}$$

The ground truth is, of course, unknown. It can, however, be approximated by denoising or aggressively filtering the image from previous training iterations, as done by Vorba and Křivánek [2016]. We denote the approximated pixel value as $\tilde{I}_{\text{px}}$, which also includes a small offset ϵ to avoid division by zero: $\tilde{I}_{\text{px}} \approx I_{\text{px}} + \epsilon$.

4.4 Spatial caches

In practice, guiding distributions are learned not for a specific point x but for a spatial cache cell S (e.g., from a grid or tree structure [Vorba et al. 2019]). Typically, the learned densities are averaged over all points $x \in S$. This averaging, however, results in low densities for directions that matter only to a small number of points.

A simple example is depicted in Fig. 7. A density $p(\omega_\mathrm{i} \mid S)$ is learned for each of four spatial cells S. There is no variance in the nested incident radiance estimate, and the target density is approximated with a high resolution histogram. Still, the rendering with previous work shows outliers along the boundaries of S.

The set of points x to which a direction ω_i contributes vanishes at the boundaries. In the limiting case, for a point on the boundary itself, there is a direction ω_i that needs to be sampled for that point, but for no other $x \in S$. When averaging across the whole spatial cell, the resulting density for such an ω_i is almost zero.

A better distribution can be found by minimizing the average error due to all points $x \in S$. For our simplest target density, the optimal one for the irradiance estimator, we change the minimization objective from (8) to:

$$p_S(\omega_\mathrm{i} \mid x \in S) = \operatorname*{arg\,min}_{p(\omega_\mathrm{i} \mid x \in S)} \mathbb{E}_{x \in S}\left[\mathbb{E}\left[\langle E(x)\rangle^2\right]\right] + \lambda(\dots). \tag{18}$$

This is analogous to the marginalization over ω_o in (11). The resulting target density is:

$$p_S(\omega_\mathrm{i} \mid x \in S) \propto \sqrt{\mathbb{E}_{x \in S}\left[\mathbb{E}\left[\langle L_\mathrm{i}(\omega_\mathrm{i}, x)\rangle^2\right]\cos^2\theta_\mathrm{i}\right]}. \tag{19}$$

Note that the square root operation is now done *after* averaging over S. Intuitively, this square root 'steepens' the fall-off for directions that contribute only to the boundary, preventing vanishing densities.

The derivation for the other two target densities is analogous, the resulting target density for the simple BSDF marginalization is:

$$p_\mathrm{simple}(\omega_\mathrm{i} \mid x \in S) \propto \sqrt{\mathbb{E}_{\omega_\mathrm{o}, x \in S}\left[B^2(\omega_\mathrm{i}, x, \omega_\mathrm{o})\,\mathbb{E}\left[\langle L_\mathrm{i}(\omega_\mathrm{i}, x)\rangle^2\right]\cos^2\theta_\mathrm{i}\right]}. \tag{20}$$

We refer to this as our 'simple' target density in the following sections. The target density to minimize the relative error is:

$$p_\mathrm{full}(\omega_\mathrm{i} \mid x \in S) \propto \sqrt{\sum_\mathrm{px} \mathbb{E}_{x \in S}\left[\int_{P_x} \frac{f_\mathrm{px}^2(\bar{x})}{\tilde{I}_\mathrm{px}^2 p(\bar{x})} B^2 \cos^2\theta_\mathrm{i}\,\mathbb{E}\left[\langle L_\mathrm{i}\rangle^2\right] \mathrm{d}\bar{x}\right]}. \tag{21}$$

We refer to this one as our 'full' target density in the following.

5 MULTIPLE IMPORTANCE SAMPLING

So far, we have discussed the variance of an estimator that uses only the learned sampling strategy. That, however, is insufficient in practice. Relying solely on learned densities can cause excessive variance, or bias [Owen and Zhou 2000]. The reasons include simplification of the integrand (e.g., no BSDF product), marginalization over important terms (e.g., outgoing direction or surface normal), and fitting a possibly inappropriate representation to noisy data. Therefore, it is common practice to combine the learned density with a conservative one, like BSDF importance sampling [Hey and Purgathofer 2002; Vorba et al. 2014].

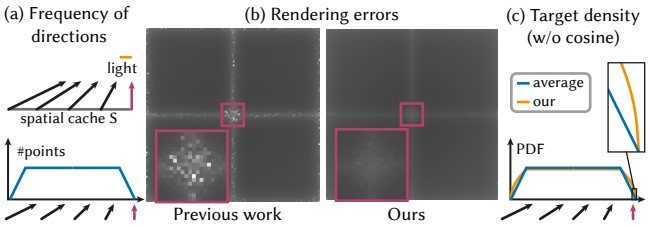


Fig. 7. Four directional distributions are learned on a diffuse plane illuminated by a small light source. (a) The points $x \in S$ see the light from different directions. We plot the number of points $x \in S$ (i.e., the surface area) to which each direction contributes. Directions at the boundaries of S are only relevant to a vanishingly small set of points. (b) Visualization of the rendering error, showing outliers at the boundaries. (c) Comparison of the PDFs when averaging (as in previous work) to our result. Our distribution steepens the fall-off at the boundary and eliminates the outliers.

The combination is usually done via one-sample MIS with the balance heuristic [Veach and Guibas 1995b], resulting in an estimator of the following form:

$$\langle L_\mathrm{o}\rangle_\mathrm{MIS} = \frac{\langle L_\mathrm{i}\rangle B|\cos\theta_\mathrm{i}|}{(1-\alpha)p_\mathrm{g}(\omega_\mathrm{i}) + \alpha p_B(\omega_\mathrm{i})}. \tag{22}$$

Here, p_g and p_B are the guiding and BSDF importance sampling distributions, respectively. First, one of the PDFs is chosen at random, where α is the probability of choosing BSDF importance sampling. Then, a direction ω_i is sampled according to the chosen PDF. In the more general case, if we allow not just the balance heuristic but arbitrary MIS weights w_g and w_B, the estimator is:

$$\langle L_\mathrm{o}\rangle_\mathrm{MIS} = \begin{cases} \frac{w_\mathrm{B}(\omega_\mathrm{i}, x, \omega_\mathrm{o})\langle L_\mathrm{i}\rangle B|\cos\theta_\mathrm{i}|}{p_\mathrm{B}(\omega_\mathrm{i}|x,\omega_\mathrm{o})\alpha(x,\omega_\mathrm{o})} & \text{, with prob. } \alpha(x,\omega_\mathrm{o}) \\ \frac{w_\mathrm{g}(\omega_\mathrm{i}, x, \omega_\mathrm{o})\langle L_\mathrm{i}\rangle B|\cos\theta_\mathrm{i}|}{p_\mathrm{g}(\omega_\mathrm{i}|x)(1-\alpha(x,\omega_\mathrm{o}))} & \text{, else.} \end{cases} \tag{23}$$

As shown here, the optimal selection probability $\alpha(x, \omega_\mathrm{o})$ generally depends on the position and outgoing direction.

In the following sections, we first show how to tune our guiding density to perform best in an MIS combination. Then, we revisit previous work on optimizing the selection probability α and demonstrate how insights from our theory can benefit that problem, too.

5.1 MIS compensation

Our target densities from Section 4 attempt to capture the full integrand. When combined with BSDF importance sampling via MIS, that is not always the best approach. Consider a case where a guiding cache spans a glossy and a diffuse surface, as shown in Fig. 8. Here, our target density (21) strikes a trade-off that minimizes the average error across both, increasing the noise on the diffuse surface to avoid outliers on the glossy one. In this example, however, the glossy reflection of the almost uniform sky is well handled by BSDF importance sampling. Hence, there is no need for our distribution to also cover that portion of the domain.

Instead of a target density that minimizes the error when used alone, we ideally want to learn the density that minimizes the error within an MIS combination. Finding the best such density has been recently proposed under the name of *MIS compensation* [Karlík et al. 2019]. The approach of Karlík et al. is to subtract a constant from

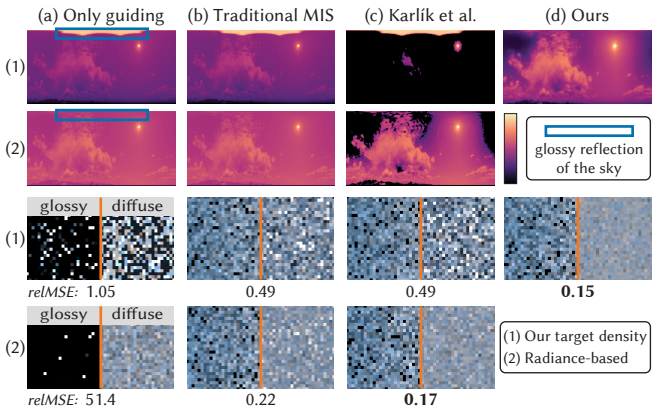


Fig. 8. A single density is learned for a half glossy, half diffuse quad, illuminated by an environment map. The top rows compare the densities, the bottom rows the rendered images. (a) uses only guiding, (b) uses one-sample MIS with BSDF importance sampling, (c) additionally applies the method of Karlík et al. [2019], and (d) uses our MIS compensation. Here, BSDF importance sampling handles the reflection of the sky in the glossy surface well. Our method successfully removes the corresponding directions from the target density, resulting in a lower error than the other approaches.

a tabulated PDF, which effectively enhances contrast. While this works well for the radiance-based target density, it does not always perform well when the BSDF is included. In the example from Fig. 8, their method cannot remove the strong glossy reflection of the sky.

We propose an alternative approximation: to pretend that the MIS weights were independent of the PDFs. Then, instead of learning to sample the full integral, we only need to learn how to sample the MIS weighted portion of the guided technique:

$$L_\mathrm{o} = \underbrace{\int_\Omega w_g \langle L_\mathrm{i}\rangle B |\cos\theta_\mathrm{i}|\,\mathrm{d}\omega_\mathrm{i}}_{\text{guided portion}} + \int_\Omega w_B \langle L_\mathrm{i}\rangle B |\cos\theta_\mathrm{i}|\,\mathrm{d}\omega_\mathrm{i}. \tag{24}$$

The corresponding target density would simply contain the MIS weight as well. For our target density marginalized over the BDSF (12), for example, that would yield:

$$p(\omega_\mathrm{i} \mid x) \propto \sqrt{\mathbb{E}_{\omega_\mathrm{o}}\left[w_\mathrm{g}^2(\omega_\mathrm{i}, x, \omega_\mathrm{o})\, B^2\, \mathbb{E}\left[\langle L_\mathrm{i}\rangle^2\right]\right]}\,|\cos\theta_\mathrm{i}|. \tag{25}$$

The balance heuristic, of course, is not constant with respect to the PDF. In an iterative learning scheme, however, the densities, and hence the balance heuristic weight, tend to change smoothly between iterations. Therefore, we multiply the current balance heuristic weight on the sample weights, starting with the first guided training iteration. In our simple example (see Fig. 8), our approach successfully eliminates the glossy reflection of the sky, which is already captured well by BSDF importance sampling.

5.2 Selection probability

Sometimes, the best guiding decision might be not to learn anything and rely solely on BSDF importance sampling. One example could be almost specular surfaces, where the incident radiance is insignificant compared to the BSDF. Using guiding on such surfaces, even in an MIS combination, can increase variance unless the selection probability α is chosen carefully. Some care has to be taken, however, since a poorly chosen α can be far worse than a uniform probability.[1]

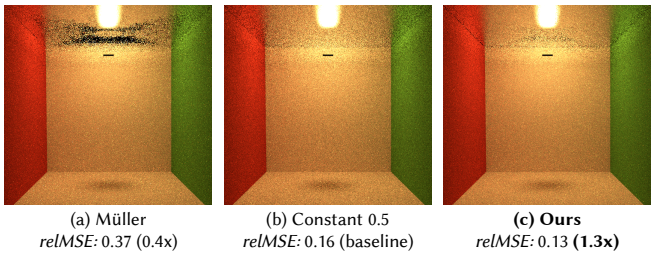


Fig. 9. A variant of the Cornell box with a glossy ceiling, rendered with different BSDF selection probabilities. The method proposed by Müller (a) performs worse than the uniform selection baseline (b). Our modification (c) increases both robustness and efficiency.

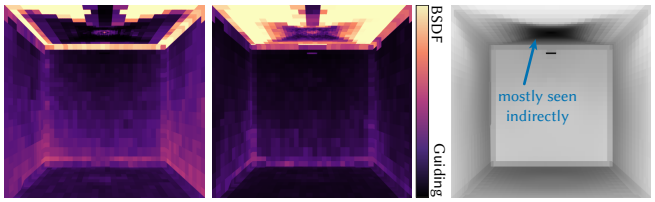


Fig. 10. Comparison of the learned selection probabilities (a-b) in the modified Cornell box. The gray-scale image (c) shows the ratio of primary to secondary rays in each guiding cache. White cells are only seen directly by the camera, black ones contribute strongly to indirect illumination. Guiding on the ceiling is only beneficial for indirect illumination, so it should only occur if the ratio (c) is close to zero, as is the case for our method.

Finding the optimal α has been investigated in previous work [Havran and Sbert 2014; Sbert et al. 2016]. A solution in the path guiding context was proposed by Müller [Vorba et al. 2019]. Their motivation is that, ideally, the effective density should correspond to the zero-variance density. Thus, they propose to minimize the divergence between the effective density $p_\mathrm{eff} = (1-\alpha)p_\mathrm{g} + \alpha p_B$ and the zero-variance density p_zv, using stochastic gradient descent.

Unfortunately, the optimal selection probability is a function of the outgoing direction: $\alpha(\omega_\mathrm{o}, x)$. In practice, however, only a single value per cache cell S is learned, i.e., $\alpha(x \in S)$. Therefore, the outcome of the gradient descent will not be the optimal α. Instead, the expected divergence over all ω_o is minimized:

$$\alpha(x \in S) = \arg\min_\alpha \mathbb{E}_{\omega_\mathrm{o}}\left[\mathcal{D}(p_\mathrm{eff} \,||\, p_\mathrm{zv})\right]. \tag{26}$$

Here, $\mathcal{D}$ denotes some divergence function, e.g., Kullback-Leibler.

Minimizing the expected divergence over a given distribution of ω_o is certainly a reasonable approach. However, it is not optimal. When comparing this to our target density (12), the potential problem becomes apparent: Only the *distribution* of ω_o is considered, not the *contributions* of the different directions.

[1] A safer alternative to optimizing the selection probability is the use of control variates [Kondapaneni et al. 2019; Owen and Zhou 2000].

An extreme case scenario is shown in Fig. 9. A small light is turned towards a glossy ceiling, indirectly illuminating a box. The scene is rendered with the original guiding approach of Müller et al. [2017]. Whether guiding is beneficial on the ceiling greatly depends on ω_o: For directly visible points, BSDF importance sampling is far superior and guiding performs poorly. For the indirect illumination on the walls, however, the light is at a grazing angle of the glossy lobe, hence guiding is the better choice there. Unfortunately, as visualized in Fig. 10, the majority of outgoing directions on the ceiling are due to the indirect reflection. Hence, the optimization for α neglects the directly visible component, producing a severe variance artefact.

To correct this issue, we apply a similar approach as with our target densities: Instead of minimizing the expected divergence, we minimize the expected divergence weighted by the contribution:

$$\alpha(x \in S) = \arg\min_{\alpha} \sum_{\text{px}} \mathbb{E}_{\bar{x}} \left[\mathcal{D}(p_{\text{eff}} \,||\, p_{\text{zv}}) \frac{f_{\text{px}}(\bar{x})}{\tilde{I}_{\text{px}}} \right], \tag{27}$$

where $\bar{x}$ is a camera path leading to the point x, $f_{\text{px}}(\bar{x})$ is the pixel contribution, and $\tilde{I}_{\text{px}}$ is the approximated ground truth value of the pixel. The resulting α does not produce the artefact in the extreme case discussed here, and performs better or equal throughout all our test scenes.

It is important to guarantee robustness when optimizing the selection probability. The guiding distribution can cause bias or unbounded variance due to estimation errors. Therefore, it is generally a good idea to, at the very least, enforce $\alpha \geq 0.1$ [Owen and Zhou 2000]. Furthermore, the stochastic gradient descent optimiziation proposed by Müller [Vorba et al. 2019] can behave unpredictably for severe outliers. Finding more robust estimation schemes is a very important but orthogonal problem, beyond the scope of this work.

6 APPLICATION I: PATH GUIDING

In this section, we discuss our primary application: designing target densities for a guided unidirectional path tracer. We apply our theory on top of the approach by Müller et al. [2017]. In the following, we first outline the mathematical formulation. Then, we present pseudo-code with the necessary changes to the implementation. Lastly, we evaluate our method on various test scenes. The source code and rendered images can be found in the supplemental materials.

6.1 Estimating the target density

A nested tree structure is used to represent the guiding distribution. The scene is partitioned into independent guiding caches S by a binary tree, each approximating a directional density $p(\omega_i \mid x \in S)$, using a quad-tree. We have derived an optimal target density for that case, (21). Note that our derivations locally optimize densities assuming that all other decisions are fixed. In the path guiding setting, decisions along the random walk of the nested estimator are also guided. Because we are training in iterations, this is not a problem: each iteration learns a density for the current variance of the nested estimator, which typically is close to the actual variance in the next iteration.

The remaining question is how to optimally approximate our PDF using a piecewise constant quad-tree. Each leaf node k in the quad-tree stores a weight γ_k, which determines the probability

```
1   function Render():
2       for i in iterations:
3           RenderImage()
4
5 +         for Leaf in NextCache:
6 +             Leaf.Value := Sqrt(
7 +                 Leaf.Value * Leaf.Area
8 +             )
9
10          NextCache.Normalize()
11          CurrentCache := NextCache
12          NextCache.Reset()
13
14  function Lo(x, ωo,
15 +             RelThroughput):
16      // One sample MIS
17      if Random() > α:
18          ωi := CurrentCache(x).Sample()
19      else:
20          ωi := BSDF(x, ωo).Sample()
21      // MIS computations
22      MisPDF := (1-α)     * CurrentCache(x).PDF(ωi)
23                + α       * BSDF(x, ωo).PDF(ωi)
24 +    MisWeight := CurrentCache(x).PDF(ωi) / MisPDF
25      // Compute recursive estimate
26      BsdfCos := BSDF(x, ωo).Eval(ωi) * Cos(θi)
27      Li := Lo(RayTrace(x, ωi), -ωi,
28 +             RelThroughput * BsdfCos / MisPDF)
29      // Update guiding cache
30      NextCache.Leaf(x, ωi) += (1 / MisPDF) *
31 -        Li
32 +        (BsdfCos * Li * MisWeight * RelThroughput)^2
33      // Update BSDF sampling fraction loss
34      MISLoss(x).Update(
35          BsdfCos * Li
36 +        * RelThroughput
37      )
38      // Compute rendering equation estimate
39      return Le(x, ωo) + BsdfCos * Li / MisPDF
```

Fig. 11. Pseudo-code with the required changes to compute our full density with the algorithm by Müller et al. [2017]. Lines starting with "-", highlighted in red, denote parts that are replaced by our approach. Those starting with "+", in green, compute our proposed target density.

of choosing the corresponding set of directions D_k for piecewise uniform sampling. The optimal value for γ_k can be computed using the approach of Pantaleoni and Heitz [2017]. For our full target density (21) the result is:

$$\begin{aligned} \gamma_k &\propto \sqrt{|D_k| \int_{\omega_i \in D_k} p^2_{\text{full}}(\omega_i \mid x \in S)\, \mathrm{d}\omega_i} \\ &= \sqrt{|D_k| \int_{\omega_i \in D_k} \mathbb{E}\left[\frac{f^2_{\text{px}}(\bar{x})}{\tilde{I}^2_{\text{px}} p^2(\bar{x})} B^2 \langle L_i \rangle^2 \cos^2\theta_i \right] \mathrm{d}\omega_i}, \end{aligned} \tag{28}$$

where $|D_k|$ is the size of the kth leaf, measured in solid angle. We can easily estimate the leaf node weights γ_k by accumulating the squared sample weights, multiplying once by the leaf size, and taking a square root prior to normalization.

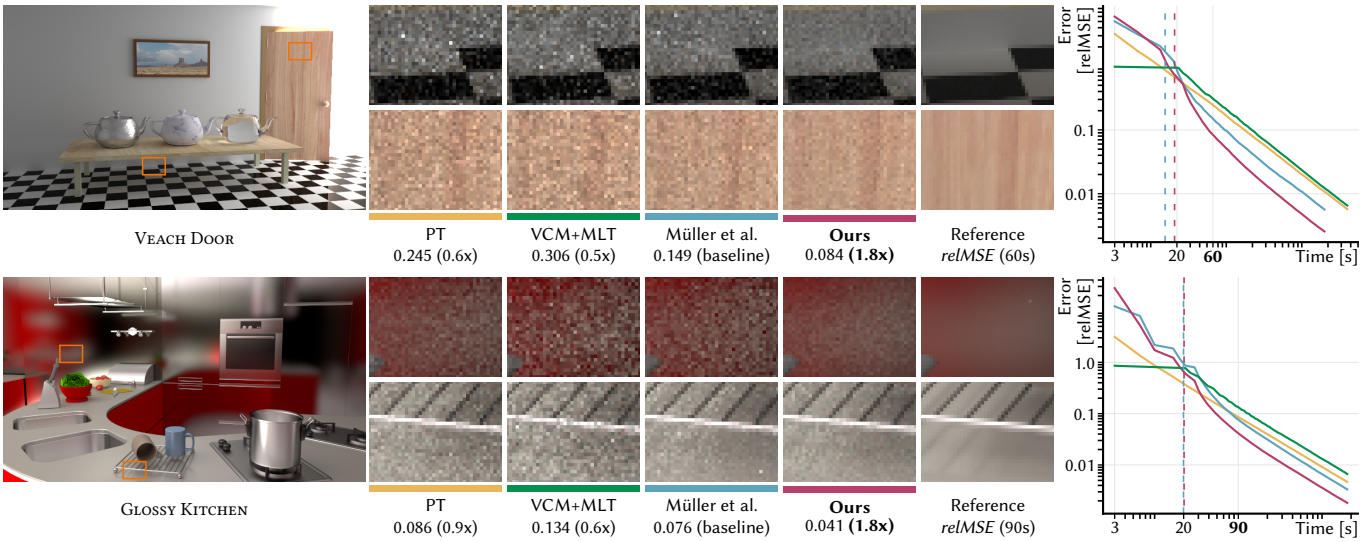


Fig. 12. Equal-time comparisons for two test scenes. The dashed lines in the plots mark the end of the last training iteration of the guiding methods. The rendering time of the comparison images is highlighted (60s for Veach Door and 90s for Glossy Kitchen).

6.2 Implementation

We have applied our theory in the Mitsuba [Jakob 2010] implementation provided by Müller et al. [2017]. The pseudo-code in Fig. 11 highlights the required changes. We have implemented our full target density (21) with the MIS compensation and selection optimiziations discussed in Section 5.

Müller et al. render the image in iterations (see the function `Render` in lines 1–3). Thus, the guiding cache learned in the previous iteration, `CurrentCache`, can be used for importance sampling while learning a new cache for the next iteration, `NextCache`. After each iteration is finished, we multiply the leaf values by the leaf sizes, take the square roots, and normalize (lines 5–12).

The $\mathtt{L_o}$ function (line 14) is called by `RenderImage` to recursively estimate the rendering equation. Note that our full distribution requires us to keep track of the relative pixel contribution $f_{\text{px}}(\bar{x})/(p(\bar{x})\tilde{I}_{\text{px}})$, see (21), which we do here using the parameter `RelThroughput`. When called directly from `RenderImage`, this parameter is set to the sensor response divided by the pixel estimate, i.e., $W_{\text{px}}/\tilde{I}_{\text{px}}$. We update `RelThroughput` for the recursive evaluation of $\mathtt{L_o}$ in lines 27–28.

Irrespective of which technique was chosen for one-sample MIS in lines 16–23, we always need to compute the `MisWeight` for the guiding strategy (line 24) for our MIS compensation (25). Furthermore, in line 32, we accumulate the squared sample weights, multiplied by the MIS weight, to estimate our target densities. Lastly, we apply our modified selection probability optimizer in line 36, again weighting by the relative contribution.

6.3 Results

We evaluated our method on a corpus of 22 scenes, all of which are rendered at a resolution of around 640×360 on an AMD Ryzen™ 9 3950X (16 cores / 32 threads @4.0 GHz) workstation with 64 GB of memory. No Russian roulette is performed to aid comparability. We compare our approach to radiance-based guiding, the target density used by previous work. Both guiding approaches make use of next event estimation. In addition, we compare to an unguided path tracer with next event estimation, and a VCM method that uses Markov chains to distribute photons [Šik et al. 2016]. For the latter, we used the authors' publicly available Mitsuba implementation.

In the following, we discuss the differences in equal time renderings on four representative examples. The full results, including convergence tests with long training times, can be found in the supplemental material.

The Veach Door scene (Fig. 12, first row) shows how our method reduces spatial caching artefacts. Both the wall and the door are very challenging, as the surfaces on both sides end up in the same spatial cache. Even though the backside is more strongly illuminated than the side seen from the camera, its contribution to the image is less important. Our density mitigates this problem by assigning lower weight to the samples from the backside illumination, resulting in lower levels of noise overall.

The Glossy Kitchen scene (Fig. 12, second row) features many glossy surfaces. By incorporating the BSDF into our density (see Section 4.2), we are able to improve performance in regions where both radiance-based guiding and BSDF sampling perform poorly.

The Pool scene (Fig. 13, first row) features caustics which are challenging to render. The caustics in the pool feature a similar light transport to Fig. 4: sunlight is seen directly through the water surface as well as reflected by the window on the right, causing outliers on the pool floor in radiance-based approaches. Our density eliminates these outliers by taking the variance due to the unguided decision on the glass into account.

The Bookshelf scene (Fig. 13, second row) features strong indirect illumination and is thus among the most challenging scenes for our guiding density. Since our density contains the full pixel

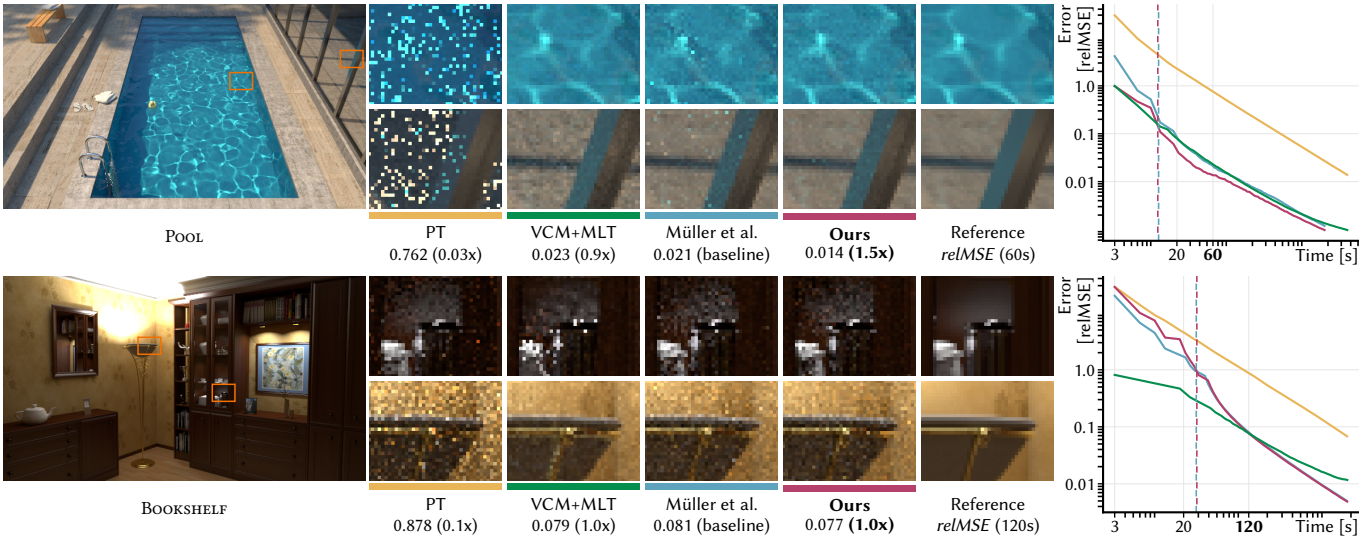


Fig. 13. Equal-time comparisons for two test scenes. The dashed lines in the plots mark the end of the last training iteration of the guiding methods. The rendering time of the comparison images is highlighted (60s for Pool and 120s for Bookshelf).

integral, opposed to just the radiance, its estimate can be noisier for longer paths. Nevertheless, our method still achieves the same performance as the original method, at least for longer renderings. This scene also shows quite well how unidirectional path guiding is still sometimes outperformed by bidirectional methods like VCM, especially for short renderings.

Our target density only consistently outperforms the baseline with sufficient training. To measure the training cost, we computed the error (relMSE) in a 512spp rendering after different training times. The ratio of that error between our method and the approach taken by previous work is shown in Fig. 14, averaged across all scenes. After 1.5 seconds of training, our target density outperformed the previous one on average, after 10 seconds we outperform it in every single scene. Hence, interactive renderings will not benefit from our results, as the training data is too scarce to accurately learn the density. Longer running renderings of more than ten minutes, however, receive consistent improvements and converge 50% faster on average.

In conclusion, our target density offers visible improvements across all scenes, at essentially no cost. While it is not suited for interactive preview renders, it is a robust alternative to radiance-based guiding for long running renders, where it can noticeably accelerate convergence.

7 APPLICATION II: LIGHT SELECTION

We tested our theory in a different context and code base, by applying it to a light source selection method [Vévoda et al. 2018]. Vévoda et al. apply a Bayesian approach, where they start with a coarse, analytic approximation as a prior distribution. During rendering, they gradually learn a better posterior distribution. Their distribution already compensates for the variance of nested estimators. That is, they effectively implemented the discrete analogy of our

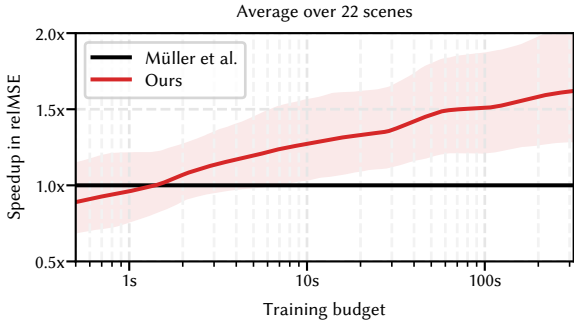


Fig. 14. Comparison of training cost. We plot the ratio of the relMSE after equal time (the 'speedup'), averaged over 22 scenes, using the geometric mean. The shaded region visualizes how much that ratio varies across scenes. The error is that of a 512spp rendering after different training times.

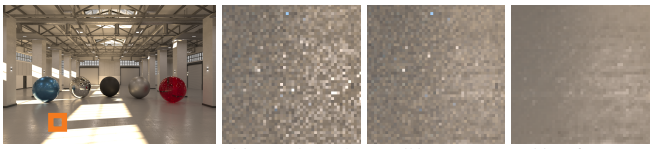
target distribution for irradiance (9). We modified their method to additionally marginalize over the BSDF, i.e., compute our simplified target distribution (20).

7.1 Implementation

We implemented the original approach and our changes in a custom renderer. Computing our target distribution is also trivial in this case. In principle, only one change is required: we remove the upper bound of the cosine term that was originally used, and instead multiply the BSDF and cosine on the weight of each sample. To that end, we modify their equation (6) to now read:

$$\hat{e} = B(\omega_o, x \to y)\frac{L_e(y \to x)V(y \leftrightarrow x)G(y \leftrightarrow x)}{P(l \mid c)\,p(y \mid l)} = \hat{e}_x. \qquad (29)$$

Here, y is a point on light source l in cluster c and x is the shading point. V is the visibility term and G the full geometry term, now



	(a) Vévoda et al.	**(b) Ours**	(c) Reference
relMSE whole image:	0.0070 (baseline)	0.0066 **(1.1x)**	
relMSE zoom-in:	1.2 (baseline)	0.33 **(3.5x)**	

Fig. 15. Results from our light selection application. The Hall scene features many small lights and a glossy surface. Here, even BSDF importance sampling performs poorly. Our method achieves visible noise reduction by marginalizing over the BSDF. The images were rendered with equal sample count, which is also equal time, since our modifications caused no overhead.

including both cosines. $P(l \mid c)$ and $p(y \mid l)$ are the probabilities of selecting light source l in cluster c and point y on that light, respectively. The sample weights $\hat{e}$ and $\hat{e}_x$, which are now equal for our target distribution, are used to learn the posterior distribution.

We also performed another change that is not strictly required but improved the learning rate. The prior distribution for the original method was built with the original target distribution in mind, which does not contain the BSDF or cosine terms. Multiplying by the BSDF and cosine yields smaller weights, which we crudely approximated by dividing equation (10) in the original paper by a factor of eight, a number we found to perform well empirically over all our tests. Finding the optimal prior distribution is an orthogonal and application specific improvement left for future work.

7.2 Results

We tested our modified algorithm across a variety of test scenes. In this application, the difference was less significant than with the local path guiding method, since the nested estimator's variance was already accounted for previously. For most scenes, our result was only slightly better than the original version. We also did not find a single scene where our target distribution performed worse.

One specific type of scene can benefit greatly from our target distributions: Scenes with glossy surfaces and small light sources. Glossy surfaces were neglected by the previous distribution and left for the BSDF importance sampling strategy to resolve. However, if the light sources are small, BSDF importance sampling can often perform poorly, even on glossy surfaces. An example is shown in Fig. 15. In this modified version of the Hall scene, we reduced the size of the light sources. Neither the original target distribution nor the BSDF importance sampling strategy can resolve all of the glossy highlights. By marginalizing over the BSDF, our target distribution achieves visible improvements in the highlights, while producing identical results everywhere else.

In conclusion, while the benefit of our target distribution is less significant in this application, we still achieved robust improvements with trivial changes and no additional overhead.

8 LIMITATIONS AND FUTURE WORK

Isolated optimization. We have optimized local decisions in isolation. The individual optimizations (target density, selection probability, and MIS compensation) are aware of each other only through the observed changes in the sample weights of future iterations. In our empirical tests, adding another isolated optimization resulted in consistent improvements. There is, however, no strict proof that the isolated decisions will converge to an optimal joint distribution. Further investigation in that direction is an interesting avenue for future work.

Short renderings. Target densities that are theoretically optimal can still result in poor rendering performance when estimated with few training samples, i.e., for short preview renderings. In Section 6, we have observed this effect, where our full density sometimes only outperforms the baseline after sufficient training. Estimating a higher dimensional integral, namely the image contribution of the guiding cache, results in higher levels of noise. There are multiple possibilities to improve performance for these cases. One option is to apply reconstruction or denoising methods to the guiding caches. Another option is to design prior distributions and utilize a Bayesian approach [Vévoda et al. 2018].

Other target densities. We have focused on target densities that would be optimal if they were estimated exactly. A different approach, that could also improve performance for short render times, would be to design target functions that are easier to learn. To that end, regularization could be employed [Kaplanyan and Dachsbacher 2013] or a binary distribution could be learned, similar to MCMC target functions that only include visibility [Hachisuka and Jensen 2011]. Exploring such target densities is an interesting avenue for future work. It could also be interesting to find target densities that minimize different error metrics, like perceptually-based ones.

Bidirectional guiding. The applications presented in this paper can easily estimate the variance due to nested estimators, as the training samples are generated from a distribution similar to the one that will be used during rendering. If that is not the case, e.g., because guiding is done bidirectionally [Jensen 1995; Vorba et al. 2014], computing the target densities is more involved, but still possible. Computing PDFs in a bidirectional setting can be tedious, a main reason why these methods are less appealing in practice. A workaround to ease implementation effort could be to only use the bidirectional samples to initialize a simpler, coarse guiding distribution. Successive iterations can then learn unidirectionally (possibly from both sides) and easily estimate our target densities.

9 CONCLUSION

Existing guiding approaches pursue the dream of zero-variance sampling: If only we could make every local sampling decision perfect, the whole estimator would have zero variance. In reality, numerous limitations currently prevent this dream from becoming a reality. Some decisions cannot be made perfect, for example, because guiding them requires too long training times. We present a general approach on how to deal with these constraints and design target densities for path guiding that are optimal if zero-variance sampling is not feasible. We apply our theory to state of the art path guiding methods. The trivial modifications necessary to compute our target densities yield significant gains in efficiency and robustness.

ACKNOWLEDGMENTS

This work was partially supported by RWTÜV-Stiftung as part of the HUSSAR project. We thank Mira Niemann for help with figure design. We also thank the following people for providing test scenes: Benedikt Bitterli [2016], Johannes Hanika (Necklace), Miika Aittala, Samuli Laine, and Jaakko Lehtinen (Veach Door), Evermotion and Tiziano Portenier (Glossy Kitchen, Bookshelf, Glossy Bathroom), Ondřej Karlík (Pool), Wig42 (Living Room, Staircase, Dining Room), Jay-Artist (Kitchen, Living Room (2)), Samuli Laine and Olesya Jakob (Hairball), Marios Papas, Maurizio Nitti and Marko Dabrović (Sponza), Nacimus (Bathroom), SlykDrako (Bedroom), thecali (Spaceship), and MrChimp2313 (House).

REFERENCES

Steve Bako, Mark Meyer, Tony DeRose, and Pradeep Sen. 2019. Offline Deep Importance Sampling for Monte Carlo Path Tracing. *Comput. Graph. Forum (Proceedings of Pacific Graphics 2019)* 38, 7 (2019), 527–542.

Thomas Bashford-Rogers, Kurt Debattista, and Alan Chalmers. 2012. A significance cache for accelerating global illumination. In *Comput. Graph. Forum*, Vol. 31. Wiley Online Library, 1837–1851.

Benedikt Bitterli. 2016. Rendering resources. https://benedikt-bitterli.me/resources/

Brent Burley, David Adler, Matt Jen-Yuan Chiang, Hank Driskill, Ralf Habel, Patrick Kelly, Peter Kutz, Yining Karl Li, and Daniel Teece. 2018. The design and evolution of disney's hyperion renderer. *ACM Trans. Graph. (TOG)* 37, 3 (2018), 33.

Norbert Bus and Tamy Boubekeur. 2017. Double Hierarchies for Directional Importance Sampling in Monte Carlo Rendering. *Journal of Computer Graphics Techniques (JCGT)* 6, 3 (28 August 2017), 25–37. http://jcgt.org/published/0006/03/02

Luca Fascione, Johannes Hanika, Mark Leone, Marc Droske, Jorge Schwarzhaupt, Tomáš Davidovič, Andrea Weidlich, and Johannes Meng. 2018. Manuka: A batch-shading architecture for spectral path tracing in movie production. *ACM Trans. Graph. (TOG)* 37, 3 (2018), 31.

Iliyan Georgiev, Thiago Ize, Mike Farnsworth, Ramón Montoya-Vozmediano, Alan King, Brecht Van Lommel, Angel Jimenez, Oscar Anson, Shinji Ogaki, Eric Johnston, et al. 2018. Arnold: A brute-force production path tracer. *ACM Trans. Graph. (TOG)* 37, 3 (2018), 32.

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

Iliyan Georgiev, Jaroslav Křivánek, Stefan Popov, and Philipp Slusallek. 2012b. Importance caching for complex illumination. In *Comput. Graph. Forum*, Vol. 31. Wiley Online Library, 701–710.

Pascal Grittmann, Arsène Pérard-Gayot, Philipp Slusallek, and Jaroslav Křivánek. 2018. Efficient Caustic Rendering with Lightweight Photon Mapping. In *Comput. Graph. Forum (EGSR '18)*, Vol. 37. Wiley Online Library, 133–142.

Adrien Gruson, Mickaël Ribardière, Martin Šik, Jiří Vorba, Rémi Cozot, Kadi Bouatouch, and Jaroslav Křivánek. 2017. A spatial target function for metropolis photon tracing. *ACM Trans. Graph. (TOG)* 36, 1 (2017), 4.

Jerry Guo, Pablo Bauszat, Jacco Bikker, and Elmar Eisemann. 2018. Primary sample space path guiding. In *Eurographics Symposium on Rendering*, Vol. 2018. The Eurographics Association, 73–82.

Toshiya Hachisuka and Henrik Wann Jensen. 2011. Robust adaptive photon tracing using photon path visibility. *ACM Trans. Graph. (TOG)* 30, 5 (2011), 114.

Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. 2008. Progressive photon mapping. In *ACM Trans. Graph. (TOG)*, Vol. 27. ACM, 130.

Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. 2012. A path space extension for robust light transport simulation. *ACM Trans. Graph. (TOG)* 31, 6 (2012), 191.

Vlastimil Havran and Mateu Sbert. 2014. Optimal Combination of Techniques in Multiple Importance Sampling. ACM, New York, NY, 141–150.

Sebastian Herholz, Oskar Elek, Jens Schindel, Jaroslav Křivánek, and Hendrik Lensch. 2018. A Unified Manifold Framework for Efficient BRDF Sampling based on Parametric Mixture Model. In *EGSR '18 EI&I (EGSR '18)*. Eurographics Association.

Sebastian Herholz, Oskar Elek, Jiří Vorba, Hendrik P. A. Lensch, and Jaroslav Křivánek. 2016. Product Importance Sampling for Light Transport Path Guiding. *Comput. Graph. Forum* 35 (2016), 67–77.

Sebastian Herholz, Yangyang Zhao, Oskar Elek, Derek Nowrouzezahrai, Hendrik P. A. Lensch, and Jaroslav Křivánek. 2019. Volume Path Guiding Based on Zero-Variance Random Walk Theory. *ACM Trans. Graph.* 38, 3, Article 25 (June 2019), 19 pages.

Heinrich Hey and Werner Purgathofer. 2002. Importance Sampling with Hemispherical Particle Footprints *(SCCG '02)*. ACM, 107–114.

Wenzel Jakob. 2010. Mitsuba renderer. http://www.mitsuba-renderer.org.

Henrik Wann Jensen. 1995. Importance Driven Path Tracing using the Photon Map. In *Rendering Techniques*.

Henrik Wann Jensen. 1996. Global illumination using photon maps. In *Rendering Techniques' 96*. Springer, 21–30.

James T. Kajiya. 1986. The Rendering Equation. *SIGGRAPH Comput. Graph.* 20, 4 (Aug. 1986), 143–150.

Anton S Kaplanyan and Carsten Dachsbacher. 2013. Path space regularization for holistic and robust light transport. In *Comput. Graph. Forum*, Vol. 32. Wiley Online Library, 63–72.

Ondřej Karlík, Martin Šik, Petr Vévoda, Tomáš Skřivan, and Jaroslav Křivánek. 2019. MIS Compensation: Optimizing Sampling Techniques in Multiple Importance Sampling. *ACM Trans. Graph. (SIGGRAPH Asia '19)* 38, 6 (2019), 12.

Alexander Keller, Luca Fascione, Marcos Fajardo, Iliyan Georgiev, Per H Christensen, Johannes Hanika, Christian Eisenacher, and Gregory Nichols. 2015. The path tracing revolution in the movie industry.. In *SIGGRAPH Courses*. 24–1.

Ivo Kondapaneni, Petr Vévoda, Pascal Grittmann, Tomáš Skřivan, Philipp Slusallek, and Jaroslav Křivánek. 2019. Optimal Multiple Importance Sampling. *ACM Trans. Graph. (SIGGRAPH 2019)* 38, 4 (July 2019), 37:1–37:14.

Eric P. Lafortune and Yves D. Willems. 1993. Bi-Directional Path Tracing. 93 (Dec. 1993), 145–153.

Eric P. Lafortune and Yves D. Willems. 1995. A 5D Tree to Reduce the Variance of Monte Carlo Ray Tracing. In *Rendering Techniques*.

Thomas Müller, Markus H. Gross, and Jan Novák. 2017. Practical Path Guiding for Efficient Light-Transport Simulation. *Comput. Graph. Forum* 36 (2017), 91–100.

Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. 2018. Neural importance sampling. *arXiv preprint arXiv:1808.03856* (2018).

Art Owen and Yi Zhou. 2000. Safe and Effective Importance Sampling. *J. Amer. Statist. Assoc.* 95, 449 (2000), 135–143.

Jacopo Pantaleoni. 2019. Importance Sampling of Many Lights with Reinforcement Lightcuts Learning. *arXiv preprint arXiv:1911.10217* (2019).

Jacopo Pantaleoni and Eric Heitz. 2017. Notes on optimal approximations for importance sampling. *arXiv preprint arXiv:1707.08358* (2017).

Vincent Pegoraro, Carson Brownlee, Peter S Shirley, and Steven G Parker. 2008. Towards interactive global illumination effects via sequential Monte Carlo adaptation. In *2008 IEEE Symposium on Interactive Ray Tracing*. IEEE, 107–114.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically based rendering: From theory to implementation*. Morgan Kaufmann.

Florian Reibold, Johannes Hanika, Alisa Jung, and Carsten Dachsbacher. 2018. Selective guided sampling with complete light transport paths. In *SIGGRAPH Asia 2018 Technical Papers*. ACM, 223.

Mateu Sbert, Vlastimil Havran, and Laszlo. Szirmay-Kalos. 2016. Variance Analysis of Multi-sample and One-sample Multiple Importance Sampling. *Comput. Graph. Forum* 35, 7 (2016), 451–460.

Martin Šik and Jaroslav Křivánek. 2018. Survey of Markov Chain Monte Carlo Methods in Light Transport Simulation. *IEEE Transactions on Visualization and Computer Graphics* (2018), 1–1.

Martin Šik and Jaroslav Křivánek. 2019. Implementing One-Click Caustics in Corona Renderer. (2019).

Martin Šik, Hisanari Otsu, Toshiya Hachisuka, and Jaroslav Křivánek. 2016. Robust light transport simulation via metropolised bidirectional estimators. *ACM Trans. Graph. (TOG)* 35, 6 (2016), 245.

Eric Veach. 1997. *Robust Monte Carlo methods for light transport simulation*. Stanford University PhD thesis.

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

Eric Veach and Leonidas J Guibas. 1995b. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *SIGGRAPH '95*. ACM, 419–428.

Eric Veach and Leonidas J Guibas. 1997. Metropolis light transport. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 65–76.

Petr Vévoda, Ivo Kondapaneni, and Jaroslav Křivánek. 2018. Bayesian online regression for adaptive direct illumination sampling. *ACM Trans. Graph. (TOG)* 37, 4 (2018), 125.

Jiří Vorba, Johannes Hanika, Sebastian Herholz, Thomas Müller, Jaroslav Křivánek, and Alexander Keller. 2019. Path Guiding in Production *(SIGGRAPH '19)*. ACM, New York, NY, USA, Article 18, 18:1–18:77 pages. https://doi.org/10.1145/3305366.3328091

Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. 2014. On-line Learning of Parametric Mixture Models for Light Transport Simulation. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2014)* 33, 4 (2014).

Jiří Vorba and Jaroslav Křivánek. 2016. Adjoint-driven Russian Roulette and Splitting in Light Transport Simulation. *ACM Trans. Graph.* 35, 4, Article 42 (July 2016), 11 pages.

Quan Zheng and Matthias Zwicker. 2019. Learning to importance sample in primary sample space. In *Comput. Graph. Forum*, Vol. 38. Wiley Online Library, 169–179.