

Less can be more: A footprint-driven heuristic to skip wasted connections and merges in bidirectional rendering

Ömercan Yazici¹ , Pascal Gittmann¹  and Philipp Slusallek^{1,2} 

¹Saarland University, Saarland Informatics Campus, Saarbrücken, Germany

²DFKI, Saarland Informatics Campus, Saarbrücken, Germany

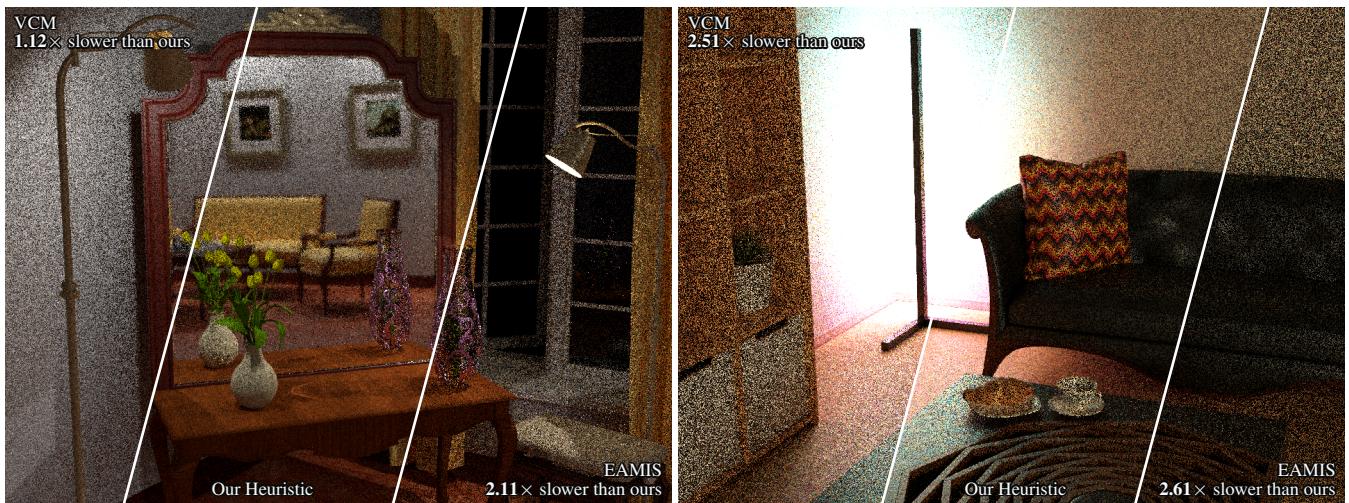


Figure 1: Equal-time (5s) renderings of two scenes that benefit from bidirectional rendering – though not for every lighting effect. VCM [GKDS12] (aka UPS [HPJ12]) wastes time doing merges and connections everywhere. EAMIS [GYGS22] uses statistics to disable those operations in some pixels, but this is coarse, suffers from noise, and the initial overhead hampers short renderings like the ones shown here. We propose a simple, footprint-based heuristic that can effectively skip wasteful computations, without introducing overhead.

Abstract

Bidirectional rendering algorithms can robustly render a wide range of scenes and light transport effects. Their robustness stems from the fact that they combine a huge number of sampling techniques: Paths traced from the camera are combined with paths traced from the lights by connecting or merging their vertices in all possible combinations. The flip side of this robustness is that efficiency suffers because most of these connections and merges are not useful – their samples will have a weight close to zero. Skipping these wasted computations is hence desirable. Prior work has attempted this via manual parameter tuning, by classifying materials as “specular”, “glossy”, or “diffuse”, or via costly data-driven adaptation. We, instead, propose a simple footprint-driven heuristic to selectively enable only the most impactful bidirectional techniques. Our heuristic is based only on readily available PDF values, does not require manual tuning, supports arbitrarily complex material systems, and does not require precomputation.

CCS Concepts

- Computing methodologies → Ray tracing; Rendering;

1. Introduction

Bidirectional rendering algorithms [VG95a; GKDS12; HPJ12] are exceptionally good at rendering complex indirect illumination or

caustics. The secret to their success lies in the vast set of sampling techniques they employ: Paths traced from the light and paths traced from the camera are combined in every possible way, using

shadow-ray connections or nearest-neighbor merges (aka photon mapping).

Thanks to this vast set of sampling techniques, bidirectional algorithms can render a wide range of scenes robustly, that is, with reasonable performance. However, only a small subset of the samples generated will actually contribute significantly to the rendered image. Consequently, a significant amount of compute resources is wasted generating irrelevant samples.

In this paper, we propose a heuristic to skip connection and merging when they are likely to be wasteful. Our heuristic is based on the observation that merging and connecting after a low-density scattering event (e.g., a diffuse or rough BSDF) is rarely useful. We want to exploit this knowledge during rendering, but without imposing constraints on the material system like prior methods that classify materials as “glossy” [Jen96]. We also want to avoid data-driven solutions that require additional memory, a training phase, and struggle with noisy samples [MRYS24; GYGS22]. Thus, we compute the sampling footprint at each vertex along a camera path. Once this footprint drops below a threshold – defined via a scene-agnostic hyperparameter – all connections and merges are disabled. The quality of our heuristic can be further improved via data-driven adaptation, if desired, but doing so is not required.

We apply our heuristic to the vertex connection and merging (VCM) algorithm [GKDS12; HPJ12]. Normally, this algorithm would perform merging and connections at every camera path vertex. By skipping most of these, we can achieve huge equal-time performance improvements, as exemplified in Figure 1.

2. Background and related work

Monte Carlo rendering computes images by sampling n random paths x to form a Monte Carlo estimate

$$\langle F \rangle = \sum_{i=1}^n \frac{f(x_i)}{np(x_i)} \approx \int_{\mathcal{X}} f(x) dx = F, \quad (1)$$

where F is the integral of a function $f(x)$ we want to compute – the rendering equation [Vea97; Kaj86] in our case – and $p(x)$ is the probability density function (PDF) of the samples x_i .

The error of such an estimator is given by the variance

$$\mathbb{V}[\langle F \rangle] = \mathbb{E}[\langle F \rangle^2] - F^2. \quad (2)$$

As different estimators usually have different cost C , we use the inefficiency $C \cdot \mathbb{V}[\langle F \rangle]$, i.e., the product of variance and cost, to quantify estimator performance.

2.1. Multiple importance sampling

The efficiency of a Monte Carlo estimator hinges on the PDF $p(x)$. If the PDF closely matches the integrand $f(x)$, the variance will be low. Sadly, it is virtually impossible to find a single PDF that satisfies this goal for the light transport equation: its high-dimensional nature and many discontinuities prevent that. A common solution, therefore, is to combine multiple techniques $t \in T$ with PDF $p_t(x)$ in the hope that at least one will work well. This is known as multiple importance sampling (MIS) [VG95b] and has the following

estimator:

$$\langle F_{\text{MIS}} \rangle = \sum_{t \in T} \sum_{i=1}^{n_t} w_t(x_{t,i}) \frac{f(x_{t,i})}{n_t p_t(x_{t,i})}. \quad (3)$$

For each technique t we generate n_t samples $x_{t,i} \sim p_t$. To combine the techniques we weight the samples with a weighting function $w_t(x)$. A common choice is the balance heuristic,

$$w_t(x) = \frac{n_t p_t(x)}{\sum_{k \in T} n_k p_k(x)}. \quad (4)$$

The balance heuristic is popular because it is easy to compute and provides worst-case bounds on the variance. However, it is not optimal [KVG*19; HGS23] and especially prone to fail for bidirectional algorithms due to sample correlation [GGSK19; GGS21]. To address these shortcomings, prior work has suggested to extend the weights with correction factors $c_t(x)$:

$$w_t(x) = \frac{c_t(x) n_t p_t(x)}{\sum_{k \in T} c_t(x) n_k p_k(x)} \quad (5)$$

to take the variance [GGSK19], correlation [GGS21; JG18] or other factors into account.

Aside from the choice of weighting function, good use of MIS requires a well-chosen set of techniques. Ideally, techniques should be chosen such that each one samples a different part of the domain well [KŠV*19]. In the context of bidirectional rendering, this idea has, e.g., been applied to limit photon mapping to caustic effects [GPSK18]. Such adaptations of sampling densities are orthogonal to our disabling of unnecessary techniques.

More closely related to our methodology are methods that adapt the sample invested in each technique. Some researchers suggested domain-specific heuristics to control sample counts [PBPP10], but most work was done on using statistics to adapt sample counts on the fly, e.g., via gradient descent [LPG13; SHS19; Mül19], direct search [GYGS22], or fixed-point schemes [MRYS24]. Such adaptation can dramatically improve rendering performance. However, it requires a training phase – during which sampling will be sub-optimal – requires careful filtering to combat noise, and introduces overhead. Our heuristic avoids overhead and noise-related issues by relying solely on PDF values.

The goal of MIS is to assign low weight to poor-quality samples. Our goal is to avoid generating such poor samples.

2.2. Bidirectional methods

Bidirectional rendering algorithms trace prefix paths \bar{y} from the camera and combine them with suffix paths \bar{z} traced from the light sources, as shown in Figure 2.

Each of the various ways to achieve such a combination forms one sampling technique; MIS is used to combine these techniques. Many such bidirectional algorithms exist, each using a different set of combination techniques. Bidirectional path tracing [LW93; VG95a] traces a pair of one camera and one light path and connects all their vertices via shadow rays. Vertex caching [DKHS14b; PRDD15; SLW22; NID20] methods adapt this idea to instead cache a set of light path vertices. Then, each camera vertex is connected

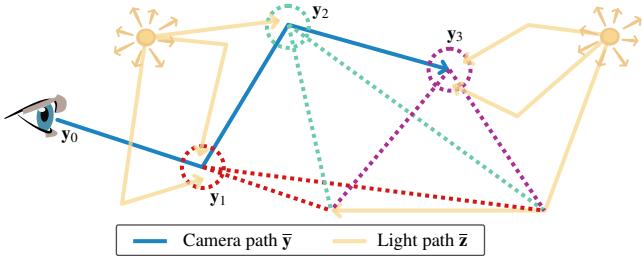


Figure 2: Bidirectional algorithms trace paths \bar{y} from the camera and paths \bar{z} from the lights. Each vertex y_i of the camera path is connected, e.g., to all vertices of one light path, and merged with all nearby vertices of all light paths.

to some number c of stochastically chosen light vertices. Virtual point light methods [Kel97] apply the same caching idea but only connect the first camera vertex that is not on a specular or glossy surface. However, for non-trivial material systems, it is not an easy task to determine if a material is sufficiently glossy or not.

Photon mapping [Jen96; HOJ08; HJJ10] also generates a cache of light path vertices. But instead of connecting these with camera paths, they are *merged*. That is, at a camera path vertex, a nearest neighbor search is performed (usually with either a maximum radius, a maximum number of neighbors, or both) and the nearby light path vertices are merged with the camera path vertex. Merging effectively pretends that the two vertices are one and the same and thus also forms a full path between the camera and a light source. Early photon mapping methods also employ material classification to determine at which camera vertex to perform merging [Jen96]. Later work on photon mapping has improved its capability to handle glossy surfaces by performing merging at multiple camera path vertices, not just the first diffuse one: The VCM/UPS algorithm [GKDS12; HJJ11] performs merging at every vertex along the camera path, using MIS (with suitably extended weights) to combine them all – just like connections in bidirectional path tracing. Our work builds on VCM/UPS.

It is well known that bidirectional methods can waste significant computation time. Prior work has tackled this, e.g., by caching visibility information to skip shadow ray evaluations [GEE20], or by using sampling statistics from a unidirectional path tracer to enable bidirectional sampling only when needed [GYGS22].

Most closely related to our idea is the use of material classification (into “diffuse”, “glossy”, and “specular”) and light path expressions [Hec90]. Such classification can be used, e.g., to only perform photon mapping on the first sufficiently diffuse surface and only with photons that underwent specular / glossy scattering [Jen96]. However, complex shading systems do not afford an obvious way to classify a material as sufficiently glossy. The lightweight photon mapping algorithm [GPSK18] uses the MIS weights of samples in lieu of material classification. But this only governs where photons are stored. We tackle an orthogonal problem and use sampling footprints to decide where to query for nearby photons or connect to light vertices.

3. Our method

Our intention is to disable wasteful connection and merging operations, while performing useful ones. To that end, we use sampling footprints as a crude variance approximation. With those, we then skip merges and connections that likely have a higher variance than other techniques for the same path.

We write the variance of a pixel as the sum of the variance $V_{\bar{y}}$ due to *camera* prefix \bar{y} and variance $V_{\bar{z}}$ due to the *light* suffix \bar{z} [BM97; GGS21]:

$$\mathbb{V}[F(\bar{x})] = \underbrace{\mathbb{V}\left[\frac{F_{\bar{z}}(\bar{y})}{p(\bar{y})}\right]}_{V_{\bar{y}}} + \underbrace{\mathbb{E}[\mathbb{V}[\langle F \rangle | \bar{y}]]}_{V_{\bar{z}}}, \quad (6)$$

where $F_{\bar{z}}(\bar{y}) = \int f(\bar{y}\bar{z}) d\bar{z}$ is our integral marginalized over the suffix paths.

Based on our intuition we assume merges and connections are likely not useful whenever the prefix \bar{y} already causes significant variance. To approximate the prefix variance, we inspect its density $p(\bar{y})$. We assume that a low-density prefix causes high variance while a high-density prefix causes low variance. That is, we effectively assume that $F_{\bar{z}}(\bar{y})$ is a high-frequency signal.

3.1. Footprint as a variance approximation

Our heuristic is based on the footprint approximation used by the correlation-aware MIS weights [GGS21]. At a camera path vertex y_i , we approximate the probability that another vertex y'_i , also sampled from the previous vertex y_{i-1} , falls within a disc of radius r around y_i as

$$P(y'_i \in D_r(y_i)) \approx \pi r^2 p(y_i | y_{i-1}). \quad (7)$$

The radius of this disc is set based on the camera footprint at the first hit. Assuming a spherical camera with a 360 degrees FOV, this radius is [GGS21]

$$r = \|y_1 - y_0\| \tan(\gamma), \quad (8)$$

where $\|y_1 - y_0\|$ is the distance between the camera and the primary hit, and the opening angle γ becomes a hyperparameter.

Our heuristic disables merges and connections at vertex y_i – and all subsequent ones – if $P(y'_i \in D_r(y_i)) < 1$. To achieve the desired behavior, we need to pick a suitable hyperparameter value γ . We start by simply setting $\gamma = 1^\circ$ and introducing a scaling factor S_r :

$$\pi r^2 p(y_i | y_{i-1}) = \pi S_r^2 \|y_1 - y_0\|^2 \tan^2(1^\circ) p(y_i | y_{i-1}) \quad (9)$$

So our final heuristic becomes

$$p(y_i | y_{i-1}) < \frac{1}{S_r^2 \pi \tan^2(1^\circ) \|y_1 - y_0\|^2}, \quad (10)$$

which we round to read

$$p(y_i | y_{i-1}) < \frac{1000}{S_r^2 \|y_1 - y_0\|^2} \quad (11)$$

We found $S_r = 2$ to perform well across all our scenes. Figure 3 shows how this parameter affects equal-time rendering quality and



Figure 3: Equal-time renderings with different values for our hyperparameter S_r . We found $S_r = 2$ to perform best on average across all our scenes. A too low value (e.g., 1) disables merge/connect too aggressively on glossy surfaces (see top row of crops for BOOKSHELF or bottom row for ROUGHGLASSESINDIRECT), while a too high value (e.g., 4) does not disable as many wasted operations as it could.

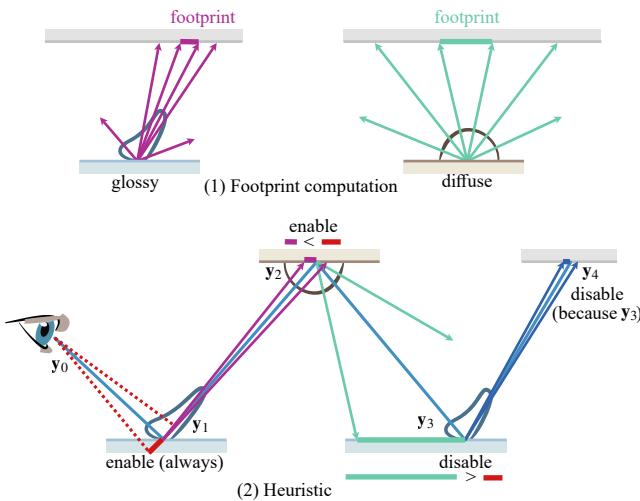


Figure 4: (1) We compute the sampling footprint of each vertex from the reciprocal of its (surface area) PDF. (2) At each vertex along a camera prefix \bar{y} our heuristic compares this footprint to the footprint of y_1 (scaled by a hyperparameter). Merges and connections are only performed until the first vertex that exceeds this threshold. Here, they are disabled at – and after – y_3 .

```

1 TraceCameraPath(Ray ray):
2     bool enableMergeConnect = true;
3     float threshold, lastSolidPdf;
4     for (int i = 0; i < maxDepth; ++i):
5         Hit hit = scene.Intersect(ray);
6         if (i == 0):
7             threshold = 1000/(Sr2 * d2); // Eq. (11)
8         else:
9             float lastAreaPdf = lastSolidPdf * cos/d2;
10            if (lastAreaPdf < threshold): // Eq. (11)
11                enableMergeConnect = false;
12            PerformNextEvent(...);
13            HandleDirectLightHit(...);
14            if (enableMergeConnect):
15                PerformMerging(...);
16                PerformConnections(...);
17                ray, lastSolidPdf = SampleDirection(...);

```

Figure 5: Pseudo-code showing how our heuristic is integrated into a typical bidirectional implementation. Not included are the necessary analog computations for the MIS weights.

sample count. If S_r is too low, then performance on rough surfaces suffers (see the ROUGHGLASSESINDIRECT scene) because our heuristic becomes too aggressive. If the value is too high, performance benefits diminish as we start to allow too many unnecessary operations.

Another footprint approximation is discussed in Appendix A with similar results. However, we stick to our approximation as it is easier to compute and works well with the correlation-aware MIS weights used in the evaluation.

3.2. Our method

The full process of our method is illustrated in Figure 4 (2), and the pseudo-code in Figure 5 sketches how it is integrated into a VCM implementation.

At the first hit after the camera, all techniques are always enabled. At each subsequent hit, we evaluate our heuristic based on the surface area PDF of sampling this hit point from the previous one. If merges and connections are disabled at a vertex, this decision remains in place for all subsequent camera path vertices: If the variance is high at any point along the path, then further continuing this path will only increase variance further.

3.3. Multiple Importance Sampling

For our method to be unbiased the following three conditions on the MIS weights must be satisfied [Vea97]:

- (1) $\forall x: f(x) \neq 0 \Rightarrow \exists t: p_t(x) \neq 0$.
For all paths x with a non-zero contribution there exists at least one technique t capable of constructing a valid path.
- (2) $\forall x, t: p_t(x) = 0 \Rightarrow w_t(x) = 0$.
For all techniques t unable to generate the corresponding path x , the associated weight must be zero.

$$(3) \forall x: f(x) \neq 0 \Rightarrow \sum_{t \in T} w_t(x) = 1.$$

For all paths x with a non-zero contribution the MIS weights sum to one.

Condition (3) is guaranteed by the balance heuristic, provided we compute its terms consistently. So we only need to worry about the other two conditions.

By construction, if there is any non-specular surface along a path, our heuristic will perform at least one merge / connect operation on the first such surface (since the footprint of specular events is zero). Additionally, we always perform unidirectional path tracing with next event estimation. Hence, condition (1) is satisfied.

For condition (2), we include our heuristic in the MIS weights:

$$w_t(\bar{x}) \propto \begin{cases} h(\bar{y}) n_t p_t(\bar{x}) & \text{if } t \in T_H, \\ n_t p_t(\bar{x}) & \text{if } t \notin T_H \end{cases} \quad (12)$$

That is, for all techniques $t \in T_H$ that are affected by our heuristic, we compute the binary value $h(\bar{y})$ of our heuristic and multiply it on the effective density in the balance heuristic. At the first hit after the camera the heuristic is $h_1 = 1$. At later vertices, the value is the product of all previous binary decisions and the current one (see Equation (11)),

$$h_i = h_{i-1} \cdot \begin{cases} 0, & \text{if } p(\mathbf{y}_i | \mathbf{y}_{i-1}) < \frac{1000}{S_r^2 \|\mathbf{y}_i - \mathbf{y}_0\|^2} \\ 1 & \end{cases} \quad (13)$$

Note that, from a practical perspective, this product can be easily computed incrementally while constructing paths and supports efficient MIS weight computation [Geo12].

3.4. Optional data-driven adaptation

The most challenging setup for our heuristic is multi-bounce paths via moderately glossy surfaces. An example is the ROUGHGLASS-ESINDIRECT scene in Figure 3. There, the rightmost glass (roughness 0.6) would benefit from additional merge operations at later bounces.

This inaccuracy only causes a moderate uptick in noise, and performance is still significantly above that of a unidirectional path tracer. Nevertheless, one might desire consistent improvements over the VCM baseline. We found that a simple pixel-variance estimation can be used to achieve this consistency by adaptively using our heuristic. For that, we estimate the inefficiency with and without our heuristic and, on a per-pixel basis, switch to the more efficient scheme.

The process is illustrated in Figure 6. We render the first iteration *without* our heuristic. During that iteration, we render an additional image: for every sample, we compute the hypothetical MIS weights of this sample if our heuristic was used. So, we effectively render an image with and one without our heuristic from a single pool of samples. The pixel variances of these images – with some generous filtering (box filter with radius 8) – provide us with variance estimates of the two options.

For the cost, we count the number of merge / connection operations that have been performed and how many of those our heuristic would have skipped. This is also done on a per-pixel basis.

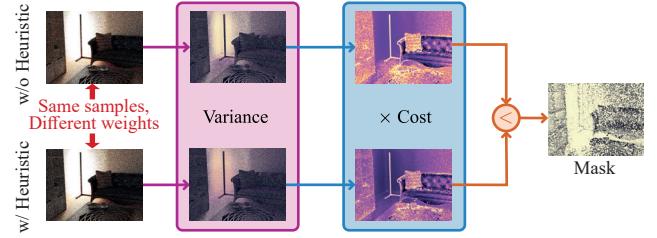


Figure 6: The computations done by our optional adaptation scheme in the first iteration. We compute two versions of MIS weights to obtain an image with and without our heuristic (left). The pixel variances of these are paired with cost statistics (center) to produce a per-pixel mask (right) of where our heuristic should be used.

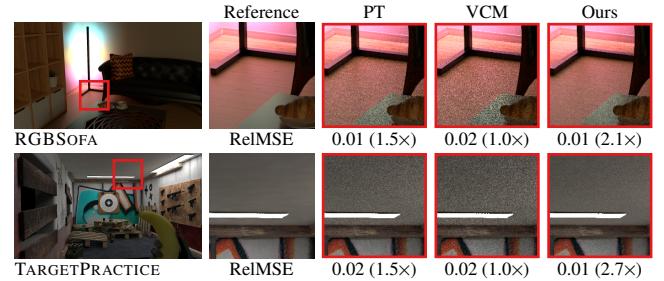


Figure 7: Equal-sample comparison (8 spp) showing that sometimes our heuristic outperforms the VCM baseline even in equal-sample. This arises from correlation-induced MIS weighting issues that we effectively correct more aggressively than the correlation-aware MIS weights [GPSK18].

With these quantities, we can then compute a per-pixel decision mask of where our heuristic should be used, simply by comparing the products of the (filtered) variance and cost estimates.

4. Evaluation

We integrated our heuristic into a VCM [GKDS12; HJ11] implementation with vertex caches [DKHS14a] and correlation-aware MIS [GGS21] in a CPU renderer. At the primary hit point, light tracing is performed instead of merging, to reduce bias. Source code will be provided on GitHub [Yaz25]. All timings are from an AMD Ryzen™ 9 3950X CPU. If not noted otherwise, results are from 30 second equal-time benchmarks.

Our main error metric is the relative mean-squared error (RelMSE), i.e., we divide each pixel's squared error by the squared reference value and then average those ratios:

$$\text{RelMSE} = \frac{1}{N} \sum_{i=1}^N \frac{\|P_i - \hat{P}_i\|^2}{\hat{P}_i^2 + \epsilon}, \quad (14)$$

with P_i and \hat{P}_i being the pixel estimate and reference value respectively. For numerical stability, we use an offset $\epsilon = 0.01$. For robustness against outliers, the 0.001% of pixels with highest error (around 10 at our resolution of 1280×720) are ignored in the sum

	VCM	EAMIS	Ours	OursAdaptive
1st. Iteration	Baseline	107%	56%	85%
nth. Iteration	Baseline	67%	56%	61%

Table 1: Cost relative to the baseline for the first-iteration and follow-up iterations. The training of EAMIS and OursAdaptive happens only in the first iteration, making the first iteration more expensive. The baseline and Ours have no training hence the first and follow-up iterations share the same number. The time measured includes performance gain due to rejecting merges and connections. Consequently, Ours is faster than the baseline, even while they share most of the same code. OursAdaptive does reject connections but not merges in the first iteration, as it trains for them. EAMIS does neither reject connections nor merges in the first iteration.

(this is only applied to full-image errors, not the zoom-ins). Other error metrics can be found in the supplemental materials.

We compare against two baselines: an unidirectional path tracer with next event estimation, and VCM with all merges and connections enabled and *four* connections at each camera path vertex. Section 4.4 discusses different numbers of connections. All our VCM results, unless explicitly noted otherwise, use correlation-aware MIS weights. We also compare to the per-pixel binary optimization from efficiency-aware MIS (EAMIS) [GYGS22]. Based on filtered second moment estimates from the first iteration, these are used to enable or disable connections or merges in a pixel. Their global optimizations (number of light paths, number of connections) are orthogonal; we leave them out to keep the evaluation focused.

4.1. Overhead

Our heuristic requires only a few multiplications, and a comparison based on readily available quantities. Hence, it does not introduce measurable overhead and only reduces the time per rendering iteration. This is not the case when used adaptively (see Section 3.4) or for the alternative EAMIS approach. For these, the first iteration, due to statistics accumulation and filtering, incurs additional cost. Table 1 quantifies this via the average cost of the first iteration and all other iterations, relative to the VCM baseline.

Compared to the purely data-driven EAMIS approach, our method yields a significantly shorter time-to-first-image. Even with our adaptive usage scheme, the overhead is still smaller, as the pixel-variance computation is much less involved than the moment-prediction scheme used by EAMIS. This could be appealing for progressive rendering applications. In later iterations, all three variants yield significant cost reduction compared to baseline VCM by skipping a substantial number of costly merge and connection operations.

4.2. Equal-sample performance

Ideally, disabling techniques should always result in worse equal-sample performance. However, due to MIS, this is not the case

as shown in Figure 7. The balance heuristic performs particularly poorly for the VCM algorithm [GGSK19; GGS21]. By disabling problematic techniques, we hence improve the equal-sample performance. Specifically, we aggressively remove techniques where the camera prefix could have high variance. Those are precisely the techniques that cause correlation-related MIS weighting issues [GGS21]. Indeed, for the same hyperparameter values, our heuristic is a lower-bound of the correlation-aware MIS weights. Because of this, we also do not need to apply correlation-aware MIS weighting in our integrator, further simplifying the implementation.

4.3. Equal-time performance

Figure 10 plots the equal-time error on ten representative test scenes. Figures 8 and 9 show the corresponding renderings for select scenes. The full set of results and HDR images can be found in the supplemental materials.

Depending on the scene, our method is generally either similar to or significantly better than baseline VCM. The fact that we can adapt per-vertex rather than per-pixel, and that we don't rely on noisy statistics gives us an edge over EAMIS, for example in the BOOKSHELF (too coarse) and LIVINGROOMVCM (noise) scenes.

Worst-case performance occurs with long paths across moderately glossy surfaces. We modeled the DRAGONINICE to provoke this issue. There, our heuristic disables some beneficial merging operations after glossy interactions. Adaptive use via variance estimation resolves most of this issue. An alternative could be to tune the hyperparameter S_r and allow more merges in this scene.

Also, there are scenes like DININGROOM, where all bidirectional sampling is wasted. Since our heuristic only disables the merge and connect operations, not the tracing of light paths, we cannot rival the unidirectional path tracer there. This could be addressed via orthogonal prior work to adapt the number of light paths [GYGS22].

4.4. Ablation: Number of connections

With a vertex cache approach [DKHS14a] we are free to choose the number of connections c to be performed at each camera path vertex. Our heuristic performs well independently of this number. Table 2 summarizes the speed-up due to our heuristic with different numbers of connections c . Our heuristic is beneficial for all values. The more connections we perform, the bigger the speed-up due to our heuristic. This is unsurprising, since a larger c implies a larger cost-savings potential for our heuristic. In all our evaluations we use $c = 4$ since this is closest to the number of connections a classic bidirectional path tracer – without vertex caching – would perform in our scenes.

5. Limitations and future work

As discussed in Section 3.4, for long paths via medium rough surfaces (neither diffuse nor highly glossy) our heuristic tends to be too aggressive. Adaptive use via variance estimates fixes this problem but introduces a training step which we would like to avoid in the future as it violates our original motivation and introduces non-negligible overhead for extremely short (< 10s) renderings.

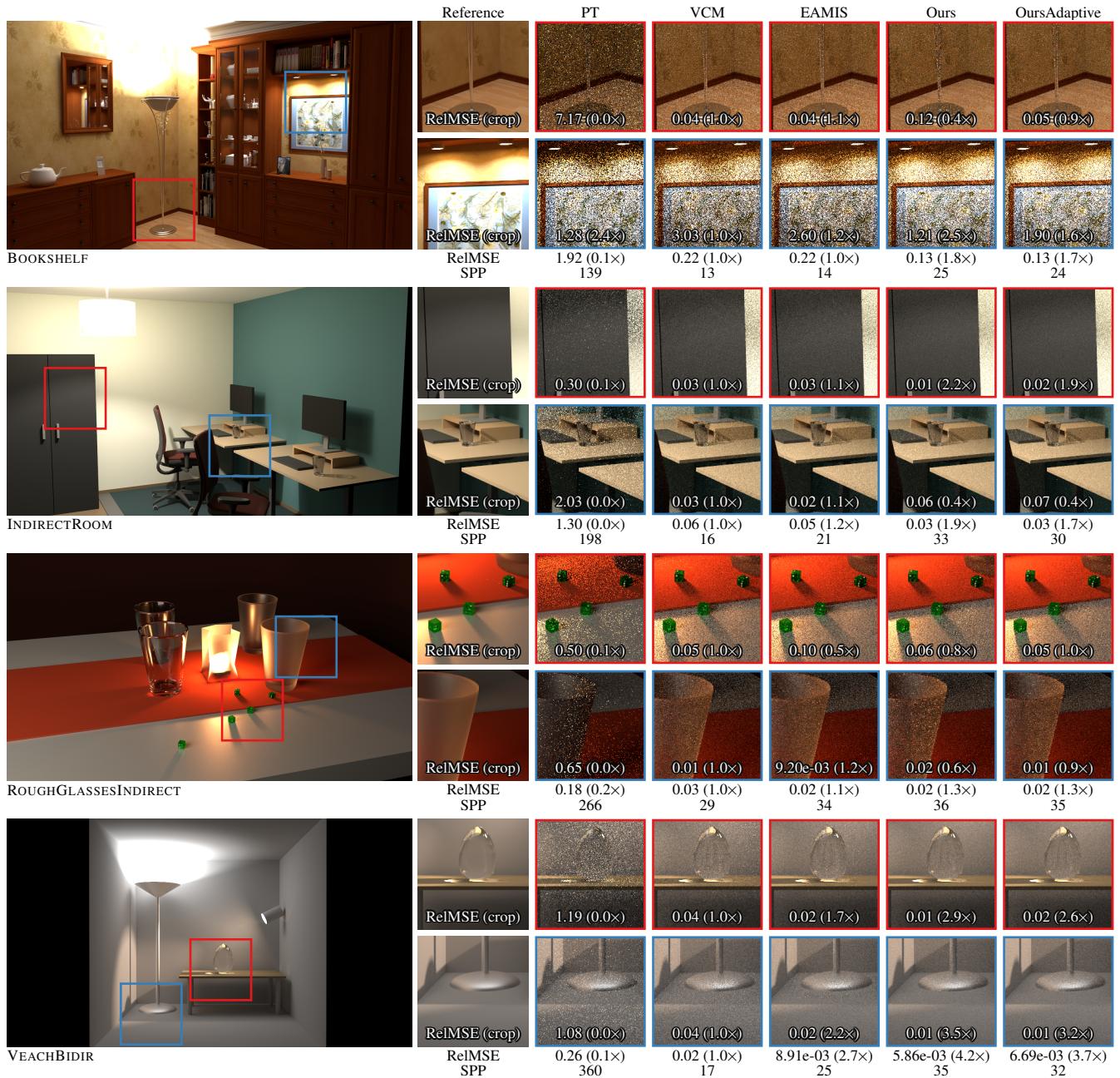


Figure 8: Equal-time (30s) comparisons. Our heuristic is most impactful in scenes featuring multi-bounce indirect illumination and long paths – because these have the highest number of vertices where computation is wasted by the baseline. However, performance on paths involving moderately glossy surfaces (BOOKSHELF, top row, and ROUGHGLASSESINDIRECT) can drop below the baseline. Adaptive use via variance estimates (rightmost column) can rectify this but adds implementation complexity and first-iteration overhead.

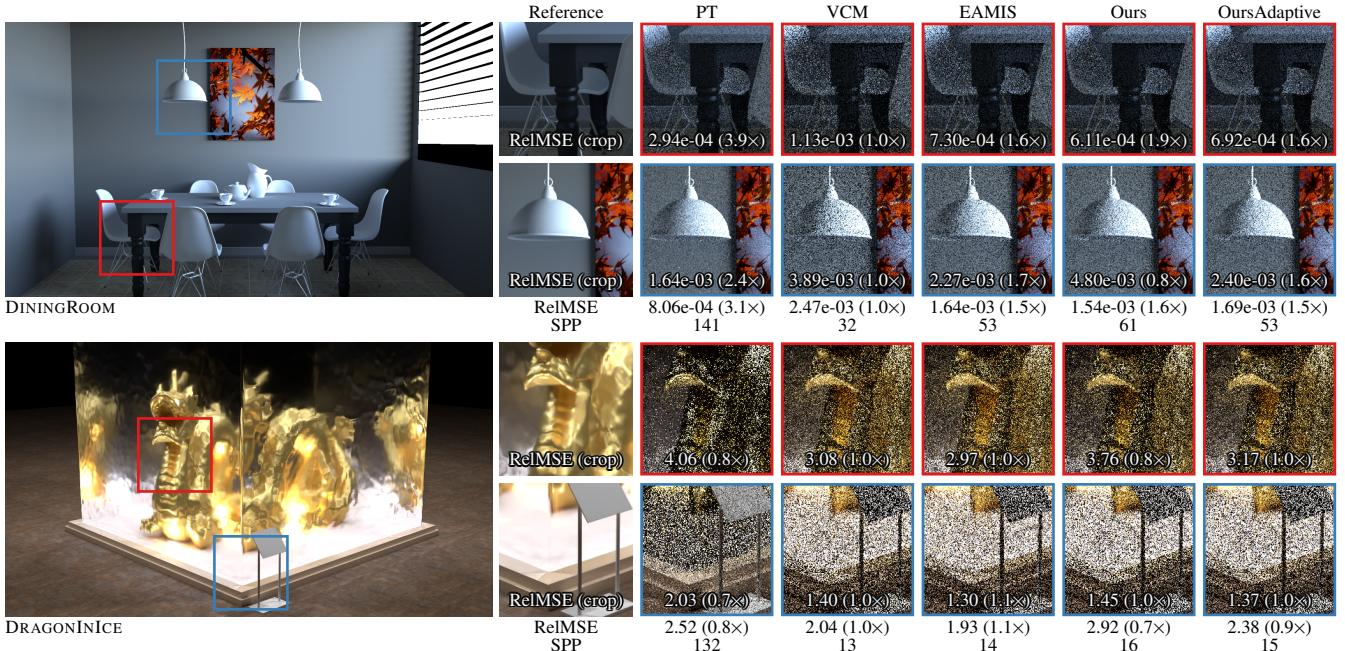


Figure 9: Equal-time (30s) results for the worst-case scenes we found. The DININGROOM is best rendered unidirectionally, any bidirectional samples are wasted. Our method improves matters there, but since we don't avoid tracing light paths we cannot rival the performance of unidirectional path tracing. The DRAGONINICE is modeled to exacerbate issues with long, moderately glossy paths.

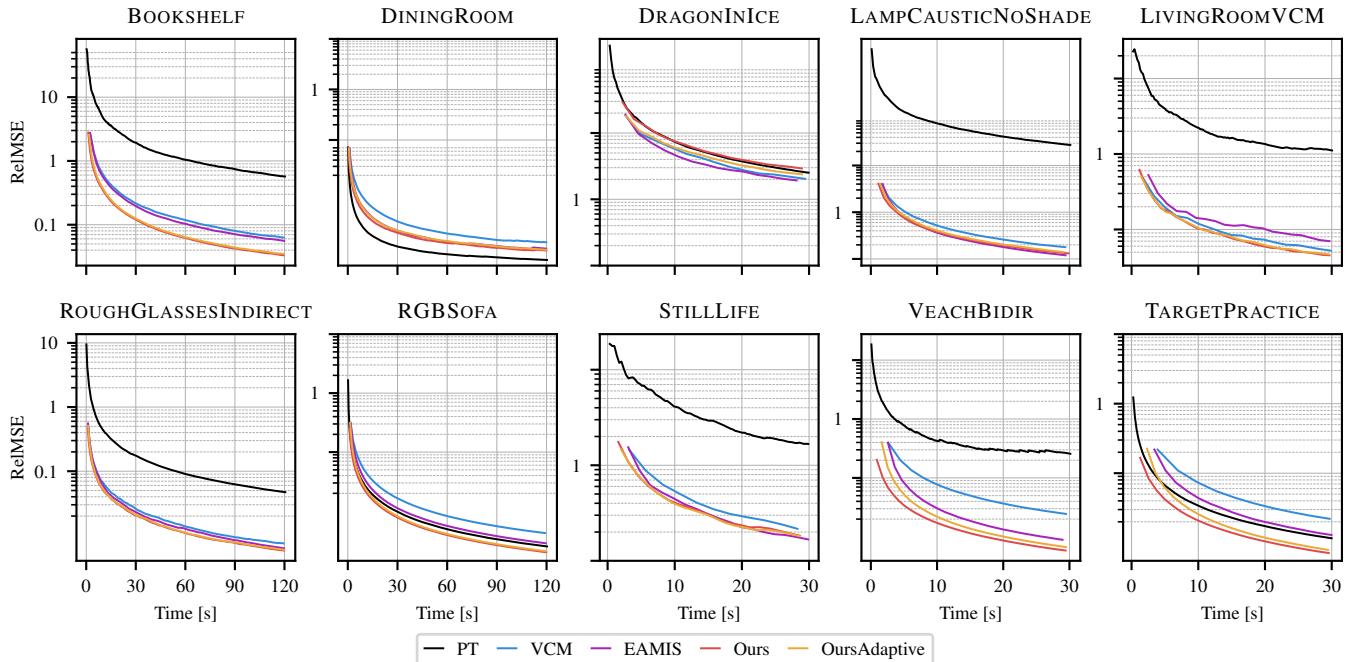


Figure 10: Convergence plots for a representative subset of our test scenes for 120s (first two columns) and 30s (last three columns). On average, our method performs considerably better than baseline VCM. We also outperform data-driven adaptation like EAMIS since we do not introduce overhead in the first iteration (visible in how much sooner we generated the first image in the plot) and apply a more fine-grained per-vertex control. In some scenes, like TARGETPRACTICE, we can even improve upon unidirectional path tracing (PT) despite baseline VCM performing worse than PT there.

	$c = 1$	2	4	8
BOOKSHELF	1.52×	1.57×	1.86×	1.96×
CORNELLBOX	2.98×	3.18×	3.23×	3.47×
ROUGHGLASSESINDIRECT	1.18×	1.31×	1.28×	1.19×
STILLLIFE	1.09×	1.16×	1.20×	1.11×
VEACHBIDIR	3.72×	3.84×	3.96×	4.13×

Table 2: Equal-time (30s) speedup of our heuristic compared to the VCM baseline for different numbers of connections c . In general, the more connections, the bigger the impact of our heuristic. The highlighted column corresponds to our default choice of four connections. This approximately matches the average length of the light paths reported over all scenes. Our choice of c resembles a “classic” bidirectional path tracer [VG95a] which connects all vertices to each other.

Earlier rejection. By design, our heuristic can only skip merges and connections after the first surface scattering event. Often, these operations are also unnecessary at the primary hit point. We did not find a way to skip these primary-hit techniques with only a footprint-based approach. Future work could look into also eliminating these operations, either with an extended heuristic, or via a data-driven approach.

Data-driven solutions. A core motivation of our heuristic was to avoid data-driven solutions if at all possible. However, while data-driven solutions struggle with noisy estimates and introduce noteworthy overhead, they seem to also be the only way to achieve optimal results. Therefore, exploring lightweight yet effective data-driven options remains an important and interesting direction for future work. For example, one could extend the fixed-point optimization of prior work [RYS23; MRYS24] to support photon mapping and vertex-cache-based connections. This encounters two challenges: (1) The severe sample correlation of merging might have to be accounted for. (2) VCM iterations are quite costly, so we need the adapted sampling decisions to converge to reasonable values as quickly as possible. For this latter quick convergence, a combined solution with our heuristic might come in handy.

Volume rendering. We focused on the rendering of surface scattering. Volumetric rendering also benefits from bidirectional sampling [KGH*14]. Since volumetric paths are often even longer than surface-based ones, a heuristic like ours should benefit volumetric rendering even more. To extend our idea to volumes, the footprint metric will have to be adapted or redesigned. In the context of volumetric rendering, even more sampling techniques are available [KGH*14]. So a more fine-grained heuristic, e.g., disabling only point-based but not beam-based techniques, might offer further benefits.

6. Conclusion

We propose an easy-to-compute, renderer-agnostic heuristic to improve the efficiency of bidirectional rendering algorithms by skipping unnecessary connection and merging operations. We show that a simple footprint-based heuristic can be used to this end, with

no precomputation or statistics gathering required. While not optimal, our heuristic yields significant improvements over the baseline in many cases and comes closest to consistently outperforming unidirectional path tracing. If consistent improvements over baseline VCM are desired, we show that a simple variance estimation scheme can be employed to adaptively use our heuristic and achieve that goal, too. We consider our heuristic a promising ingredient towards making bidirectional algorithms competitive with unidirectional path tracing as a default choice for the rendering algorithm.

7. Acknowledgements

We wish to thank Vladislav Hnatovskiy and Mira Niemann for the TARGET PRACTICE scene and Wig42 for the DINING ROOM scene. This project was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 540768663.

References

- [BM97] BOLIN, MARK R and MEYER, GARY W. “An error metric for Monte Carlo ray tracing”. *Rendering Techniques (Eurographics Workshop on Rendering)*. Springer. 1997, 57–68. DOI: [10.1007/978-3-7091-6858-5_6](https://doi.org/10.1007/978-3-7091-6858-5_6).
- [BSC*03] BEKAERT, PHILIPPE, SLUSALLEK, PHILIPP, COOLS, RONALD, et al. *A custom designed density estimation method for light transport*. Research Report MPI-I-2003-4-004. Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany: Max-Planck-Institut für Informatik, Sept. 2003 [10, 11](#).
- [DKHS14a] DAVIDOVIĆ, TOMÁŠ, KŘIVÁNEK, JAROSLAV, HAŠAN, MILOŠ, and SLUSALLEK, PHILIPP. “Progressive Light Transport Simulation on the GPU: Survey and Improvements”. *ACM Trans. Graph.* 33.3 (June 2014). ISSN: 0730-0301. DOI: [10.1145/2602144.5.6](https://doi.org/10.1145/2602144.5.6).
- [DKHS14b] DAVIDOVIĆ, TOMÁŠ, KŘIVÁNEK, JAROSLAV, HAŠAN, MILOŠ, and SLUSALLEK, PHILIPP. “Progressive light transport simulation on the GPU: Survey and improvements”. *ACM Trans. Graph.* 33.3 (2014). DOI: [10.1145/2602144.2](https://doi.org/10.1145/2602144.2).
- [GEE20] GUO, JERRY JINFENG, EISEMANN, MARTIN, and EISEMANN, ELMAR. “Next Event Estimation++: Visibility Mapping for Efficient Light Transport Simulation”. *Computer Graphics Forum (Proc. of Pacific Graphics PG)* 39.7 (Nov. 2020), 205–217. DOI: [10.1111/cgf.14138.3](https://doi.org/10.1111/cgf.14138.3).
- [Geo12] GEORGIEV, ILIYAN. *Implementing Vertex Connection and Merging*. Tech. rep. Saarland University, 2012. URL: <http://www.iliyan.com/publications/ImplementingVCM5.11>.
- [GGS21] GRITTMANN, PASCAL, GEORGIEV, ILIYAN, and SLUSALLEK, PHILIPP. “Correlation-Aware Multiple Importance Sampling for Bidirectional Rendering Algorithms”. *Comput. Graph. Forum (EG 2021)* 40.2 (2021), 231–238. DOI: [10.1111/cgf.142628.2,3,5,6](https://doi.org/10.1111/cgf.142628.2,3,5,6).
- [GGSK19] GRITTMANN, PASCAL, GEORGIEV, ILIYAN, SLUSALLEK, PHILIPP, and KŘIVÁNEK, JAROSLAV. “Variance-Aware Multiple Importance Sampling”. *ACM Trans. Graph. (SIGGRAPH Asia 2019)* 38.6 (2019). DOI: [10.1145/3355089.3356515.2,6](https://doi.org/10.1145/3355089.3356515.2,6).
- [GKDS12] GEORGIEV, ILIYAN, KŘIVÁNEK, JAROSLAV, DAVIDOVIĆ, TOMÁŠ, and SLUSALLEK, PHILIPP. “Light transport simulation with vertex connection and merging.” *ACM Trans. Graph. (SIGGRAPH Asia 2012)* 31.6 (2012). DOI: [10.1145/2366145.2366211.1-3,5](https://doi.org/10.1145/2366145.2366211.1-3,5).
- [GPSK18] GRITTMANN, PASCAL, PÉRARD-GAYOT, ARSÈNE, SLUSALLEK, PHILIPP, and KŘIVÁNEK, JAROSLAV. “Efficient Caustic Rendering with Lightweight Photon Mapping”. *Comput. Graph. Forum (EGSR '18)* 37.4 (2018), 133–142. DOI: [10.1111/cgf.13481.2,3,5](https://doi.org/10.1111/cgf.13481.2,3,5).

- [GYGS22] GRITTMANN, PASCAL, YAZICI, ÖMERCAN, GEORGIEV, ILYAN, and SLUSALLEK, PHILIPP. “Efficiency-Aware Multiple Importance Sampling for Bidirectional Rendering Algorithms”. *ACM Trans. Graph. (SIGGRAPH 2022)* 41.4 (2022). DOI: [10.1145/3528223.3530126](https://doi.org/10.1145/3528223.3530126) 1–6.
- [Hec90] HECKBERT, PAUL S. “Adaptive radiosity textures for bidirectional ray tracing”. *SIGGRAPH Comput. Graph.* 24.4 (Sept. 1990), 145–154. ISSN: 0097-8930. DOI: [10.1145/97880.97895.3](https://doi.org/10.1145/97880.97895.3).
- [HGS23] HUA, QINGQIN, GRITTMANN, PASCAL, and SLUSALLEK, PHILIPP. “Revisiting Controlled Mixture Sampling for Rendering Applications”. *ACM Trans. Graph. (SIGGRAPH 2023)* 42.4 (2023). DOI: [10.1145/3592435.2](https://doi.org/10.1145/3592435.2).
- [HJ11] HACHISUKA, TOSHIYA and JENSEN, HENRIK WANN. “Robust adaptive photon tracing using photon path visibility”. *ACM Trans. Graph. (TOG)* 30.5 (2011), 114. DOI: [10.1145/2019627.2019633](https://doi.org/10.1145/2019627.2019633) 3, 5.
- [HJJ10] HACHISUKA, TOSHIYA, JAROSZ, WOJCIECH, and JENSEN, HENRIK WANN. “A progressive error estimation framework for photon density estimation”. *ACM Trans. Graph. (SIGGRAPH 2010)* 29.6 (2010). DOI: [10.1145/1882261.1866170](https://doi.org/10.1145/1882261.1866170) 3.
- [HOJ08] HACHISUKA, TOSHIYA, OGAKI, SHINJI, and JENSEN, HENRIK WANN. “Progressive photon mapping”. *ACM Trans. Graph. (SIGGRAPH Asia 2008)*. Vol. 27. 5. ACM. 2008. DOI: [10.1145/1409060.1409083](https://doi.org/10.1145/1409060.1409083).
- [HPJ12] HACHISUKA, TOSHIYA, PANTALEONI, JACOPO, and JENSEN, HENRIK WANN. “A path space extension for robust light transport simulation”. *ACM Trans. Graph. (SIGGRAPH Asia 2012)* 31.6 (2012). DOI: [10.1145/2366145.2366210](https://doi.org/10.1145/2366145.2366210) 1, 2.
- [Jen96] JENSEN, HENRIK WANN. “Global illumination using photon maps”. *Rendering Techniques (Eurographics Workshop on Rendering)*. Springer, 1996, 21–30. DOI: [10.5555/275458.275461](https://doi.org/10.5555/275458.275461) 2, 3.
- [JG18] JENDERSIE, J. and GROSCH, T. “An improved multiple importance sampling heuristic for density estimates in light transport simulations”. *Proceedings of the Eurographics Symposium on Rendering: Experimental Ideas & Implementations. SR ’18*. Karlsruhe, Germany: Eurographics Association, 2018, 65–72. DOI: [10.2312/sre.20181173](https://doi.org/10.2312/sre.20181173) 2.
- [Kaj86] KAJIYA, JAMES T. “The Rendering Equation”. *Computer Graphics (SIGGRAPH 1986)* 20.4 (1986), 143–150. ISSN: 0097-8930. DOI: [10.1145/15886.15902](https://doi.org/10.1145/15886.15902) 2.
- [Kel97] KELLER, ALEXANDER. “Instant radiosity”. *Annual Conference Series (SIGGRAPH 1997)*. 1997, 49–56. DOI: [10.1145/258734.258769](https://doi.org/10.1145/258734.258769) 3.
- [KGH*14] KRIVÁNEK, JAROSLAV, GEORGIEV, ILYAN, HACHISUKA, TOSHIYA, et al. “Unifying Points, Beams, and Paths in Volumetric Light Transport Simulation”. *ACM Trans. Graph. (SIGGRAPH 2014)* 33.4 (2014). DOI: [10.f6cz72](https://doi.org/10.f6cz72) 9.
- [KŠV*19] KARLÍK, ONDŘEJ, ŠIK, MARTIN, VÉVODA, PETR, et al. “MIS Compensation: Optimizing Sampling Techniques in Multiple Importance Sampling”. *ACM Trans. Graph. (SIGGRAPH Asia 2019)* 38.6 (2019). DOI: [10.1145/3355089.3356565](https://doi.org/10.1145/3355089.3356565) 2.
- [KVG*19] KONDAPANENI, IVO, VÉVODA, PETR, GRITTMANN, PASCAL, et al. “Optimal Multiple Importance Sampling”. *ACM Trans. Graph. (SIGGRAPH 2019)* 38.4 (2019). DOI: [10.1145/3306346.3323009](https://doi.org/10.1145/3306346.3323009) 2.
- [LPG13] LU, HEQI, PACANOWSKI, ROMAIN, and GRANIER, XAVIER. “Second-Order Approximation for Variance Reduction in Multiple Importance Sampling”. *Comput. Graph. Forum (Pacific Graphics 2013)* 32.7 (2013), 131–136. DOI: [10.1111/cgf.12220](https://doi.org/10.1111/cgf.12220) 2.
- [LW93] LAFORTUNE, ERIC P. and WILLEMS, YVES D. “Bi-Directional Path Tracing”. 93 (1993), 145–153. DOI: [10.1145/1409060.1409083](https://doi.org/10.1145/1409060.1409083).
- [MMK*21] MAJERCIK, ZANDER, MUELLER, THOMAS, KELLER, ALEXANDER, et al. “Dynamic Diffuse Global Illumination Resampling”. *ACM SIGGRAPH 2021 Talks. SIGGRAPH ’21. Virtual Event, USA: Association for Computing Machinery*, 2021. ISBN: 9781450383738. DOI: [10.1145/3450623.3464635](https://doi.org/10.1145/3450623.3464635) 10.
- [MRNK21] MÜLLER, THOMAS, ROUSSELLE, FABRICE, NOVÁK, JAN, and KELLER, ALEXANDER. “Real-time neural radiance caching for path tracing”. *ACM Trans. Graph.* 40.4 (July 2021). ISSN: 0730-0301. DOI: [10.1145/3450626.3459812](https://doi.org/10.1145/3450626.3459812) 10.
- [MRYS24] MEYER, JOSHUA, RATH, ALEXANDER, YAZICI, ÖMERCAN, and SLUSALLEK, PHILIPP. “MARS: Multi-sample Allocation through Russian roulette and Splitting”. *SIGGRAPH Asia 2024 Conference Papers. SA ’24. Association for Computing Machinery*, 2024. ISBN: 9798400711312. DOI: [10.1145/3680528.3687636](https://doi.org/10.1145/3680528.3687636) 2, 9.
- [Mül19] MÜLLER, THOMAS. ““Practical Path Guiding” in Production”. *ACM SIGGRAPH Courses: Path Guiding in Production, Chapter 10*. Los Angeles, California: ACM, 2019, 18:35–18:48. DOI: [10.1145/3305366.3328091](https://doi.org/10.1145/3305366.3328091) 2.
- [NID20] NABATA, KOSUKE, IWASAKI, KEI, and DOBASHI, YOSHINORI. “Resampling-aware Weighting Functions for Bidirectional Path Tracing Using Multiple Light Sub-Paths”. *ACM Trans. Graph.* 39.2 (2020). DOI: [10.1145/3338994](https://doi.org/10.1145/3338994) 2.
- [PBPP10] PAJOT, ANTHONY, BARTHE, LOIC, PAULIN, MATHIAS, and POULIN, PIERRE. “Representativity for robust and adaptive multiple importance sampling”. *IEEE Trans. Vis. Comput. Graph* 17.8 (2010), 1108–1121. DOI: [10.1109/TVCG.2010.230](https://doi.org/10.1109/TVCG.2010.230) 2.
- [PRDD15] POPOV, STEFAN, RAMAMOORTHI, RAVI, DURAND, FREDO, and DRETTAKIS, GEORGE. “Probabilistic connections for bidirectional path tracing”. *Comput. Graph. Forum (EGSR 2015)*. Vol. 34. 4. Wiley Online Library. 2015, 75–86. DOI: [10.5555/2858834.2858843](https://doi.org/10.5555/2858834.2858843) 2.
- [RYS23] RATH, ALEXANDER, YAZICI, ÖMERCAN, and SLUSALLEK, PHILIPP. “Focal Path Guiding for Light Transport Simulation”. New York, NY, USA: Association for Computing Machinery, 2023. DOI: [10.1145/3588432.3591543](https://doi.org/10.1145/3588432.3591543) 9.
- [SHS19] SBERT, MATEU, HAVRAN, VLASTIMIL, and SZIRMAY-KALOS, LÁSZLÓ. “Optimal Deterministic Mixture Sampling”. *Eurographics (Short Papers)*. 2019, 73–76. DOI: [10.2312/egs.20191018](https://doi.org/10.2312/egs.20191018) 2.
- [SLW22] SU, FUJIA, LI, SHENG, and WANG, GUOPING. “SPCBPT: subspace-based probabilistic connections for bidirectional path tracing”. *ACM Trans. Graph. (SIGGRAPH 2022)* 41.4 (2022). DOI: [10.1145/3528223.3530183](https://doi.org/10.1145/3528223.3530183) 2.
- [Vea97] VEACH, ERIC. *Robust Monte Carlo methods for light transport simulation*. Stanford University PhD thesis, 1997. DOI: [10.5555/927297](https://doi.org/10.5555/927297) 2, 4.
- [VG95a] VEACH, ERIC and GUIBAS, LEONIDAS. “Bidirectional estimators for light transport”. *Photorealistic Rendering Techniques*. Springer, 1995, 145–167. DOI: [10.1007/978-3-642-87825-1_11](https://doi.org/10.1007/978-3-642-87825-1_11) 1, 2, 9.
- [VG95b] VEACH, ERIC and GUIBAS, LEONIDAS. “Optimally Combining Sampling Techniques for Monte Carlo Rendering”. *SIGGRAPH 1995*. ACM. 1995, 419–428. DOI: [10.1145/218380.218498](https://doi.org/10.1145/218380.218498) 2.
- [Yaz25] YAZICI, ÖMERCAN. *Implementation of Footprint-driven cost reduction for bidirectional rendering*. 2025. URL: <https://github.com/PearCoding/FootprintVCM> (visited on 06/04/2025) 5.

Appendix A: Using other footprint measures

Various footprint approximations have been proposed in prior work and used for multiple purposes. For example, some radiance caching methods [MRNK21; MMK*21] have adopted a footprint metric designed to control the photon mapping kernel [BSC*03]

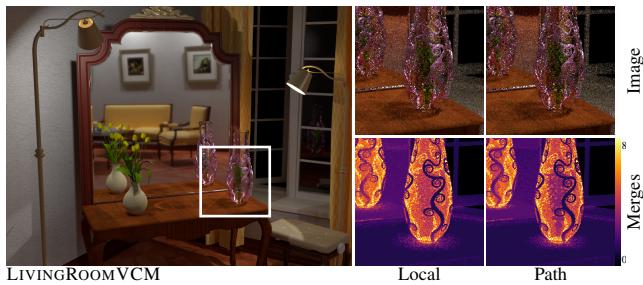


Figure 11: Equal-sample (8 spp) comparison of our local footprint and the path footprint based on [BSC*03]. Both approaches produce a similar result, while ours is easier to compute and does not require a provably costly square root for computation. Additionally, we show the number of merges as a normalized AOV to illustrate the similarity.

and reported good results. This footprint formulation reads:

$$a_0 = \frac{||\mathbf{x}_1 - \mathbf{x}_0||^2}{4\pi \cos \theta_1} \quad (15)$$

$$a_i = \left(\sum_{j=2}^i \frac{1}{\sqrt{p(\mathbf{x}_{j-1} \rightarrow \mathbf{x}_j)}} \right)^2, \quad (16)$$

where a_i is the footprint at the i th vertex. This metric was used in the context of radiance caching to terminate paths when $a_i > c \cdot a_0$, for a hyperparameter $c = 0.01$. This usage of footprints is conceptually different from our heuristic: They decide when to use the cached quantities, while we decide when to stop using costly techniques.

This footprint measure is mathematically quite similar to ours. The first difference is that they include the cosine at the first hit. We found that this causes issues on surfaces seen at a grazing angle and hence do not include it. The second difference is that they compute a global, full-path footprint by summing over the reciprocal square-roots of the surface area PDFs. We instead compute a simpler local footprint and propagate the termination decision along the path.

Figure 11 compares the average number of merge operations in each pixel with our local footprint and when replacing it by the above, full-path footprint. Results are very similar, except that the full-path version is slightly more aggressive, which is undesirable.

Beyond that, our local formulation is also easier to include in MIS weight computations, especially when the weights are computed in an incremental fashion [Geo12].