

Two-stage Resampling for Bidirectional Path Tracing with Multiple Light Sub-paths

K. Nabata¹ and K. Iwasaki^{1,3}  Y. Dobashi^{2,3} 

¹Wakayama University, Japan, ²Hokkaido University, Japan, ³Prometech CG Research, Japan

Abstract

Recent advances in bidirectional path tracing (BPT) reveal that the use of multiple light sub-paths and the resampling of a small number of these can improve the efficiency of BPT. By increasing the number of pre-sampled light sub-paths, the possibility of generating light paths that provide large contributions can be better explored and this can alleviate the correlation of light paths due to the reuse of pre-sampled light sub-paths by all eye sub-paths. The increased number of pre-sampled light sub-paths, however, also incurs a high computational cost. In this paper, we propose a two-stage resampling method for BPT to efficiently handle a large number of pre-sampled light sub-paths. We also derive a weighting function that can treat the changes in path probability due to the two-stage resampling. Our method can handle a two orders of magnitude larger number of pre-sampled light sub-paths than previous methods in equal-time rendering, resulting in stable and better noise reduction than state-of-the-art methods.

CCS Concepts

- Computing methodologies → Computer graphics; Ray tracing;

1. Introduction

Bidirectional path tracing (BPT) with multiple importance sampling (MIS) is a robust light transport algorithm used in physically-based rendering. Traditional BPT iteratively generates a single light sub-path from the light source and a single eye sub-path from the camera, and deterministically connects each vertex of the eye and light sub-paths to generate full light paths. Unfortunately, the traditional BPT often generates full light paths that give little contribution since it ignores both the visibility between the vertices of the two sub-paths and the BSDFs at the vertices.

Recent developments in BPT [PRDD15, NID20] have demonstrated that the use of multiple light sub-paths and the resampling of a small number of light sub-paths from them taking into account the contribution of each connection can improve the rendering efficiency (i.e., the noise reduction). Since the pre-sampled light sub-paths are shared and reused by all the eye sub-paths, preparing a small number of pre-sampled light sub-paths leads to correlation of the light sub-paths, resulting in artifacts. On the other hand, using a large number of pre-sampled light sub-paths increases the computation time for the resampling process, making the possible iteration count smaller, resulting in noise.

To address this problem, we propose a novel two-stage resampling method for BPT using multiple light sub-paths. In the first stage, important light sub-paths are quickly resampled from a large number of pre-sampled light sub-paths. In the second stage, the ordinary resampling method is used to choose a single light sub-path

from the light sub-paths obtained in the first stage. We also propose a weighting function tailored to two-stage resampling. By combining our weighting function with this two-stage resampling method, our method can reduce the variance and achieve stable noise reduction compared with previous BPT methods using multiple light sub-paths [PRDD15, NID20].

The contributions of our method can be summarized as follows:

- we propose a novel two-stage resampling estimator and formulate its variance.
- we apply two-stage resampling to BPT so that it can handle a large number of light sub-paths without incurring high computational cost.
- we devise a weighting function capable of treating the changes in the probability of the light path due to two-stage resampling.

2. Related Work

Bidirectional Path Tracing: BPT iteratively samples sub-paths from both the camera and the light source for each pixel. The pixel intensity is estimated by using the connected full light paths weighted by MIS weighting functions [VG94, VG95, LW93]. Several methods aim to improve the connections between sub-paths by using all pairs of sub-paths across the entire image [PBPP11], re-using light sub-paths using light vertex caching [DKHS14], and re-ordering sub-paths [CBH*18]. These methods, however, do not consider the contribution of the connections (e.g., the visibilities

between the connected vertices), resulting in the generation of full light paths with little contribution.

Probabilistic connections for BPT (PCBPT) provides importance-sampled connections between the vertices of sub-paths [PRDD15]. PCBPT prepares multiple light sub-paths shared by all the eye sub-paths and then resamples a small number of connections through importance sampling. Tokuyoshi and Harada proposed Hierarchical Russian Roulette (HRR) for efficient connections of specular-diffuse-glossy or glossy-diffuse-glossy paths [TH19]. Unfortunately, the MIS weighting functions used in PCBPT and HRR remain unaware of the change in the path probability caused by resampling.

Nabata et al. built upon PCBPT and proposed a resampling-aware weighting function for BPT using multiple light sub-paths [NID20]. Although this MIS weighting function can handle the change in probability due to resampling and can be more effective when a large number of light sub-paths are used, this method shares the same problem as PCBPT in that the number of pre-sampled light sub-paths is limited to several hundreds due to the overhead of constructing a probability mass function (pmf) for resampling. We address this problem by introducing a two-stage resampling technique.

Many-light Methods and Photon Mapping: Many-light rendering [Kel97, DKH^{*}14] uses multiple light sub-paths the vertices of which are connected to the last vertex of each eye sub-path whose length is restricted to one, which can be considered as a variant of BPT using a single path sampling technique. Georgiev et al. proposed importance caching that resamples a small number of light sub-path vertices by using cached pmfs [GKPS12], which is later extended to PCBPT. While our method also uses cached pmfs for resampling, importance caching is based on many-light rendering that can suffer from splotches due to the singularity of the geometry term. Several methods have been proposed to extend the eye sub-path length [KK04, WKB12] and these methods weigh the sampled full path using MIS. The MIS weighting functions used in these methods are designed to reduce the bias, which is different from our aim of reducing variance. Photon mapping (PM) also uses multiple light sub-paths shared by all the eye sub-paths. PM can be combined with BPT as an additional path sampling technique [GKDS12, HPJ12] to improve the robustness. Since our method is an extension of BPT, our method can be combined with PM via MIS.

Path guiding: Path guiding is an adaptive importance sampling method that learns the distributions of the incoming radiance or the importance prior to sampling. The precomputed distributions are represented by a Gaussian mixture model (GMM) [VKv^{*}14, HEV^{*}16], an adaptive tree structure [MGN17], or deep neural networks [MMR^{*}19, BMDS19]. Since these methods are orthogonal to our work, path guiding can be combined with our method for efficient sampling of sub-paths.

MIS weighting function: In recent years, increasing attention has been paid to MIS weighting functions. Kondapaneni et al. [KVG^{*}19] presented optimal MIS weighting functions by lifting the unnecessary assumption of the non-negativity of the weighting function. Karlik et al. [KŠV^{*}19] optimized the pdf of the one

sampling technique to reduce the variance. Although these methods significantly improve the efficiency of MIS, the application to complex light transport algorithms (e.g., BPT) remains challenging. Grittman et al. [GGSK19] proposed variance-aware MIS weights that can be easily incorporated into the traditional BPT. This method, however, is not applied to BPT using multiple light sub-paths.

Resampled importance sampling and Two-stage Sampling: Talbot et al. introduced resampled importance sampling (RIS) for direct illumination computation of image-based lighting [TCE05]. We build upon RIS and briefly review RIS in Sec. 3.1. Recently, Bitterli et al. presented a reservoir resampling method with spatiotemporal reuse [BWP^{*}20]. This method is currently applied to real-time rendering of direct illumination from a massive amount of light sources, and adaptation to BPT remains challenging. Cline et al. proposed a two-stage importance sampling method for image-based lighting [CETC06]. This method employs a two-stage approach to efficiently generate samples proportional to the product of lighting and BRDF, which is different from our purpose to handle a large number of samples efficiently.

3. Background

3.1. Resampled Importance Sampling

Let us consider estimating the integral $F = \int f(x)dx$ using Monte Carlo integration. The integral F is estimated using the following N -sample estimator:

$$F \approx \langle F \rangle^N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)},$$

where X_i is the i -th sample drawn from the probability density function (pdf) $p(x)$, and N is the number of samples. If the pdf $p(x)$ is closely proportional to the integrand f , the variance of the estimator $\langle F \rangle^N$ is low. In general, however, it is difficult to find a pdf that is close to f and is easy to sample since the pdf p should be normalized. Thus the pdf p is usually designed to be partially proportional to the integrand in practice.

Hereinafter, we explain the estimators by assuming the number of samples N to be one for brevity, and do not denote the number of samples in the superscript of the estimator. The N -sample estimator is simply calculated by drawing N samples and averaging.

Resampled importance sampling (RIS) [Tal05] can draw samples proportional to any target distribution \hat{q} , which can be (a part of) the integrand but not necessarily normalized. RIS first generates M samples (referred to as *proposals*) $\{X_1, \dots, X_M\}$ from the pdf p , which is readily sampled. Then RIS draws samples from the set of proposals $\mathbf{X} = \{X_1, \dots, X_M\}$ based on a probability mass function (pmf) proportional to the target distribution \hat{q} . The (one-sample) RIS estimator $\langle F \rangle_{\text{ris}}$ is calculated using:

$$\begin{aligned} \langle F \rangle_{\text{ris}} &= \frac{f(X)}{\hat{q}(X)} \left(\frac{1}{M} \sum_{j=1}^M \frac{\hat{q}(X_j)}{p(X_j)} \right) = \frac{1}{M} \frac{f(X)}{p(X)} \left(\frac{\sum_{j=1}^M \frac{\hat{q}(X_j)}{p(X_j)}}{\frac{\hat{q}(X)}{p(X)}} \right) \\ &= \frac{1}{M} \frac{f(X)}{p(X) P_r(X|\mathbf{X})}, \end{aligned} \quad (1)$$

Table 1: Notations. Subscripts i and j represent the index of the sample. Subscripts s and t indicate the number of vertices for light and eye sub-paths, respectively. Concatenations of variables for paths (e.g., $\bar{y}\bar{z}$, $\bar{y}\bar{Z}_t$) indicate the full path connecting the last vertices of two paths.

symbol	meaning
f	measurement contribution function
$\bar{x}, \bar{y}, \bar{z}$	full light path $x_0 \dots x_k$, light and eye sub-paths
M_1, M_2	number of light sub-path samples in each stage
A, A^s	scene surface, s -dimensional product of A
$\hat{q}_1(\bar{x}), \hat{q}_2(\bar{x})$	target distributions (a part of f)
$q_1(\bar{x}), q_2(\bar{x})$	target pdfs (normalize \hat{q}_1 and \hat{q}_2)
Q_1, Q_2	normalization factors (integrate \hat{q}_1 and \hat{q}_2)
\bar{Y}_n	set of light sub-paths in n -th iteration
\bar{Y}_c	subset of \bar{Y}_n stored at cache point c

where P_r is the pmf for sampling X from the set of proposals \mathbf{X} :

$$P_r(X|\mathbf{X}) = \frac{\hat{q}(X)/p(X)}{\sum_{j=1}^M \hat{q}(X_j)/p(X_j)}. \quad (2)$$

The variance of the RIS estimator $V[\langle F \rangle_{\text{ris}}]$ is given by (the arguments are omitted for simplicity):

$$V[\langle F \rangle_{\text{ris}}] = \frac{1}{M} V\left[\frac{f}{p}\right] + \left(1 - \frac{1}{M}\right) V\left[\frac{f}{q}\right], \quad (3)$$

where q is the pdf of the target distribution $\hat{q}(x)$ (i.e., $q(x) = \hat{q}(x) / \int \hat{q}(x) dx$), referred to as the *target pdf*. The variance of the RIS estimator $V[\langle F \rangle_{\text{ris}}]$ is a blend of two variance terms $V[f/p]$ and $V[f/q]$ with the reciprocal of the number of the proposals M being the blending factor. This indicates that **the distribution of the resampled sample X is a blend of the sampling pdf p and the target pdf q , and the distribution approaches the target pdf q as the number of proposals M increases**.

3.2. Bidirectional Path Tracing (BPT)

Veach formulated the light transport problem as a path integral formulation [VG94], where the pixel measurement I is written as:

$$I = \int_{\Omega} f(\bar{x}) d\mu(\bar{x}), \quad (4)$$

where $\bar{x} = x_0 \dots x_k$ is a light path, Ω is the path space, $d\mu(\bar{x}) = dA(x_0) \dots dA(x_k)$ is the area product measure, and f is the measurement contribution function given by:

$$\begin{aligned} f(\bar{x}) &= L_e(x_0, x_1) T(\bar{x}) W_e(x_{k-1}, x_k), \\ T(\bar{x}) &= GV(x_0, x_1) \left[\prod_{i=1}^{k-1} \rho(x_{i-1}, x_i, x_{i+1}) GV(x_i, x_{i+1}) \right], \end{aligned}$$

where L_e is the emittance, W_e is the pixel sensitivity, ρ is the BSDF, and GV is the geometry term including the visibility. We summarize our notation in Table 1.

BPT generates an eye sub-path from the camera and a light sub-path from the light source for each pixel. Then the t -th vertex of the eye sub-path and the s -th vertex of the light sub-path are connected

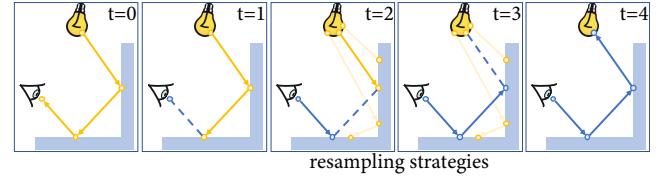


Figure 1: $k+2$ strategies can sample the same light path of length k ($k=3$ is illustrated). Resampling strategies ($t=2, 3$) resample light sub-paths from pre-sampled light sub-paths. Other strategies including unidirectional sampling ($t=0, 4$) and light tracing ($t=1$) are handled by the traditional BPT.

to generate a full light path $\bar{x} = \bar{y}\bar{z} = y_0 \dots y_{s-1} z_{t-1} \dots z_0$, where \bar{y} , \bar{z} are the light and the eye sub-paths, respectively. BPT can generate the same full light path of length $k (= s+t-1)$ in multiple ways (referred to as *strategies*). For a full light path of length k , $k+2$ strategies can sample the same full light path, where each strategy is identified by an integer t of the connected eye sub-path vertex z_{t-1} .

BPT weighs each strategy t using the weighting function w_t in multiple importance sampling (MIS) [Vea97] so that the sum of the weighting functions for a particular light path \bar{x} is equal to one (i.e., $\sum_{t=0}^{k+1} w_t(\bar{x}) = 1$). Under the condition that the sum of the weighting functions is equal to one and the weighting function returns zero whenever the pdf returns zero, the weighting functions can be (freely) designed to reduce the variance. The balance heuristic [Vea97] is designed to reduce the upper bound of the variance:

$$w_t(\bar{x}) = \frac{n_t p_t(\bar{x})}{\sum_{i=0}^{k+1} n_i p_i(\bar{x})}, \quad (5)$$

where n_t is the number of samples for the t -th strategy, and the pdf $p_t(\bar{x})$ for the t -th strategy is the product of sampling pdfs for the sub-paths $p(\bar{y})p(\bar{z})$.

3.3. Resampled Importance Sampling BPT (RISBPT)

Instead of generating a *single* light sub-path for each eye sub-path, several BPT-based methods generate *multiple* light sub-paths shared by all the eye sub-paths [PRDD15, NID20]. Then for each eye sub-path, a small number of light sub-paths are resampled from the pre-sampled multiple light sub-paths.

Nabata et al. interpret the resampling of light sub-paths from the pre-sampled light sub-paths as RIS in the context of BPT [NID20]. We refer to this as RISBPT. RISBPT uses the *resampling* strategies that connect an eye sub-path path vertex to a vertex of the resampled light sub-path taking into account the contribution of the connection. The other sampling strategies (i.e., unidirectional sampling and light tracing) are handled by BPT, since the unidirectional sampling strategy does not require the connections, and light tracing is efficiently handled by BPT. For a light path \bar{x} with length k , the unidirectional sampling strategies correspond to $t=0, k+1$, and the light tracing corresponds to $t=1$, and the resampling strategies correspond to $t=2, \dots, k$ as shown in Fig. 1.

We now focus on the resampling strategies. The pixel measurement I_t for the resampling strategy is calculated by integrating the

product of the measurement contribution function f and the weighting function w_t over the light sub-paths of arbitrary length ($s \geq 1$). The pixel measurement I_t is calculated by sampling an eye sub-path sample \bar{Z}_t with t vertices as follows:

$$\frac{1}{p(\bar{Z}_t)} \sum_{s \geq 1} \int_{A^s} w_t(\bar{y}_s \bar{Z}_t) f(\bar{y}_s \bar{Z}_t) d\mu(\bar{y}_s),$$

where $p(\bar{Z}_t)$ is the sampling pdf of the eye sub-path sample \bar{Z}_t , A^s is the s -dimensional Cartesian product of the scene surface A , \bar{y}_s is the integral variable for a light sub-path with s vertices. Analogous to the RIS estimator $\langle F \rangle_{\text{ris}}$ in Eq. (1), the RIS estimator $\langle I_t \rangle_{\text{ris}}$ is calculated from the following equations:

$$\langle I_t \rangle_{\text{ris}} = \frac{1}{M} \frac{w_t(\bar{Y} \bar{Z}_t) f(\bar{Y} \bar{Z}_t)}{p(\bar{Y}) p(\bar{Z}_t) P_r(\bar{Y} | \bar{Y})}, \quad (6)$$

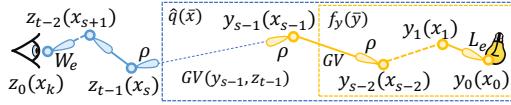
$$P_r(\bar{Y} | \bar{Y}) = \frac{\hat{q}(\bar{Y} \bar{Z}_t) / p(\bar{Y})}{\sum_{s \geq 1} \sum_{i=1}^M \hat{q}(\bar{Y}_{s,i} \bar{Z}_t) / p(\bar{Y}_{s,i})}, \quad (7)$$

where \bar{Y} is the light sub-path sample resampled from the set \bar{Y} , and P_r is referred to as the *resampling pmf*. In the context of BPT, a proposal for RIS is a light sub-path and the set of proposals \bar{Y} is $\{\bar{Y}_{1,1}, \dots, \bar{Y}_{s,i}, \dots\}$ where $\bar{Y}_{s,i}$ is the i -th light sub-path sample with s vertices. The target distribution $\hat{q}(\bar{x})$ in RISBPT is calculated as:

$$\hat{q}(\bar{x}) = \hat{q}(\bar{y}, z_{t-1}) = f_y(\bar{y}) p(y_{s-2}, y_{s-1}, z_{t-1}) G V(y_{s-1}, z_{t-1}), \quad (8)$$

where $f_y(\bar{y})$ is a part of the measurement contribution function f that depends only on the light sub-path \bar{y} as:

$$f_y(\bar{y}) = L_e(y_0, y_1) G V(y_0, y_1) \prod_{i=1}^{s-2} p(y_{i-1}, y_i, y_{i+1}) G V(y_i, y_{i+1}).$$



The key to RISBPT is the resampling-aware weighting function that can handle the changes in probability density due to resampling as described in Sec. 3.1. The resampling-aware weighting function uses the following density p_{ris} instead of p_t for the resampling strategy t as:

$$p_{\text{ris}}(\bar{x}) = \left(\frac{1}{M} \frac{1}{p_t(\bar{x})} + \left(1 - \frac{1}{M}\right) \frac{1}{q(\bar{x})} \right)^{-1}. \quad (9)$$

Different from RIS, RISBPT requires the normalization factor Q of the target distribution \hat{q} to calculate the target pdf q for the weighting functions:

$$q(\bar{x}) = \frac{\hat{q}(\bar{y}, z_{t-1})}{Q} p(\bar{z}) = \frac{\hat{q}(\bar{y}, z_{t-1})}{\sum_{s \geq 1} \int_{A^s} \hat{q}(\bar{y}_s, z_{t-1}) d\mu(\bar{y}_s)} p(\bar{z}) \\ = q(\bar{y} | z_{t-1}) p(\bar{z}), \quad (10)$$

where the normalization factor Q is estimated using Monte-Carlo integration with pre-sampled light sub-paths.

Since the computation of the resampling pmf P_r and the normalization factor Q for each eye sub-path vertex is costly, RISBPT computes the resampling pmf and the normalization factor Q at sparsely distributed cache points (e.g., a small subset of eye sub-path vertices). The resampling pmf P_r and the normalization factor

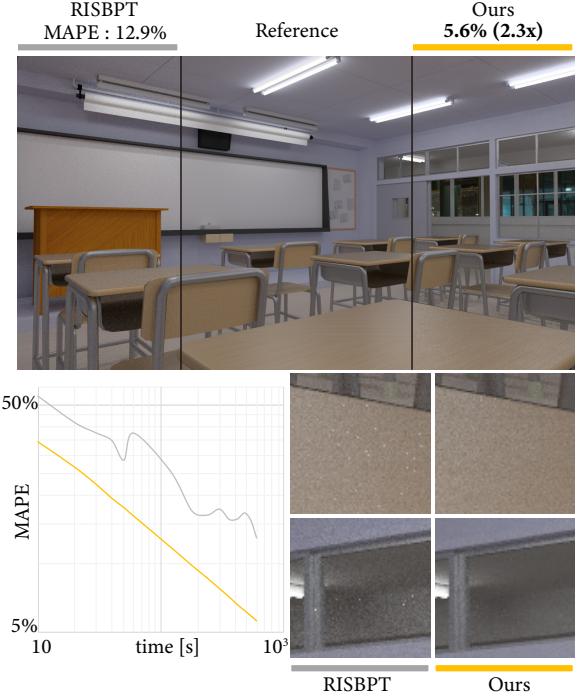


Figure 2: RISBPT is susceptible to light sub-paths with extremely high contribution. The entire image (especially the whiteboard and the wall) rendered by RIBPT (top left) is brighter than the reference (top center). The convergence plot of RISBPT (gray) shows unstable noise reduction for a small number ($M = 50$) of light sub-paths, while that of our two-stage resampling method (light orange) shows stable noise reduction where the same number ($M_2 = 50$) of light sub-paths are used in the second stage. RISBPT also suffers from high intensity noise (right bottom images) due to the path correlation. Our method can alleviate this by using a large number of pre-sampled light sub-paths in the first stage and yield $2.3 \times$ smaller mean absolute percentage errors (MAPEs).

Q for each eye sub-path vertex are replaced by those stored at the nearest cache point. The target pdf q is also approximated by using the nearest cache point c_t of the t -th eye sub-path vertex z_{t-1} as $q(\bar{x}) \approx q(\bar{y} | c_t) p(\bar{z})$.

3.3.1. Discussion

RISBPT inherits the properties of RIS. That is, the distribution of the sampled path approaches the target pdf q when the number of traced light sub-paths M increases. Since the target pdf $q(\bar{x} | f_y p G V)$ is closer to the measurement contribution function f than the sampling pdf $p(\bar{x} | f_y)$, the use of larger M leads to reduced variance in general. However, the use of larger M also incurs increased computational time for the resampling pmf P_r as it is linear in M as shown in Eq. (7). In the context of BPT, the computation of the target distribution \hat{q} further compounds this problem since the target distribution \hat{q} involves the costly visibility tests $G V$ in Eq. (8). Therefore, the increase in M can decrease the possible iteration count, resulting in increased noise in equal-time rendering.

Smaller M can do more iterations in equal-time rendering, but this leads to the correlation of light sub-paths since the pre-sampled light sub-paths are shared by all the eye sub-paths, and RISBPT uses the *same* sets of pre-sampled light sub-paths for all the cache points. In addition, a smaller number of pre-sampled light sub-paths induces unstable noise reduction as shown in the convergence plots in Fig. 2. The cause of this unstable noise reduction is mainly two-fold. Firstly, the use of a relatively small number of light sub-paths can suffer from the light sub-paths with extremely high contributions. This affects the entire image since the pre-sampled light sub-paths are shared by all the eye sub-paths, which can be found in the whiteboard of Fig. 2 (top left). Secondly, the small number of light sub-paths can undermine the estimation accuracy of the normalization factor Q , and thus the weighting function w_t . Therefore, handling a large number of pre-sampled light sub-paths in an efficient manner is crucial for efficient and stable rendering.

4. Two-stage Resampling

In this section, we propose the theory of two-stage resampling capable of handling a large number of proposals. Its application to BPT is described in the next section. Two-stage resampling first samples M_1 proposals $\mathbf{X} = \{X_1, \dots, X_{M_1}\}$ using a sampling pdf p which is readily sampled. Then M_2 proposals $\mathbf{X} = \{X_{i_1}, \dots, X_{i_{M_2}}\}$ are resampled from the set of proposals \mathbf{X} based on the target distribution \hat{q}_1 , where \mathbf{X} is a subset of proposals \mathbf{X} (i.e., the index of the j -th proposal $i_j \in \{1, \dots, M_1\}$). We refer to this resampling as the *first resampling stage*. Two-stage resampling again draws samples from the subset of proposals \mathbf{X} based on the target distribution \hat{q}_2 , which is referred to as the *second resampling stage*. To estimate the integral of f , the one-sample estimator of the two-stage resampling is calculated as:

$$\begin{aligned} \langle F \rangle_{\text{tsr}} &= \frac{f(X)}{\hat{q}_2(X)} \left(\frac{1}{M_1} \sum_{j=1}^{M_1} \frac{\hat{q}_1(X_j)}{p(X_j)} \right) \left(\frac{1}{M_2} \sum_{j=1}^{M_2} \frac{\hat{q}_2(X_{i_j})}{\hat{q}_1(X_{i_j})} \right) \\ &= \frac{1}{M_1 M_2} \frac{f(X)}{p(X)} \left(\frac{\sum_{j=1}^{M_1} \frac{\hat{q}_1(X_j)}{p(X_j)}}{\frac{\hat{q}_1(X)}{p(X)}} \right) \left(\frac{\sum_{j=1}^{M_2} \frac{\hat{q}_2(X_{i_j})}{\hat{q}_1(X_{i_j})}}{\frac{\hat{q}_2(X)}{\hat{q}_1(X)}} \right) \\ &= \frac{1}{M_1 M_2} \frac{f(X)}{p(X) P_r(X|\mathbf{X}, \mathbf{X})}, \end{aligned} \quad (11)$$

where the pmf P_r in two-stage resampling is the product of the two pmfs P_1 and P_2 given by:

$$\begin{aligned} P_1(X|\mathbf{X}) &= \frac{\hat{q}_1(X)/p(X)}{\sum_{j=1}^{M_1} \hat{q}_1(X_j)/p(X_j)}, \\ P_2(X|\mathbf{X}) &= \frac{\hat{q}_2(X)/\hat{q}_1(X)}{\sum_{j=1}^{M_2} \hat{q}_2(X_{i_j})/\hat{q}_1(X_{i_j})}. \end{aligned}$$

Owing to the similarity between ordinary RIS and two-stage resampling, the variance for the two-stage resampling estimator $V[\langle F \rangle_{\text{tsr}}]$ is represented by: (the derivation is shown in the supplemental material)

$$\frac{1}{M_1} V\left[\frac{f}{p}\right] + \left(1 - \frac{1}{M_1}\right) \left(\frac{1}{M_2} V\left[\frac{f}{q_1}\right] + \left(1 - \frac{1}{M_2}\right) V\left[\frac{f}{q_2}\right] \right), \quad (12)$$

where q_1 and q_2 are the target pdfs of \hat{q}_1 and \hat{q}_2 , respectively.

Intuitively, two-stage resampling is identical to ordinary RIS when $M_1 = M, M_2 = 1$, and $q = q_1$, and the variance in Eq. (12) matches the variance in Eq. (3).

We now analyze the computational cost of two-stage resampling and ordinary RIS. To simplify the explanation and the analysis, we equalize the computational cost of ordinary RIS and that of the second resampling stage. That is, we assume that the number of proposals M_2 is equal to M and the target distribution \hat{q}_2 is equal to \hat{q} . We also assume that the pmfs P_1 for two-stage resampling are constructed K_1 times, and the pmf P_r (and P_2) is constructed K times. Then the computational times for the ordinary RIS and the two-stage resampling are KMT and $K_1 M_1 T_1 + KMT$, where T_1 and T are the computational times to evaluate the target distributions \hat{q}_1 and \hat{q} , respectively. Since two-stage resampling performs another resampling process, the two-stage resampling intrinsically incurs the computational overhead $K_1 M_1 T_1$. However, we have a degree of freedom to design the target distribution \hat{q}_1 and choose the parameter K_1 to reduce the overhead even for a large value of M_1 . If M_1 is large enough to draw subsets of $M_2 (= M)$ proposals multiple times (K) from a single set of proposals \mathbf{X} ($K_1 = 1$), and we can design the target distribution \hat{q}_1 such that its computational cost T_1 is small compared to T for \hat{q}_2 , the computational overhead to construct the pmf P_1 for a large number of proposals M_1 can be amortized.

The advantages of two-stage resampling are two-fold. Firstly, the variance of two-stage resampling $V[\langle F \rangle_{\text{tsr}}]$ in Eq. (12) can be smaller than that of ordinary RIS $V[\langle F \rangle_{\text{ris}}]$ in Eq. (3). When M_1 is large enough to neglect the variance term $V[f/p]$ in $V[\langle F \rangle_{\text{tsr}}]$ and the target distribution \hat{q}_1 is designed to be closer to the integrand f than the sampling pdf p , the difference between $V[\langle F \rangle_{\text{tsr}}]$ and $V[\langle F \rangle_{\text{ris}}]$ is that between the variance terms $V[f/q_1]$ and $V[f/p]$. Since the target pdf q_1 is closer to the integrand f than the sampling pdf p , the variance term $V[f/q_1]$ is smaller than $V[f/p]$ in general, and thus the variance $V[\langle F \rangle_{\text{tsr}}]$ is smaller than $V[\langle F \rangle_{\text{ris}}]$. Secondly, two-stage resampling can generate multiple *different* subsets of M_2 proposals from a single set of proposals \mathbf{X} . This is beneficial to applications that require M_2 proposals multiple times. Two-stage resampling can mitigate the correlation due to reusing the same set of proposals multiple (K) times in ordinary RIS by using potentially different proposals.

5. Bidirectional Path Tracing with Two-stage Resampling

We apply the two-stage resampling approach to RISBPT to handle a large number of pre-sampled light sub-paths in an efficient manner. An overview of the method is shown in Fig. 3. In the first stage, we resample a smaller number (e.g. 10^2) of light sub-paths from a large number (e.g. 10^4) of pre-sampled light sub-paths with a simplified and inexpensive target distribution \hat{q}_1 . Then, in the second stage, we again resample a light sub-path from the smaller set to be connected to a given eye sub-path. We devise a weighting function that can properly treat changes in probability density due to two-stage resampling.

We show our two-stage resampling approach in Algorithm 1.

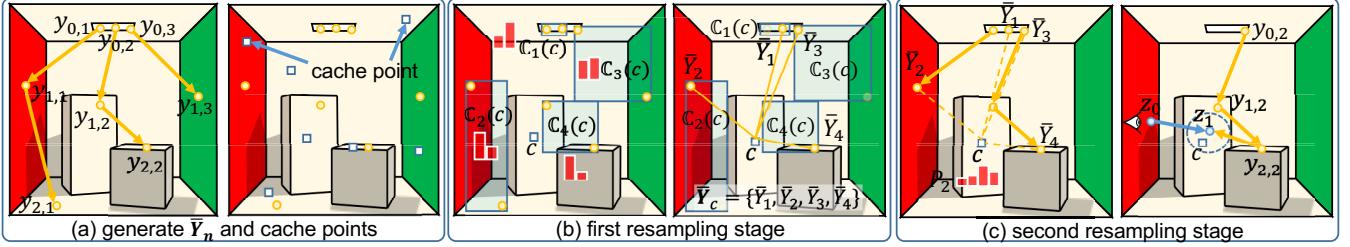


Figure 3: Overview of our method. (a) For each iteration, multiple light sub-paths are generated as proposals in RIS. In this case, $M_1 = 3$ light sub-paths are traced and the set \bar{Y}_n consists of eight proposals $\bar{Y}_n = \{y_{0,1}, y_{0,1}y_{1,1}, y_{0,1}y_{1,1}y_{2,1}, y_{0,2}, y_{0,2}y_{1,2}, y_{0,2}y_{1,2}y_{2,2}, y_{0,3}, y_{0,3}y_{1,3}\}$. Cache points are also generated on the surface of the scene at each iteration. (b) In the first resampling stage, \bar{Y}_n is partitioned into M_2 clusters at each cache point c . In this case, \bar{Y}_n is classified into $M_2 = 4$ clusters $\mathbb{C}_1, \mathbb{C}_2, \mathbb{C}_3, \mathbb{C}_4$. The subset \bar{Y}_c comprised of M_2 light sub-paths is generated for each cache point c by resampling one light sub-path from each cluster \mathbb{C}_i . (c) In the second resampling stage, P_2 is calculated for each cache point. Then a light sub-path is resampled from the subset of pre-sampled light sub-paths stored at the closest cache point c , and a full light path is sampled.

Algorithm 1 Our two-stage resampling BPT algorithm. The underlined sentences show the steps different from RISBPT.

```

1: for  $n \leftarrow 1$  to maxIterations do
2:   generate  $\bar{Y}_n$  by tracing  $M_1$  paths from light sources
3:   generate cache points
4:   calculate  $P_1 \triangleright$  Eq. (14)
5:   //first resampling stage
6:   for each cache point  $c$  do
7:     classify  $\bar{Y}_n$  into  $M_2$  clusters
8:     generate  $\bar{Y}_c$  by resampling from each cluster
9:   // second resampling stage
10:  for each pixel do
11:    generate one light sub-path and one eye sub-path  $\bar{z}$ 
12:    calculate estimators for other strategies using BPT
13:    for each vertex of  $\bar{z}$  with  $(t \geq 2)$  do
14:      find the nearest cache point  $c_t$ 
15:      resample light sub-path  $\bar{y}$  from  $\bar{Y}_{c_t}$  using  $P_2$ 
16:      generate a full path  $\bar{x} = y_0 \dots y_{s-1} z_{t-1} \dots z_0$ 
17:      calculate weighting function  $w_t$  using  $p_{tsr} \triangleright$  Eq. (17)
18:      calculate  $\langle I_t \rangle_{tsr}$  using  $w_t$  and cached  $P_r \triangleright$  Eq. (15)
19:    update pixel intensity using  $\langle I_t \rangle_{tsr}$  and other estimators

```

Our method renders an image by iteratively updating the pixel intensities. At each iteration n , we first generate a pre-sampled set \bar{Y}_n of light sub-paths, which is constructed by tracing M_1 paths from the light sources (Line 2). Next, a set of cache points is generated on the surface of the scene (Line 3). The first resampling stage applies to the cache points. For each cache point c , all the light sub-paths in \bar{Y}_n are classified into M_2 clusters (Line 7). The pmf P_1 is computed for each element in each cluster. Using P_1 , a light sub-path is resampled from each cluster to create a smaller subset \bar{Y}_c whose size is $M_2 (\ll M_1)$ (Line 8). Next, the pmf P_2 used for the second stage is calculated for each of the light sub-paths in the subset \bar{Y}_c .

5.1. First Resampling Stage

The purpose of the first resampling stage is to create a subset \bar{Y}_c comprising M_2 light sub-paths for each cache point c by resampling M_2 samples from \bar{Y}_n . We use RIS for this resampling process. Since RIS requires the construction of a pmf whose cost is proportional to the number of proposals, the first resampling stage is expensive when there are many light sub-paths in \bar{Y}_n . To address this problem, we cluster the light sub-paths into a binary light tree used in Lightcuts [WFA⁰⁵], by treating the last vertex of each light sub-path as a point source. Furthermore, we use the following simplified function for the target distribution \hat{q}_1 :

$$\hat{q}_1(\bar{y}, c) = f_y(\bar{y})G(y_{s-1}, c), \quad (13)$$

where G is the geometry term. The BSDF and the visibility terms are excluded for efficient evaluation and conservative sampling, since the discrepancy between the visibility terms for the cache point c and the eye sub-path vertex can miss light sub-paths with large contributions. The light sub-paths in \bar{Y}_n are partitioned into M_2 clusters $\mathbb{C}_1(c), \dots, \mathbb{C}_{M_2}(c)$ so that the geometry term G for the light sub-paths in each cluster becomes nearly constant. Then from each cluster $\mathbb{C}_j(c)$, a single light sub-path sample \bar{Y}_j is resampled using the following pmf P_1 :

$$P_1(\bar{Y}_j, c) = \frac{f_y(\bar{Y}_j)/p(\bar{Y}_j)}{\sum_{\bar{Y} \in \mathbb{C}_j(c)} f_y(\bar{Y})/p(\bar{Y})}. \quad (14)$$

Our method uses f_y instead of \hat{q}_1 for efficient resampling, since the values of G in \hat{q}_1 are considered to be constant in each cluster. Moreover, the use of f_y enables us to compute the sum of f_y/p only once for all the clusters in the binary light tree since f_y is independent of the positions of the cache points, making the construction of the pmfs P_1 more efficient. The M_2 samples resampled from the clusters are the set of pre-sampled light sub-paths for cache point c , $\bar{Y}_c = \{\bar{Y}_1, \dots, \bar{Y}_{M_2}\}$, which is used in the subsequent resampling stage.

5.2. Second Resampling Stage

In this second stage, we use the target distribution $\hat{q}_2 = \hat{q}$ in Eq. (8), which is more expensive to evaluate, but the number of propos-

als, M_2 , is much smaller than M_1 . For each vertex of a given eye sub-path \bar{Z}_t , the two-stage resampling estimator $\langle I_t \rangle_{\text{tsr}}$ is calculated using the following equation:

$$\langle I_t \rangle_{\text{tsr}} = \frac{1}{M_1 M_2} \frac{w_t(\bar{Y} \bar{Z}_t) f(\bar{Y} \bar{Z}_t)}{p(\bar{Y}) p(\bar{Z}_t) P_r(\bar{Y} | \bar{Y}, \bar{Y})}, \quad (15)$$

where \bar{Y} is the resampled light sub-path from the subset \bar{Y} . The resampling pmf $P_r(\bar{Y} | \bar{Y}, \bar{Y})$ is approximated with that stored at the closest cache point c_t . The resampling pmf P_r at the closest cache point c_t is the product of $P_1(\bar{Y}, c_t)$ and $P_2(\bar{Y}, c_t)$, given by:

$$P_2(\bar{Y}, c_t) = \frac{\hat{q}_2(\bar{Y}, c_t) / \hat{q}_1(\bar{Y}, c_t)}{\sum_{i=1}^{M_2} \hat{q}_2(\bar{Y}_i, c_t) / \hat{q}_1(\bar{Y}_i, c_t)}. \quad (16)$$

5.3. Weighting Function for Two-stage Resampling

Next, we explain the weighting functions that aim to reduce the variance for two-stage resampling. Unfortunately, when the clustering approach (i.e., stratified sampling) is used in the first resampling stage, the variance is no longer expressed in the simple form of Eq. (12) that can be applied to the same derivation technique as the balance heuristic. Therefore, we assume that importance sampling is used in the first resampling stage during the derivation of the variance. This is a reasonable assumption since the variance of stratified sampling is smaller than that of importance sampling in general. By using the variance in Eq. (12) and the similarity between the variance of RIS in Eq. (3) and the density in Eq. (9), the density for two-stage resampling $p_{\text{tsr}}(\bar{x})$ in the weighting function w_t is given by (the derivation is shown in the supplemental material):

$$p_{\text{tsr}}(\bar{x}) = \left(\frac{1}{M_1} \frac{1}{p_t(\bar{x})} + \left(1 - \frac{1}{M_1} \right) \left(\frac{1}{M_2} \frac{1}{q_1(\bar{x})} + \left(1 - \frac{1}{M_2} \right) \frac{1}{q_2(\bar{x})} \right) \right)^{-1}, \quad (17)$$

where $q_i(\bar{x})$ is defined as:

$$q_i(\bar{x}) = \frac{\hat{q}_i(\bar{y}, z_{t-1})}{Q_i} p(\bar{z}) = q_i(\bar{y} | z_{t-1}) p(\bar{z}),$$

where Q_i is the normalization factor for the i -th distribution function \hat{q}_i . The target pdfs q_1 and q_2 are approximated by $q_1(\bar{y} | c_t) p(\bar{z})$, and $q_2(\bar{y} | c_t) p(\bar{z})$ of the closest cache point c_t for the t -th eye sub-path vertex z_{t-1} . The weighting function w_t for two-stage resampling uses the probability density p_{tsr} instead of p_t for the resampling strategies. The probability density p_{tsr} approaches the target pdf q_2 for sufficiently large M_1 and M_2 .

5.4. Implementation Details

Our two-stage resampling BPT method builds upon RISBPT, and thus inherits from the implementation of RISBPT. Specifically, our method estimates the normalization factors Q_i (used for the weighting function w_t) and the measurement contribution function f , independently, since $E[w_t f] = E[w_t] E[f]$ holds only when w_t and f are estimated independently. To do so, our method uses the set of proposals \bar{Y}_n to estimate the measurement contribution function f , and the normalization factors Q_i are estimated by using the set of proposals \bar{Y}_{n-1} of the previous iteration. Since the positions of the

cache points change at each iteration, the normalization factor Q_i at cache point c of the current iteration is calculated from the average of Q_i estimated in the previous iteration at nearby cache points of c .

Our method can use multiple cache points, so as not to miss important light sub-paths due to the difference between the target distribution function \hat{q}_2 stored at the nearest cache point and the eye sub-path vertex. Our method uses N_c closest cache points and a virtual cache point that stores a special resampling pmf with uniform distribution. We select one cache point from these $(N_c + 1)$ cache points uniformly, then the resampling process is performed using the selected cache point. For multiple cache points, the density p_{tsr} used in the weighting function w_t is replaced by the average of the densities calculated at $(N_c + 1)$ cache points and is given by:

$$p_{\text{tsr}}(\bar{x}) = \sum_{k=1}^{N_c+1} p_{\text{tsr},k}(\bar{x}) / (N_c + 1),$$

where $p_{\text{tsr},k}$ is calculated by Eq. (17) using the target pdfs q_1 and q_2 stored at the k -th nearest cache point. The two-stage resampling estimator $\langle I_t \rangle_{\text{tsr}}$ for multiple cache points is then modified as follows:

$$\langle I_t \rangle_{\text{tsr}} = \frac{1}{M_1 M_2} \frac{w_t(\bar{Y} \bar{Z}_t) f(\bar{Y} \bar{Z}_t)}{p(\bar{Y}) p(\bar{Z}_t) P_r(\bar{Y} | \bar{Y}, \bar{Y})} \frac{p_{\text{tsr},i}(\bar{Y} \bar{Z}_t)}{p_{\text{tsr}}(\bar{Y} \bar{Z}_t)},$$

where i is the index of the selected cache point.

5.5. Discussion

We validate our two-stage resampling approach by comparing it with RISBPT in equal-iteration rendering and equal-time rendering. Fig. 4 shows the rendering results by RISBPT with $M = 10^4$ and our two-stage resampling with $M_1 = 10^4$ and $M_2 = 512$. Figs. 4(a) and (b) show a comparison for an equal number of iterations (16 iterations) comparison. The image in Fig. 4(a) rendered by our two-stage resampling approach is almost the same quality as that in Fig. 4(b) rendered by RISBPT, albeit the original target distribution \hat{q}_2 is not used for all proposals. In equal-time rendering (2 min), more iterations can be performed in two-stage resampling (96 iterations) than with RISBPT (16 iterations), resulting in reduced noise as shown in Fig. 4(b) and (c).

6. Results

We evaluate our two-stage resampling method by comparing with BPT, PCBPT [PRDD15], and RISBPT [NID20]. All the images are rendered at a resolution of 1280×720 on a PC with an Intel Core i9-7890XE CPU with 18 cores. Fig. 5 shows comparisons between BPT, PCBPT, RISBPT, and our method. We measure mean absolute percentage error (MAPE). The numbers of pre-sampled light subpaths, M_1 and M_2 , are set to 10^4 and 200, respectively. These values are determined experimentally, discussed in Sec. 6.1. To investigate the effectiveness of our two-stage resampling method, we use the same number of light sub-paths ($M = M_2$) and the same target distribution $\hat{q} = \hat{q}_2$ (Eq. (8)) in PCBPT and RISBPT. We use the vertices of $0.4\% \times W \times H$ eye sub-paths as the cache points where W and H are the width and the height of the image, and the number of nearest cache points N_c is set to three in PCBPT, RISBPT, and

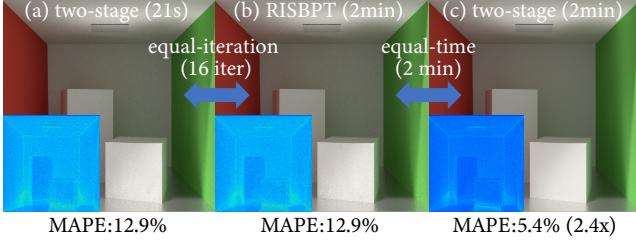


Figure 4: Comparison between RISBPT [NID20] and two-stage resampling in equal number of iterations (16 iterations) and equal rendering time (2 min). After 16 iterations, (a) two-stage resampling yields similar results as (b) RISBPT, though \hat{q}_2 is not used for all proposals in two-stage resampling. The computation time for (a) two-stage resampling is 21 seconds and that for (b) one-stage resampling is 2 min. In equal rendering time, (c) two-stage resampling can iterate 96 times, while the iteration count for (b) RISBPT is 16. The two-stage resampling yields much better results (2.4× smaller errors) as shown in the insets that visualize the absolute percentage errors.

our method. The number of resampled light sub-paths is set to one in PCBPT, RISBPT, and our method. As shown in the zoomed-in images and the MAPEs, our method outperforms BPT, PCBPT, and RISBPT in these scenes.

The Sponza scene is illuminated by an environment map, where the illumination enters from a small region at the top of the building. The inner wall beyond the arches is illuminated by indirect illumination, and it is difficult to sample such light paths due to the occlusions. RISBPT yields better results than BPT and PCBPT since it considers the contribution of the connections and provides proper weights. However, the right arch (bottom image) in RISBPT is slightly brighter than the reference and noise can be seen on the wall beyond the arch. Since our method uses a large number of light sub-paths and different subsets of pre-sampled light sub-paths at cache points, it can sample light paths with non-zero contributions for such difficult cases.

The Door scene is illuminated by an area light source located in the room next to the door. As shown in the zoomed-in image of RISBPT, the floor under the table is brighter than the reference due to the effects of light sub-paths with high contribution. Our method can avoid such artifacts due to the use of $M_1 = 10^4$ pre-sampled light sub-paths and can reduce noise as shown in the zoomed-in images.

The Classroom scene is illuminated by area light sources located on the ceiling and an environment map. In this scene, RISBPT can considerably reduce noise compared with BPT and PCBPT. However, the artifacts can still be seen in RISBPT (i.e., the wall of the hallway in RISBPT is brighter than the reference) even though the number of pre-sampled light sub-paths increases from $M = 50$ (in Fig. 2) to $M = 200$ (in Fig. 5). Our method can mitigate such artifacts and can reduce noise further (2.61× smaller MAPE than BPT).

The Bedroom scene is illuminated by an environment map, where the illumination enters from a small gap in the curtain.

Although PCBPT considers the visibilities to resample light sub-paths, PCBPT has larger variance than the traditional BPT, since the weighting functions used in PCBPT are unaware of the changes in path probability and it does not assign larger weights to resampling strategies. RISBPT can assign larger weights to resampling strategies and can reduce variance. However, RISBPT can be easily affected by light sub-paths with high contribution since it assigns large weights to light paths sampled by resampling strategies. The zoomed-in images (the bed comforter and the wall) in RISBPT are brighter than the reference. On the other hand, since our method generates a large number of light sub-paths in the first resampling stage, the effects of light sub-paths with high contribution are mitigated and our method can render less noise images.

The House scene is illuminated by area light sources inside the house and an environment map. The House scene is rendered by each BPT method combined with PM via VCM/UPS [GKDS12, HPJ12]. In this scene, the bottom zoomed-in image of RISBPT is slightly brighter than the reference, but that of our method is much more similar to the reference image.

Compared with RISBPT, our method yields 1.15, 1.38, 1.47, 1.33, and 1.10 times smaller MAPEs in these scenes (from top to bottom in Fig. 5), which correspond, approximately, to 1.33, 1.91, 2.15, 1.76, and 1.20 times faster convergence.

The convergence plots on log-log scales of the four methods are shown in the rightmost column of Fig. 5. As shown in the convergence plots of RISBPT (especially in the Classroom and Bedroom scenes), the noise reduction in RISBPT is unstable, though it eventually converges. This unstable noise reduction is problematic for progressive rendering. Although Fig. 5 uses $M = 200$ pre-sampled light sub-paths, the convergence plot of the Classroom scene in RISBPT still looks unstable. On the other hand, our two-stage resampling method achieves stable and better noise reduction as shown in the convergence plots.

6.1. Number of light sub-paths M_1 and M_2

We first measured the MAPEs and the number of iterations of the Sponza scene with 10 min rendering for sparsely sampled values of $M_1 = 10^3, 10^4, 10^5$ and $M_2 = 100, 500, 1000$ as shown in Fig. 6. In this experiment, $M_1 = 10^4$ and $M_2 = 100$ or 500 yield small values of the MAPEs, and $M_2 = 10^3$ leads to larger errors and a smaller number of iterations since the computational cost of the resampling pmf P_2 is high. Additionally, we measured the MAPEs for $M_1 = 100$ and 5×10^5 with fixed $M_2 = 100$. When $M_1 = 100$ is used, MAPE increases from 5.63% ($M_1 = 10^3$) to 7.53%, indicating that small M_1 does not make effective use of two-stage resampling. When $M_1 = 5 \times 10^5$ is used, MAPE increases from 5.74% ($M_1 = 10^5$) to 6.38% and the iteration count decreases from 413 to 353, since large M_1 incurs a high computational cost for constructing the pmf P_1 even though P_1 is constructed once per iteration and reused for all the cache points. We found that $M_1 = 10^4$ and relatively small values (around 100) of M_2 work fine.

As shown in the graphs on the right of Fig. 6, the iteration count in equal-time rendering (i.e., the computational time per each iteration) depends largely on the number of light sub-paths M_2 in the second resampling stage, not on M_1 in the first resampling stage.

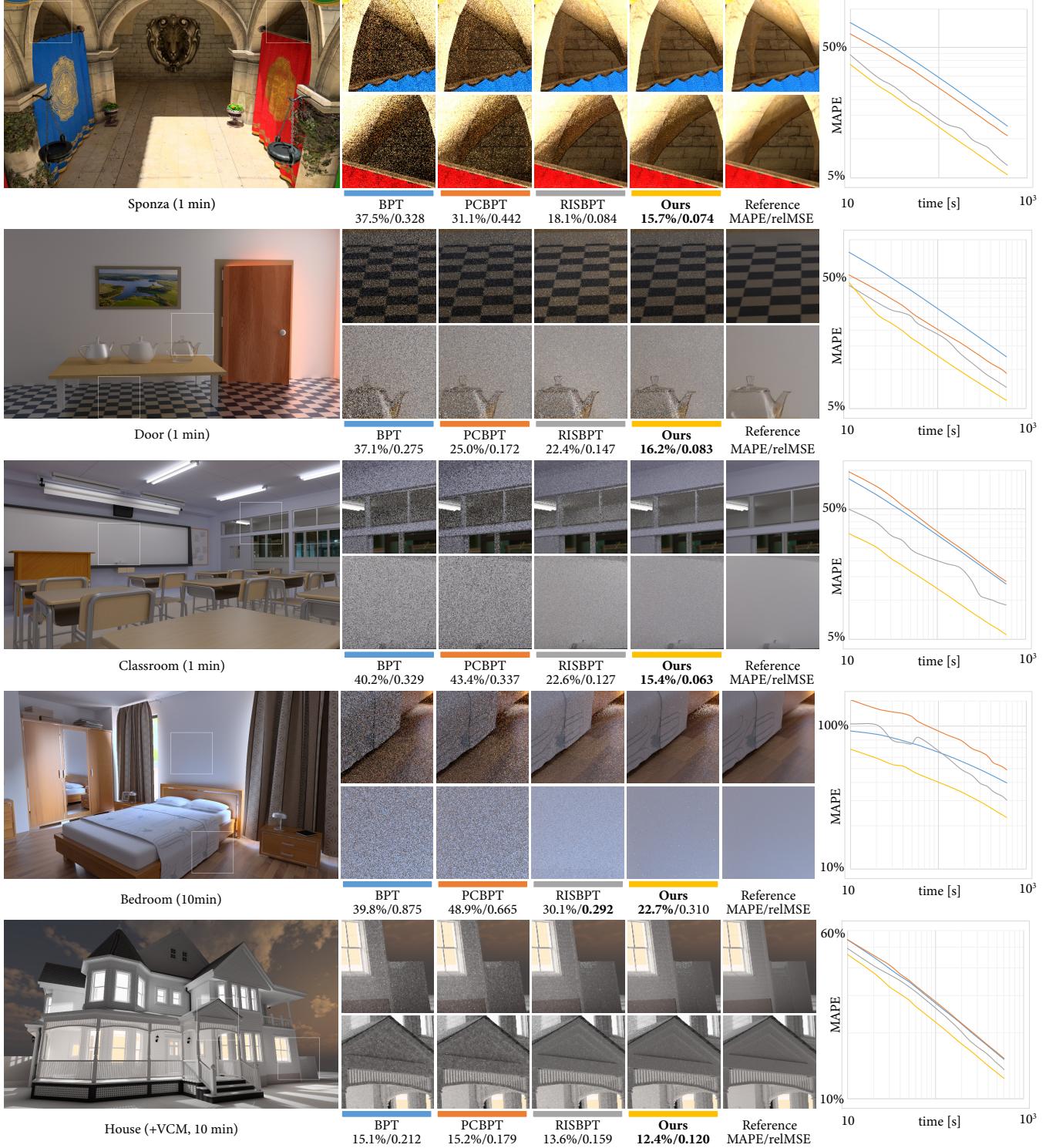


Figure 5: Equal-time comparisons between BPT, PCBPT, RISBPT, and our method. $M = 200$ is used for PCBPT and RISBPT, and $M_1 = 10^4, M_2 = 200$ is used for our method. Bold entries indicate the lowest MAPEs or the lowest relative mean square error (relMSE) among four BPT methods. Our method yields the lowest MAPE in all these scenes. The rightmost images show the convergence plots on log-log scale of four methods. As shown in the plots, RISBPT with $M = 200$ (gray colored plots) shows unstable noise reduction (Classroom and Bedroom) though it eventually converges, while our method (light orange colored plots) provides stable and better noise reduction due to the use of two orders of magnitude larger number of light sub-paths than PCBPT and RISBPT. Moreover, RISBPT can suffer from the light sub-path with extremely high contribution, resulting in bright images as shown in the floor of Door scene and the wall of Bedroom scene.

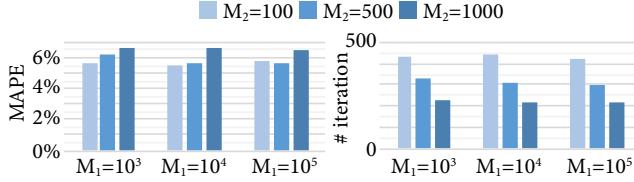


Figure 6: MAPEs (left) and the iteration count (right) of the Sponza scene (rendering in 10 min) for changing the parameters M_1 and M_2 . Since our method uses the inexpensive target distribution \hat{q}_1 and the pmf P_1 is constructed once per iteration, the computational time depends largely on M_2 .

This indicates that the additional computational overhead of the first resampling stage is negligible and can be amortized using our method.

Next, we measured the MAPEs and the number of iterations for various scenes by changing M_2 (M_1 is fixed to 10^4 and the rendering time is 10 min) as shown in Fig. 7. Fig. 7 shows a trade-off between the computational time and the number M_2 of pre-sampled light sub-paths in the second resampling stage. Using a large number for M_2 , the distribution of the sampled light path approaches the target pdf q_2 , which is much closer to the integrand than the sampling pdf p and q_1 , resulting in reduced noise. However, it also incurs high computational costs to construct the resampling pmf P_r , resulting in a smaller number of iterations as shown in the bottom of Fig. 7. On the other hand, using a small number for M_2 , we can do more iterations due to the small computational time required for resampling, but it cannot benefit from resampling and can suffer from correlation of the light sub-paths. Although the best performance (the lowest MAPE) depends on the scene, $M_2 = 200$ yields good results for various scenes (except for the Bedroom scene rendered by our method with path guiding, which we will discuss in Sec. 6.4).

We also compared our method with RISBPT for various numbers of $M (= M_2)$ in Fig. 8. Our method consistently yields smaller MAPEs for various numbers of M , except for the Sponza scene with $M (= M_2) = 800$. However, the difference is subtle (6.31% for RISBPT and 6.33% for our method) in this case.

6.2. Performance breakdown of RISBPT and our method

Fig. 9 shows the performance breakdown of RISBPT (resampling once) and our method (two-stage resampling) for various numbers $M (= M_2)$ of light sub-paths. The legends *pmf*, *render*, and *total* indicate the average computational times for constructing the resampling pmfs, and the rendering and total times (including the generation of light sub-paths and cache points), respectively. As expected, the computational time for the pmf scales linearly with M for both RISBPT and our method. This indicates that, for RISBPT, simply increasing the number M of pre-sampled light sub-paths to that used in our method ($M_1 + M_2 \approx 10^4$) is impractical. Although the computational time for the pmf using our method are about as twice as those for RISBPT due to the overhead of the first resampling stage (Lines 7&8 in Algorithm 1), the net overhead of two-

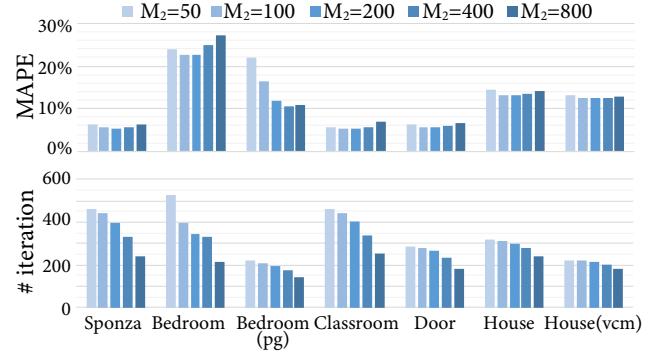


Figure 7: MAPEs (top) and the iteration count (bottom) for various scenes for changing M_2 (M_1 is fixed to 10^4 , rendering in 10 min). Bedroom(pg) and House(vcm) indicate that our method is combined with path guiding and VCM, respectively. $M_2 = 200$ works fine in various scenes except for the Bedroom scene rendered by our method with path guiding.

stage resampling ranges from 27% ($M = 200$) to 39% ($M = 800$) of the total rendering time.

6.3. Effects of Weighting Function

We evaluate the effects of our weighting function using p_{tsr} in Fig. 10. We compare our weighting function with the balance heuristic (i.e., using p instead of p_{tsr}) used in PCBPT [PRDD15] and HRR [TH19] by incorporating it into our two-stage resampling BPT. As shown in the zoomed-in images and the MAPEs, the balance heuristic cannot reap the benefits of two-stage resampling since it is unaware of the change in the probability density. When the balance heuristic is used in two-stage resampling BPT, the number of iterations decreases from 688 for PCBPT (i.e., resampling once with the balance heuristic) to 304 due to the overhead, resulting in increased noise from 10.2% in PCBPT to 18.0% as shown in Fig. 10 right, while our method yields 2.86× smaller MAPEs as shown on the left of Fig. 10.

We also incorporate the resampling-aware weighting function using p_{ris} [NID20] into our two-stage resampling BPT by substituting $M = M_2$ and $q = q_2$ in Eq. (9). The value of MAPE in this case is 14.5%, and our method yields 2.30× smaller MAPE (6.3%). This indicates that our method can properly treat the changes in probability density due to two-stage resampling, even though our method approximates the variance by assuming that importance sampling is used in the first resampling stage.

6.4. Discussion

We render the Bedroom scene using BPT, PCBPT, RISBPT, and our method, all of which are combined with path guiding [VKv*14]. Fig. 11 shows the convergence plots for $M (= M_2) = 200$ and the MAPEs of RISBPT and our method with varying numbers of $M (= M_2)$ and a rendering time of 10 min. For a relatively small number of light sub-paths (e.g., $M = 50$), RISBPT outperforms our method since path guiding can generate important light

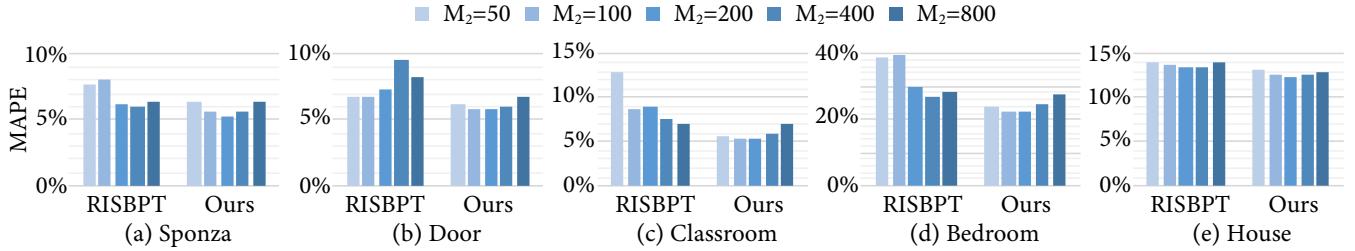


Figure 8: Comparison between RISBPT and our method for various numbers of $M (= M_2)$. Our method yields smaller MAPEs than RISBPT for various M_2 .

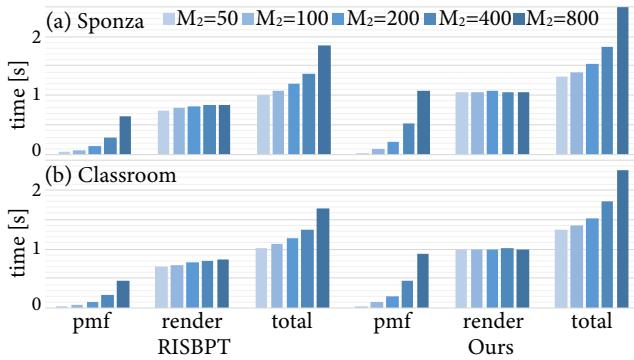


Figure 9: The average computational times per iteration of constructing pmf (pmf), rendering (render), and the total (total) in RISBPT (left) and ours (right) of (a) Sponza and (b) Classroom scenes. As expected, the computational times for constructing resampling pmfs increase linearly in the number of light subpaths M .

sub-paths even for scenes with difficult visibilities. In the case of $M = 200$, the MAPEs of RISBPT and our method are 11.82% and 11.79%, respectively. When combined with path guiding, the computational times for rendering are dominant (74% for $M = 50$), and the ratios for constructing the pmfs become relatively small (26% for $M = 800$). Therefore, the overhead due to the increase in M is small compared with other scenes, and thus our method combined with path guiding becomes more effective for larger M (MAPEs are 10.5% and 10.9% for $M = 400$ and 800, respectively).

6.5. Limitation and Future Work

Cache-based methods: Our method inherits the drawbacks of PCBPT and RISBPT (i.e., cache-based methods). The discrepancy between the target pdfs of the eye sub-path vertex and the nearest cache point (e.g., shadow boundaries) can deteriorate the rendering.

Difficult light paths: Similar to BPT and PCBPT, our method has difficulties in connecting specular-diffuse-glossy or glossy-diffuse-glossy paths. To handle such difficult paths, our method can be combined with a complementary method (e.g., HRR [TH19]), but we leave this for future work.

Increasing the number of proposals in the second stage: Although our two-stage resampling method can handle a large num-

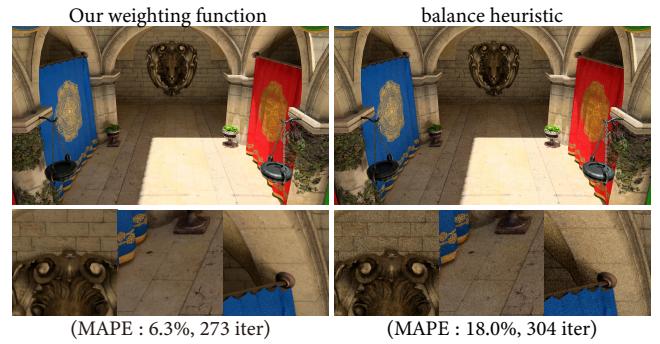


Figure 10: Comparison between our weighting function using p_{sr} and the balance heuristic using p_t which is incorporated into our two-stage resampling BPT. Simply incorporating our two-stage resampling incurs the computational overhead, leads to the increased noise. The combination of our weighting function and the two-stage resampling BPT can yield $2.86 \times$ smaller MAPEs in equal-time (10 min) rendering.

ber of proposals in total, it is currently difficult to increase the number of proposals used in the second resampling stage, same as PCBPT and RISBPT. However, as shown in Fig. 7, the MAPEs for small M_2 are comparable to the best cases thanks to our two-stage resampling.

In this paper, the two-stage resampling method is applied to BPT. We believe that two-stage resampling can be applied to other applications, such as direct illumination computation for many lights.

7. Conclusions

We presented a two-stage resampled importance sampling (RIS) method and applied it to BPT. Our two-stage resampling BPT can handle a two orders of magnitude larger number of pre-sampled light sub-paths, which can alleviate the artifacts due to light sub-paths with high contributions and improve the estimation accuracy of the weighting function, resulting in stable noise reduction. Two-stage resampling BPT enables us to use different subsets of pre-sampled light sub-paths at cache points, which can mitigate the path correlation due to the reuse of light sub-paths. We also proposed a weighting function that can handle changes in path probability due to two-stage resampling. By combining our two-stage resampling

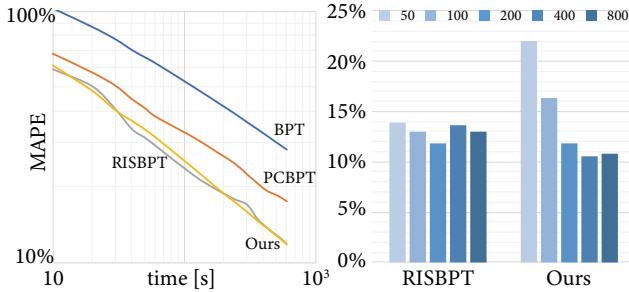


Figure 11: Convergence plots (left) for the Bedroom scene rendered by BPT, PCBPT, RISBPT, and our method, all of which are combined with path guiding with $M = 200$. In this case, MAPE of our method (11.79%) is slightly better than that of RISBPT (11.82%). MAPEs of RISBPT and our method combined with path guiding for varying numbers of pre-sampled light sub-paths $M (= M_2)$ are shown in the right.

BPT with the weighting function, our method achieves up to 2.3 times smaller MAPEs than state-of-the-art methods in equal-time rendering.

Acknowledgements

This research was supported by JSPS KAKENHI Grant Numbers 15H05924 and 18H03348. We would like to thank Prof. Wenzel Jakob for the proofreading of our early draft.

References

- [BMDS19] BAKO S., MEYER M., DEROSSE T., SEN P.: Offline deep importance sampling for Monte Carlo path tracing. *Computer Graphics Forum* 38, 7 (2019), 527–542. [2](#)
- [BWP*20] BITTERLI B., WYMAN C., PHARR M., SHIRLEY P., LEFOHN A., JAROSZ W.: Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics* 39, 4 (2020), 148:1–148:17. [2](#)
- [CBH*18] CHAITANYA C. R. A., BELCOUR L., HACHISUKA T., PREMOZE S., PANTALEONI J., NOWROUZEZHRAI D.: Matrix bidirectional path tracing. In *Eurographics Symposium on Rendering - Experimental Ideas & Implementations* (2018), pp. 23–34. [1](#)
- [CTC06] CLINE D., EGBERT P. K., TALBOT J. F., CARDON D. L.: Two Stage Importance Sampling for Direct Lighting. In *Eurographics Symposium on Rendering* (2006), pp. 103–113. [2](#)
- [DKH*14] DACHSBACHER C., KŘIVÁNEK J., HAŠAN M., ARBREE A., WALTER B., NOVÁK J.: Scalable realistic rendering with many-light methods. *Computer Graphics Forum* 33, 1 (2014), 88–104. [2](#)
- [DKHS14] DAVIDOVIČ T., KŘIVÁNEK J., HAŠAN M., SLUSALLEK P.: Progressive light transport simulation on the gpu: Survey and improvements. *ACM Transactions on Graphics* 33, 3 (2014), 29:1–29:19. [1](#)
- [GGSK19] GRITTMANN P., GEORGIEV I., SLUSALLEK P., KŘIVÁNEK J.: Variance-aware multiple importance sampling. *ACM Transactions on Graphics* 38, 6 (2019). [2](#)
- [GKDS12] GEORGIEV I., KŘIVÁNEK J., DAVIDOVIČ T., SLUSALLEK P.: Light transport simulation with vertex connection and merging. *ACM Transactions on Graphics* 31, 6 (2012), 192:1–192:10. [2, 8](#)
- [GKPS12] GEORGIEV I., KŘIVÁNEK J., POPOV S., SLUSALLEK P.: Importance caching for complex illumination. *Computer Graphics Forum* 31, 2 (2012), 701–710. [2](#)
- [HEV*16] HERHOLZ S., ELEK O., VORBA J., LENSCHE H., KŘIVÁNEK J.: Product importance sampling for light transport path guiding. *Computer Graphics Forum* 35, 4 (2016), 67–77. [2](#)
- [HPJ12] HACHISUKA T., PANTALEONI J., JENSEN H. W.: A path space extension for robust light transport simulation. *ACM Transactions on Graphics* 31, 6 (2012), 191:1–191:10. [2, 8](#)
- [Kel97] KELLER A.: Instant radiosity. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (1997), SIGGRAPH '97, pp. 49–56. [2](#)
- [KK04] KOLLIG T., KELLER A.: Illumination in the presence of weak singularities. In *Monte Carlo and Quasi-Monte Carlo Methods* (2004), pp. 245–257. [2](#)
- [KŠV*19] KARLÍK O., ŠIK M., VÉVODA P., SKŘIVAN T., KŘIVÁNEK J.: MIS compensation: Optimizing sampling techniques in multiple importance sampling. *ACM Transactions on Graphics* 38, 6 (2019). [2](#)
- [KVG*19] KONDAPANENI I., VÉVODA P., GRITTMANN P., SKŘIVAN T., SLUSALLEK P., KŘIVÁNEK J.: Optimal multiple importance sampling. *ACM Transactions on Graphics* 38, 4 (2019), 37:1–37:14. [2](#)
- [LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bi-directional path tracing. In *Computergraphics '93* (1993), pp. 145–153. [1](#)
- [MGN17] MÜLLER T., GROSS M., NOVÁK J.: Practical path guiding for efficient light-transport simulation. *Computer Graphics Forum* 36, 4 (2017), 91–100. [2](#)
- [MMR*19] MÜLLER T., MCWILLIAMS B., ROUSSELLE F., GROSS M., NOVÁK J.: Neural importance sampling. *ACM Transactions on Graphics* 38, 5 (2019), 145:1–145:19. [2](#)
- [NID20] NABATA K., IWASAKI K., DOBASHI Y.: Resampling-aware weighting functions for bidirectional path tracing using multiple light sub-paths. *ACM Transcacsions on Graphics* 39, 2 (2020), 15:1–15:11. [1, 2, 3, 7, 8, 10](#)
- [PBPP11] PAJOT A., BARTHE L., PAULIN M., POULIN P.: Combinatorial bidirectional path-tracing for efficient hybrid CPU/GPU rendering. *Computer Graphics Forum* 30, 2 (2011), 315–324. [1](#)
- [PRDD15] POPOV S., RAMAMOORTHI R., DURAND F., DRETTAKIS G.: Probabilistic connections for bidirectional path tracing. *Computer Graphics Forum* 34, 4 (2015), 75–86. [1, 2, 3, 7, 10](#)
- [Tal05] TALBOT J. F.: *Importance Resampling for Global Illumination*. Master's thesis, Brigham Young University, 2005. [2](#)
- [TCE05] TALBOT J. F., CLINE D., EGBERT P.: Importance resampling for global illumination. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques* (2005), EGSR '05, pp. 139–146. [2](#)
- [TH19] TOKUYOSHI Y., HARADA T.: Hierarchical russian roulette for vertex connections. *ACM Transactions on Graphics* 38, 4 (2019), 36:1–36:12. [2, 10, 11](#)
- [Vea97] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997. [3](#)
- [VG94] VEACH E., GUIBAS L.: Bidirectional estimators for light transport. pp. 145–167. [1, 3](#)
- [VG95] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques* (1995), SIGGRAPH '95, pp. 419–428. [1](#)
- [VKV*14] VORBA J., KARLÍK O., ŠIK M., RITSCHEL T., KŘIVÁNEK J.: On-line learning of parametric mixture models for light transport simulation. *ACM Transactions on Graphics* 33, 4 (2014), 101:1–101:11. [2, 10](#)
- [WFA*05] WALTER B., FERNANDEZ S., ARBREE A., BALA K., DONIKIAN M., GREENBERG D. P.: Lightcuts: A scalable approach to illumination. *ACM Transactions on Graphics* 24, 3 (2005), 1098–1107. [6](#)
- [WKB12] WALTER B., KHUNGURN P., BALA K.: Bidirectional lightcuts. *ACM Transactions on Graphics* 31, 4 (2012), 59:1–59:11. [2](#)