Resampling-aware Weighting Functions for Bidirectional Path Tracing using Multiple Light Sub-Paths

KOSUKE NABATA, Wakayama University, Japan
KEI IWASAKI, Wakayama University, Prometech CG Research, Japan
YOSHINORI DOBASHI, Hokkaido University, Prometech CG Research, Japan

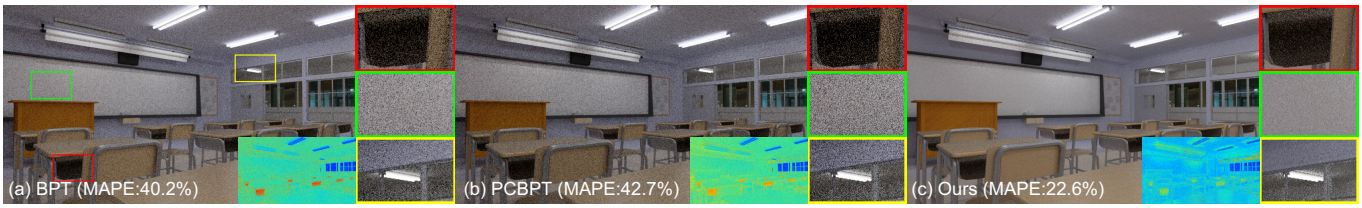


Fig. 1. Equal-time (1 min) comparisons between (a) BPT [Veach 1997], (b) PCBPT [Popov et al. 2015], and (c) our method of Classroom scene. Our resampling-aware weighting functions can substantially reduce the noise as shown in error images (insets) and mean absolute percentage errors (MAPEs).

Bidirectional path tracing (BPT) with multiple importance sampling (MIS) is a popular technique for rendering realistic images. Recently, it has been shown that BPT can be improved by preparing multiple light sub-paths and by resampling a small number of light sub-paths from them to generate full paths with large contribution. Traditionally, for MIS weights, the balance heuristic has widely been used to minimize the upper bound of variance, where each full path is weighted in proportion to the probability of the path. Although the probability of the path can change due to the resampling process, the weighting functions used in the previous methods remain unaffected by the change in probability, resulting in less efficiency. To address this problem, we propose new weighting functions for BPT with multiple light sub-paths. Our main contribution is a precise formulation of the variance and the derivation of the weighting functions that can appropriately treat the change in probability. We demonstrate that our weighting functions significantly improve the image quality. We will release a simple version of our implementation as open source to ensure reproducibility.

CCS Concepts: • **Computing methodologies** → **Rendering**.

Additional Key Words and Phrases: Bidirectional Path Tracing, Multiple Importance Sampling, Resampling Importance Sampling

ACM Reference Format:
Kosuke Nabata, Kei Iwasaki, and Yoshinori Dobashi. 2020. Resampling-aware Weighting Functions for Bidirectional Path Tracing using Multiple Light Sub-Paths. *ACM Trans. Graph.* 1, 1, Article 1 (January 2020), 11 pages. https://doi.org/10.1145/3338994

Authors' addresses: Kosuke Nabata, Wakayama University, 930 Sakaedani, Wakayama, Japan, 640-8510, nabata.kosuke@g.wakayama-u.ac.jp; Kei Iwasaki, Wakayama University, Prometech CG Research, 930 Sakaedani, Wakayama, Japan, 640-8510, iwasaki@wakayama-u.ac.jp; Yoshinori Dobashi, Hokkaido University, Prometech CG Research, Kita 14, Nishi 9, Kita-ku, Sapporo, Japan, 060-0814, doba@ime.ist.hokudai.ac.jp.

0730-0301/2020/1-ART1
https://doi.org/10.1145/3338994

1 INTRODUCTION

Bidirectional path tracing (BPT) is one of the most versatile methods in light transport simulation. BPT generates sub-paths from the camera and the light sources, and a full light path is created by connecting the vertices in an eye sub-path to each of the vertices in a light sub-path. The pixel intensity is then estimated using multiple importance sampling (MIS) [Veach 1997]. Each path sample is weighted by a weighting function. The balance heuristic that weights a path sample in proportion to the probability density of the path has widely been used as a weighting function. The robustness of BPT is a consequence of using weighting functions that aim to reduce variance.

In the traditional BPT framework, only a single light sub-path is generated and connected to a given eye sub-path. Popov et al. [2015] generalized BPT by using multiple light sub-paths and developed probabilistic connections for bidirectional path tracing (PCBPT). This method generates a light sub-path from a pre-sampled set of multiple light sub-paths; firstly, a set of light sub-paths are generated by probabilistically sampling the path space with a certain probability density function (pdf), and then a small number of light sub-paths are *resampled* probabilistically from the sample set. Similar to the balance heuristic, the MIS weights used in PCBPT are proportional to the pdf of the path. This resampling process can change the probability of the resampled light sub-path from the pdf used in generating the pre-sampled set. A key issue is that the weighting functions used in PCBPT remain unaffected by the change in the pdfs of resampled light sub-path samples, making the MIS weights inefficient.

In this paper, we focus on the weighting functions used in the above generalized BPT framework. We propose new weighting functions which are theoretically derived. Our formulation reveals that the weighting functions depend on the size of the pre-sampled set of light sub-paths. When the size is small, the balance heuristic works fine but it becomes less effective when many paths have been sampled. Our weighting functions automatically adapt to the number

of pre-sampled light sub-paths. We derive them by interpreting the *resampling* of a light sub-path from multiple pre-sampled light sub-paths as a *resampled importance sampling* (RIS) process [Talbot 2005]. Our theoretical contribution resides in the precise formulation of new RIS-based estimators for the pixel intensity, which we call *resampling estimators*, and their variance based on this interpretation, which allows us to derive the weighting functions.

The contributions of our method can be summarized as follows:

- we derive the resampling estimator and its variance based on an interpretation of the generation of full paths in BPT as RIS from multiple light sub-paths,
- we provide new weighting functions that can appropriately handle the change in pdfs with respect to the number of pre-sampled light sub-paths.

2 RELATED WORK

BPT is a robust light transport algorithm that samples paths by tracing these from both the camera and the light source, and connecting the vertices of the eye sub-path to those of the light sub-path [Lafortune and Willems 1993; Veach and Guibas 1994, 1995]. Davidovič et al. were the first to propose reusing light sub-paths in BPT within a GPU implementation [Davidovič et al. 2014]. The full connected paths in BPT, however, do not always have large contributions (e.g., the visibility between the vertices of an eye sub-path and a light sub-path is not considered when connected).

To address this problem, Popov et al. [2015] proposed probabilistic connections for bidirectional path tracing (PCBPT) that first sample multiple light sub-paths shared by the eye sub-paths and then resample a small number of connections through importance sampling. Unfortunately, the MIS weighting functions proposed by Popov et al. do not handle the change in pdfs caused by resampling, making the combination of these multiple strategies inefficient. We address this problem by deriving the variance for multiple light sub-paths and weighting functions minimizing the upper bound of variance.

Adaptive importance sampling methods also aim to generate important light paths by using precomputed radiance and importance distributions. Vorba et al. [2014] represent directional sampling distributions using a Gaussian mixture model (GMM) for path guiding. Herholz et al. [2016] extended Vorba's method by representing the product of the bidirectional scattering distribution function (BSDF) and the directional distribution of radiance or importance as GMM. Müller et al. [2017] proposed an adaptive spatio-directional tree (SD-tree) to represent the incident radiance field for path guiding. Recent advances in adaptive importance sampling have improved the direct illumination computation for many lights [Estevez and Kulla 2018; Vévoda et al. 2018]. Since our method does not impose any restrictions on sub-path generation, adaptive importance sampling methods (e.g., path guiding) can be combined with our method as shown in Sec. 6.1.2.

Many-light rendering methods [Dachsbacher et al. 2014; Keller 1997] generate many light sub-paths but the length of each eye sub-path is restricted to one. The last vertex of each eye sub-path is connected to all the vertices of the many light sub-paths, resulting in a prohibitive computation time. While efficient methods [Georgiev et al. 2012b; Hašan et al. 2007; Walter et al. 2006, 2005] have been proposed to accelerate many-light rendering by resampling a small number of light sub-paths, these methods use only one path sampling technique and suffer from splotches due to high variance. To alleviate this problem, several methods relaxed the restriction on the eye sub-path length and used multiple path sampling techniques combined with MIS [Kollig and Keller 2004; Walter et al. 2012]. Since they clamp splotches and suffer from energy loss due to clamping, these methods derive MIS weights to compensate for the energy loss, while our MIS weights are derived to minimize the upper bound of variance.

Photon mapping (PM) also uses multiple light sub-paths where the light sub-path vertices are called photons. In recent years, unified sampling methods [Georgiev et al. 2012a; Hachisuka et al. 2012] that combine PM and BPT using MIS have been proposed to increase robustness. Since our method is an extension of BPT, it can be combined with PM by replacing BPT with our method as demonstrated in Sec. 6.1.3.

Kondapaneni et al. [2019] derived optimal MIS weighting functions by using negative weights. This method, however, requires to solve the linear system whose computational cost grows superlinearly with the number of sampling techniques. Karlík et al. [2019] proposed a new technique called *MIS-compensation* that designs the pdf of one sampling technique to reduce the variance. Grittmann et al. [2019] presented variance-aware MIS weighting functions that injects variance estimates into weighting functions and applied them to BPT. Although these methods improve the efficiency and the robustness of MIS, none of these methods have been applied to BPT with multiple light sub-paths.

3 BACKGROUND

3.1 Resampled Importance Sampling

Talbot et al. introduced resampled importance sampling (RIS) to computer graphics. This method can generate samples approximately proportional to any target distribution [Talbot 2005]. Suppose we want to estimate the integral $I = \int_\Omega f(x) d\mu(x)$ where μ is a measure in the integration domain Ω. The integrand f is well approximated by a function $q^*(x)$, which is not necessarily normalized. $q(x)$ is the probability density function (pdf) normalized by integrating q^* over Ω (i.e., $q(x) = q^*(x)/\int_\Omega q^*(x')d\mu(x')$), but it is difficult to sample directly from q. We refer to q^* and q as *target distribution* and *target pdf*, respectively.

RIS first generates M proposals $\{X_1, \ldots, X_M\}$ from the pdf p, which is easy to sample. Then N samples $\{Y_1, \ldots, Y_N\}$ are drawn from the M proposals using the probability proportional to $q^*(Y)/p(Y)$ (we follow the RIS algorithm in Talbot's master thesis [Talbot 2005], while another paper on RIS [Talbot et al. 2005] proposes a slightly different algorithm that prepares M proposals for each sample Y). The RIS estimator $\hat{I}_{ris}$ [Talbot 2005, Eq. (4.1) on p. 29] is expressed

by:

$$\hat{I}_{ris} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(Y_i)}{q^*(Y_i)}\cdot\frac{1}{M}\sum_{j=1}^{M}\frac{q^*(X_j)}{p(X_j)} = \frac{1}{MN}\sum_{i=1}^{N}\frac{f(Y_i)}{p(Y_i)}\left(\frac{\sum_{j=1}^{M}\frac{q^*(X_j)}{p(X_j)}}{\frac{q^*(Y_i)}{p(Y_i)}}\right)$$
$$= \frac{1}{MN}\sum_{i=1}^{N}\frac{f(Y_i)}{p(Y_i)P_r(Y_i|\{X_1,\ldots,X_M\})},$$

where P_r is the probability mass function (pmf) for sampling Y_i defined as:

$$P_r(Y|\{X_1,\ldots,X_M\}) = \frac{q^*(Y)/p(Y)}{\sum_{j=1}^{M} q^*(X_j)/p(X_j)}. \tag{1}$$

The variance of the RIS estimator $V[\hat{I}_{ris}]$ is given by the following equation: [Talbot 2005, Eq. (4.2) on p. 30]:

$$V[\hat{I}_{ris}] = \frac{1}{M}V\left[\frac{f}{p}\right] - \frac{1}{MN}V\left[\frac{f}{q}\right] + \frac{1}{N}V\left[\frac{f}{q}\right]. \tag{2}$$

The above equation indicates that $V[f/q]$ becomes dominant as M increases. RIS can thus generate samples approximately proportional to the pdf q, rather than p, for sufficiently large M.

3.2 Multiple Importance Sampling in BPT

In the path integral formulation, the pixel measurement I is given by:

$$I = \int_{\Omega} f(\bar{x})d\mu(\bar{x}), \tag{3}$$

where Ω is the space of the light paths and $\bar{x} = x_0 \ldots x_k$ is a light path of length $k \geq 1$, where the path vertices x_0 and x_k are on a light source and the camera, respectively. $d\mu(\bar{x}) = dA(x_0)\ldots dA(x_k)$ is the differential area product and f is the measurement contribution function as:

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

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.

MIS [Veach 1997] is an integral estimation method that uses multiple sampling strategies. In BPT, each sampling strategy is identified by a pair of non-negative integers (s,t); the sampling strategy (s,t) connects the s-th vertex in a light sub-path to the t-th vertex in an eye sub-path to create a full light path $\bar{x} = \bar{y}\bar{z} = y_0\ldots y_{s-1}z_{t-1}\ldots z_0$, where $\bar{y} = y_0\ldots y_{s-1}$ represents the light sub-path and $\bar{z} = z_0\ldots z_{t-1}$ the eye sub-path. The pdf $p_{s,t}(\bar{x})$ of the full path is given by the product of those of the eye and the light sub-paths, $p_s(\bar{y})$ and $p_t(\bar{z})$, respectively. Using MIS, the pixel measurement I can be estimated by:

$$I \approx \sum_{s,t}\frac{1}{n_{s,t}}\sum_{i=1}^{n_{s,t}} w_{s,t}(\bar{X}_{s,t,i})\frac{f(\bar{X}_{s,t,i})}{p_{s,t}(\bar{X}_{s,t,i})}, \tag{4}$$

where $w_{s,t}$ is the weighting function, $n_{s,t}$ the number of samples, and $\bar{X}_{s,t,i}$ the i-th path sample of the sampling strategy (s,t). The key to the efficiency of MIS is that the weighting function affects the variance, that is, large weights should be assigned to strategies with low variance. The *balance heuristic* is the most widely-used weighting function [Veach 1997]:

$$w_{s,t}(\bar{x}) = \frac{n_{s,t}p_{s,t}(\bar{x})}{\sum_{s',t'} n_{s',t'}p_{s',t'}(\bar{x})}. \tag{5}$$

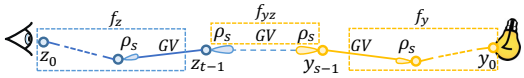
3.3 Probabilistic Connections for BPT (PCBPT)

PCBPT importance-samples connections between each vertex of the multiple light sub-paths $\{\bar{y}_1,\bar{y}_2,\ldots\}$ and each vertex of the eye sub-path $\bar{z}$ to generate a full light path. To generate a full light path with large contributions, the pmf P_r proportional to the contribution f of the full light path is calculated as follows:

$$P_r(\bar{x}_j) = \frac{f(\bar{x}_j)/p(\bar{x}_j)}{\sum_{i=1} f(\bar{x}_i)/p(\bar{x}_i)} = \frac{f_{yz}(\bar{y}_j,\bar{z})f_y(\bar{y}_j)/p(\bar{y}_j)}{\sum_{i=1} f_{yz}(\bar{y}_i,\bar{z})f_y(\bar{y}_i)/p(\bar{y}_i)},$$

where $\{\bar{x}_1,\bar{x}_2,\ldots\}$ is a set of potential full light paths that are generated by connecting each vertex of pre-sampled light sub-paths and the eye sub-path $\bar{z}$. The contribution f is decomposed into $f_z f_{yz} f_y$ where f_z and f_y depend only on the eye sub-path and the light sub-path respectively, and f_{yz} depends on both the eye and light sub-paths as:

$$f_y(\bar{y}) = L_e(y_0,y_1)GV(y_0,y_1)\prod_{i=1}^{s-2}\rho(y_{i-1},y_i,y_{i+1})GV(y_i,y_{i+1}),$$
$$f_{yz}(\bar{y},\bar{z}) = \rho(y_{s-2},y_{s-1},z_{t-1})GV(y_{s-1},z_{t-1})\rho(y_{s-1},z_{t-1},z_{t-2}),$$
$$f_z(\bar{z}) = W_e(z_0,z_1)GV(z_0,z_1)\prod_{i=1}^{t-2}\rho(z_{i-1},z_i,z_{i+1})GV(z_i,z_{i+1}).$$



$f_z(\bar{z})/p(\bar{z})$ is canceled out in $P_r(\bar{x}_j)$ since the eye sub-path $\bar{z}$ is common. Instead of constructing pmfs for each eye sub-path vertex, they are constructed on cache points that are distributed on the surface of the scene, and those for each eye sub-path vertex are interpolated by those stored at nearby cache points.

Importance sampling of connections for pre-sampled light sub-paths in PCBPT can be interpreted as resampling from the proposals in RIS. Specifically, the set of potential full light paths can be considered as the proposals in RIS, and the pmf for RIS is considered to be the same as that of PCBPT when $f_y f_{yz}$ is used as the target distribution q^*. Resampling from the proposals causes changes to the pdf, as explained in Sec. 3.1. However, PCBPT uses the sampling pdf p of the light sub-paths for the weighting function [Popov et al. 2015, Eqs. (16) and (17)], rather than the pdf proportional to $f_y f_{yz}$, resulting in less efficient MIS weights. Thus, developing a weighting function by a proper treatment of the changes to the pdf in the resampling process remains a challenging problem.

We focus on developing the weighting functions to solve this problem. Our method builds upon PCBPT that iteratively generates full light paths. Although the resampling process of our method is almost identical to that of PCBPT, our weighting functions yield superior results as shown in Figs. 1, 2 and 4.

Table 1. Notations and symbols. 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
$\bar{x}$	full light path $x_0 \dots x_k$
$\bar{y}$, $\bar{z}$	light sub-path and eye sub-path
$\bar{Y}_{s,i}$	i-th light sub-path sample with s vertices
$\bar{Z}_t$	eye sub-path sample with t vertices
M	number of light sub-path samples
N	number of resampling light sub-paths per iteration
A, A^s, $\mathcal{A}$	scene surface, s-dimensional product of A, union of A^s
$q^*(\bar{x})$	target distribution (a part of contribution function f)
$q(\bar{x})$, $q_s(\bar{x})$	target pdfs (normalize q^* over $\mathcal{A}$ and A^s)
Q, Q_s	normalization factors (integrate q^* over $\mathcal{A}$ and A^s)

4 RESAMPLING-AWARE WEIGHTING FUNCTION

In this section, we develop *resampling-aware* weighting functions for BPT using multiple light sub-paths, which can handle the change made to the path pdf due to resampling of light sub-paths. We explain the basic idea on how we derive the weighting function. A detailed derivation is provided in the supplemental material.

We focus on developing the weighting functions for the sampling strategy that connects an eye sub-path with $t \geq 2$ vertices to a light sub-path *resampled* from a pre-sampled set of light sub-paths. The pixel measurement I_t for this sampling strategy corresponds to contributions from all the paths having eye sub-paths with t vertices and light sub-paths with arbitrary length. We refer to the estimator for the pixel measurement I_t as *resampling estimator*. Other strategies including the unidirectional sampling techniques ($s = 0, t \geq 2$) and ($s \geq 2, t = 0$), and the light tracing ($s \geq 1, t = 1$) are handled by the traditional BPT, since the unidirectional sampling techniques do not require connections and the light tracing is efficiently handled by BPT.

4.1 The Resampling Estimator and Its Variance

In this section, we describe the RIS formulation of the resampling estimator $\hat{I}_t$ and its variance $V[\hat{I}_t]$. $\hat{I}_t$ is computed by generating eye sub-paths with $t(\geq 2)$ vertices connected to light sub-paths with arbitrary length. For a given eye sub-path sample $\bar{Z}_t$ with sampling pdf p, I_t is estimated by integrating f weighted by the weighting function w_t as follows:

$$\hat{I}_t = \frac{1}{p(\bar{Z}_t)} \int_{\mathcal{A}} w_t(\bar{y}\bar{Z}_t) f(\bar{y}\bar{Z}_t) d\mu(\bar{y}),$$

where $\bar{y}$ is an integral variable of light sub-paths, and $\bar{y}\bar{Z}_t$ is a full light path by connecting the last vertices of the eye sub-path sample $\bar{Z}_t$ and the light sub-path $\bar{y}$. The integral domain $\mathcal{A}$ is defined as $\mathcal{A} = \cup_{s\geq 1} A^s$, where A^s is the s-dimensional Cartesian product over the scene surface A. I_t is estimated by partitioning the integral domain $\mathcal{A}$ into each dimension A^s and generating M light sub-path samples from A^s as follows:

$$\hat{I}_t = \frac{1}{p(\bar{Z}_t)} \sum_{s\geq 1} \frac{1}{M} \sum_{i=1}^{M} \frac{w_t(\bar{Y}_{s,i}\bar{Z}_t) f(\bar{Y}_{s,i}\bar{Z}_t)}{p(\bar{Y}_{s,i})},$$

where $\bar{Y}_{s,i}$ is the i-th light sub-path sample with s vertices, $\bar{Y}_{s,i}\bar{Z}_t$ is a potential full light path connecting $\bar{Y}_{s,i}$ and the eye sub-path $\bar{Z}_t$.

In the context of BPT with multiple light sub-paths, we have a pre-sampled set of light sub-paths, which is constructed by tracing M light sub-paths from the light sources. Each of the paths as well as its sub-paths are included in the pre-sampled set.[1] We represent a proposal for RIS by a light sub-path $\bar{Y}_{s,i}$ and represent the set of proposals $\bar{\mathbf{Y}} = \{\bar{Y}_{1,1}, \dots, \bar{Y}_{s,i}, \dots\}$. By drawing N samples from the set, the resampling estimator $\hat{I}_t$ is given by:

$$\hat{I}_t = \frac{1}{MN} \sum_{j=1}^{N} \frac{w_t(\bar{Y}_j\bar{Z}_t) f(\bar{Y}_j\bar{Z}_t)}{p(\bar{Y}_j) p(\bar{Z}_t) P_r(\bar{Y}_j|\bar{\mathbf{Y}})}, \tag{6}$$

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

where $\bar{Y}_j$ is the j-th light sub-path sample resampled from the set $\bar{\mathbf{Y}}$, and P_r is referred to as *resampling pmf*. The conditional variance of the resampling estimator $V\hat{I}_t|\bar{Z}_t$ is expressed by the following equation (the derivation is shown in Sec. 1 of the supplemental document):

$$V\left[\hat{I}_t|\bar{Z}_t\right] = \frac{1}{M}\sum_{s\geq 1} V\left[\frac{w_t f}{p}|\bar{Z}_t\right] - \frac{1}{MN}\sum_{s\geq 1} V\left[\frac{w_t f}{q_s}|\bar{Z}_t\right] + \frac{1}{N} V\left[\frac{w_t f}{q}|\bar{Z}_t\right], \tag{8}$$

where q and q_s are the target pdfs normalized by integrating the target distribution $q^*(\bar{x})$ over $\mathcal{A}$ and A^s, respectively. Please note that the above variance $V[\hat{I}_t|\bar{Z}_t]$ with no partitioning (i.e., $s = 1$ and $q_s = q$) matches the variance of the ordinary RIS in Eq. (2).

We now derive the variance $V[\hat{I}_t]$ from $V[\hat{I}_t|\bar{Z}_t]$ by considering the randomness of the eye sub-path $\bar{Z}_t$ (i.e., considering both $\hat{I}_t$ and $\bar{Z}_t$ as random variables). The variance $V[\hat{I}_t]$ is calculated by the law of total variance $V[\hat{I}_t] = E[V[\hat{I}_t|\bar{Z}_t]] + V[E[\hat{I}_t|\bar{Z}_t]]$ as follows (the derivation is shown in Sec. 2 of the supplemental document):

$$\begin{aligned} V\left[\hat{I}_t\right] = {} & \frac{1}{M}\sum_{s\geq 1} V\left[\frac{w_t f}{p}\right] - \frac{1}{MN}\sum_{s\geq 1} V\left[\frac{w_t f}{q_s}\right] + \frac{1}{N} V\left[\frac{w_t f}{q}\right] \\ & + \left(1 - \frac{1}{N}\right) V\left[\frac{1}{p(\bar{Z}_t)} \int_{\mathcal{A}} w_t(\bar{y}\bar{Z}_t) f(\bar{y}\bar{Z}_t) d\mu(\bar{y})\right] \\ & - \frac{1}{M}\left(1 - \frac{1}{N}\right) \sum_{s\geq 1} V\left[\frac{1}{p(\bar{Z}_t)} \int_{A^s} w_t(\bar{y}\bar{Z}_t) f(\bar{y}\bar{Z}_t) d\mu(\bar{y})\right]. \end{aligned} \tag{9}$$

To derive the weighting functions, we use the same derivation technique of the balance heuristic [Veach 1997, p. 288] that makes use of the fact that the weight of a path $w_t(\bar{y}\bar{Z}_t)$ is independent of the weights of other paths $w_t(\bar{y}'\bar{Z}_t)$. However, in our case, the variance terms including integrals cause additional dependencies between the paths. While one could simply ignore these terms, the resulting weights would be far from optimal. Our solution is to set

[1] For example, when the i-th light sub-path $\bar{Y}_i$ traced from a light source consists of three vertices, $\bar{Y}_i = y_0 y_1 y_2$, we include three light sub-paths, $\bar{Y}_{1,i} = y_0$, $\bar{Y}_{2,i} = y_0 y_1$, and $\bar{Y}_{3,i} = y_0 y_1 y_2$ in the pre-sampled set.

the number of resampling light sub-path samples N to one. This causes these dependencies to vanish, and we obtain:

$$V\left[\hat{I}_t\right] = \frac{1}{M}\sum_{s\geq 1} V\left[\frac{w_t f}{p}\right] - \frac{1}{M}\sum_{s\geq 1} V\left[\frac{w_t f}{q_s}\right] + V\left[\frac{w_t f}{q}\right]. \tag{10}$$

Although only one light sub-path from the pre-sampled light sub-paths is sampled, our method iteratively generates eye sub-paths and multiple light sub-paths to estimate the pixel measurement I. Therefore, the variance of the pixel measurement decreases inversely proportional to the number of iterations. Hereinafter, we focus on deriving the variance $V[\hat{I}]$ at each iteration.

4.2 MIS Weighting Function

We derived our weighting function by reducing the variance $V[\hat{I}]$ that includes $V[\hat{I}_t]$ in Eq. (10) and the variance terms for other strategies (i.e., unidirectional sampling and light tracing). To generate a path with length k, there are $k+2$ strategies and each strategy can be identified by the number of eye sub-path vertices t. The strategies with $t = 2, \ldots, k$ are handled by our method, and the other three strategies with $t = 0, 1, k+1$ are handled by BPT. We define $\Lambda_{IS} = \{0, 1, k+1\}$ and $\Lambda_{RIS} = \{2, \ldots, k\}$ to represent these two types of strategies.

Similar to the balance heuristic, instead of minimizing the variance $V[\hat{I}] = E[\hat{I}^2] - E[\hat{I}]^2$ itself, we consider the second moment $E[\hat{I}^2]$ as the upper bound of variance $V[\hat{I}]$ and minimizes the upper bound as:

$$\sum_{t\in\Lambda_{RIS}}\left(\frac{1}{M}E\left[\frac{w_t^2 f^2}{p^2}\right] - \frac{1}{M}E\left[\frac{w_t^2 f^2}{q_s^2}\right] + E\left[\frac{w_t^2 f^2}{q^2}\right]\right) + \sum_{t\in\Lambda_{IS}}\frac{1}{N_t}E\left[\frac{w_t^2 f^2}{p^2}\right],$$

where N_t is the number of samples for BPT with strategy t, and the summation over s vanishes since s is uniquely determined as $s = k+1-t$ for a path $\bar{x}$ with length k. We focus on a single light path $\bar{x}$ and drop $f^2(\bar{x})$ from the upper bound since $f(\bar{x})$ is constant for all strategies. This leads to the following for the upper bound as:

$$\sum_{t\in\Lambda_{RIS}}\left(\frac{1}{M}\frac{w_t^2(\bar{x})}{p(\bar{x})} - \frac{1}{M}\frac{w_t^2(\bar{x})}{q_s(\bar{x})} + \frac{w_t^2(\bar{x})}{q(\bar{x})}\right) + \sum_{t\in\Lambda_{IS}}\frac{1}{N_t}\frac{w_t(\bar{x})^2}{p(\bar{x})},$$

subject to the condition $\sum_{t=0}^{k+1} w_t = 1$. To further simplify the upper bound, we define the normalization factors[2] Q and Q_s for the target distribution q^* as:

$$Q = \int_{\mathcal{A}} q^*(\bar{x})d\mu(\bar{y}), \quad Q_s = \int_{A^s} q^*(\bar{x})d\mu(\bar{y}). \tag{11}$$

By using Q and Q_s, the following relation between the pdfs q_s and q holds:

$$\begin{aligned} q_s(\bar{x}) &= \frac{q^*(\bar{x})}{\int_{A^s} q^*(\bar{x})d\mu(\bar{y})} = \frac{q^*(\bar{x})}{\int_{\mathcal{A}} q^*(\bar{x}')d\mu(\bar{y}')}\cdot\frac{\int_{\mathcal{A}} q^*(\bar{x}')d\mu(\bar{y}')}{\int_{A^s} q^*(\bar{x})d\mu(\bar{y})} \\ &= q(\bar{x})\cdot\frac{Q}{Q_s}, \end{aligned}$$

[2]When the contribution $f_y f_{yz}$ is used as the target distribution q^*, the normalization factor Q is the radiance at the connection vertex z_{t-1} towards z_{t-2}, and Q_s is the fraction of the radiance due to light sub-paths with s vertices.

By substituting this relation, the variance bound simplifies to:

$$\sum_{t\in\Lambda_{RIS}}\left(\frac{1}{M}\frac{1}{p(\bar{x})} + \left(1-\frac{1}{M}\frac{Q_s}{Q}\right)\frac{1}{q(\bar{x})}\right)w_t(\bar{x})^2 + \sum_{t\in\Lambda_{IS}}\frac{1}{N_t}\frac{w_t(\bar{x})^2}{p(\bar{x})}.$$

To unify the notation, we define the following function p_{ris}:

$$p_{\text{ris}}(\bar{x}) = \left(\frac{1}{M}\frac{1}{p(\bar{x})} + \left(1-\frac{1}{M}\frac{Q_s}{Q}\right)\frac{1}{q(\bar{x})}\right)^{-1}.$$

Then the bound simplifies to $\sum_{t=0}^{k+1} w_t^2(\bar{x})/n_t p_t(\bar{x})$, where n_t denotes the sample count and p_t is the associated density:

$$n_t = \begin{cases} 1 & (t\in\Lambda_{RIS}) \\ N_t & (t\in\Lambda_{IS}) \end{cases}, \quad p_t(\bar{x}) = \begin{cases} p_{\text{ris}}(\bar{x}) & (t\in\Lambda_{RIS}) \\ p(\bar{x}) & (t\in\Lambda_{IS}) \end{cases}. \tag{12}$$

Using these definitions, we are able to use the balance heuristic [Veach 1997, p. 289]:

$$w_t(\bar{x}) = \frac{n_t p_t(\bar{x})}{\sum_{i=0}^{k+1} n_i p_i(\bar{x})}. \tag{13}$$

The key difference to the balance heuristic used in BPT and PCBPT is that the density p_{ris} explicitly accounts for the resampling step.

To use the weighting function, two normalization factors Q and Q_s are required. Although these normalization factors can be estimated using Monte Carlo methods, we cannot keep all the values of Q_s for arbitrary non-negative integers s. Fortunately, the value Q_s/Q is bound between zero and one, since the integral over $\mathcal{A}$ is the sum of all the integrals over A^s. We approximate the value by the upper bound, i.e., one, assuming that the parameter M is sufficiently large. In summary, $p_{\text{ris}}(\bar{x})$ is calculated by:

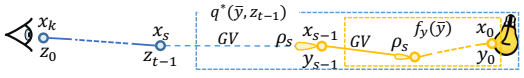
$$p_{\text{ris}}(\bar{x}) = \left(\frac{1}{M}\frac{1}{p(\bar{x})} + \left(1-\frac{1}{M}\right)\frac{1}{q(\bar{x})}\right)^{-1}. \tag{14}$$

This function has the desired property that p_{ris} approaches the target density $q(\bar{x})$ as M increases.

4.3 Target Distribution q^* and Target Pdf q

To use the density p_{ris} and the resampling pmf P_r in Eq. (7), we need an actual form of the target distribution q^* and its pdf q. Similar to PCBPT, we approximate the resampling pmf P_r at each eye sub-path vertex with that computed for the closest cache point for efficiency. Our target distribution, which we design conservatively to reduce the error caused by this approximation, is:

$$q^*(\bar{x}) = q^*(\bar{y}, z_{t-1}) = f_y(\bar{y})\rho(y_{s-2}, y_{s-1}, z_{t-1})GV(y_{s-1}, z_{t-1}). \tag{15}$$



Although PCBPT uses $f_y f_{yz}$ to construct the resampling pmf P_r, the BSDF ρ at the eye sub-path vertex z_{t-1} in f_{yz} can be significantly different from that at the closest cache point of z_{t-1} (e.g., z_{t-1} is on a highly glossy material while the cache point is on a diffuse material). Therefore, we ignore the BSDF at the eye sub-path vertex z_{t-1} in the target distribution q^*. The target pdf $q(\bar{x})$ is calculated

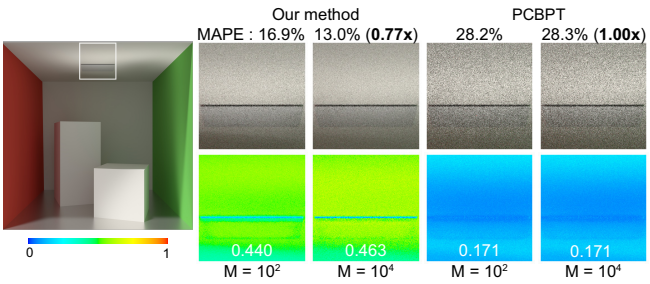


Fig. 2. Comparison between our weighting functions and those used in PCBPT [Popov et al. 2015] in equal-iteration (16) rendering. The relative contributions of $t = 2$ strategy ($\int_\Omega w_2(\bar{x})f(\bar{x})d\mu(\bar{x}) / \int_\Omega f(\bar{x})d\mu(\bar{x})$) for our method and PCBPT are shown in the second row in false color. As the number of the pre-sampled light sub-paths M increases from 10^2 to 10^4, the relative contribution for $t = 2$ increases and the MAPE decreases in our method as shown in the MAPE improvement (in parenthesis), while those do not change in PCBPT.

by the product of the target distribution $q^*(\bar{x})$ normalized by the normalization factor Q (Eq. (11)) and the pdf of the eye sub-path as:

$$q(\bar{x}) = \frac{q^*(\bar{y}, z_{t-1})}{Q} p(\bar{z}) = \frac{q^*(\bar{y}, z_{t-1})}{\int_{\mathcal{A}} q^*(\bar{y}', z_{t-1}) d\mu(\bar{y}')} p(\bar{z}) = q(\bar{y}|z_{t-1})p(\bar{z})$$
$$= q(x_0 \dots x_{s-1}|x_s)p(x_s \dots x_k), \tag{16}$$

where the full light path $\bar{x}$ consists of the light sub-path $\bar{y}$ and the eye sub-path $\bar{z}$ as $\bar{x} = \bar{y}\bar{z} = y_0 \dots y_{s-1} z_{t-1} \dots z_0 = x_0 \dots x_{s-1} x_s \dots x_k$.

4.4 Discussion

We now discuss the relationship between our resampling-aware weighting functions and those used in PCBPT, and compare the effects of weighting functions using different numbers of pre-sampled light sub-paths M. Our resampling-aware weighting functions subsume the weighting functions used in PCBPT. That is, the weighting functions of PCBPT are represented by using $p_t(\bar{x}) = p(\bar{x})$ for all strategies in Eq. (12). In other words, the weighting functions of PCBPT are a special case of our weighting functions using the density $p_{\text{ris}}(\bar{x})$ with $M = 1$ in Eq. (14). This indicates that the weighting functions of PCBPT are unaware of the change of the pdf due to resampling and do not benefit from the increase in the number of pre-sampled light sub-paths M, while our resampling-aware weighting functions can handle the change of the pdf by blending the sampling pdf p and the target pdf q where the reciprocal of the number of pre-sampled light sub-paths M acts as an interpolation factor.

Fig. 2 shows an equal-iteration comparison between the weights of our method and PCBPT using different numbers of pre-sampled light sub-paths M ($M = 10^2$ and $M = 10^4$). To equalize the target distribution between PCBPT and ours, $q^* = f_y f_{yz}$ is used for our method in this comparison while Eq. (15) is used in other results. Our weighting functions considerably reduce the variance compared with those used in PCBPT. The bottom row visualizes the relative contributions of the $t = 2$ strategy for our method and PCBPT, respectively. Our method assigns larger weights to the $t = 2$ strategy than PCBPT does, resulting in considerably reduced noise.

Algorithm 1 Our rendering algorithm. Our rendering algorithm builds upon PCBPT and differences between PCBPT and ours are highlighted as underlines. Note that the pseudocode for the weighting function w_t (lines 16-22) is a proof of concept. Our implementation uses more efficient recursive formulations to compute w_t. Please see our sample code for details.

1: **for** $n \leftarrow 1$ **to** maxIterations **do**
2: generate $\bar{\mathbf{Y}}_n$ by tracing M paths from light sources
3: generate cache points
4: **for** each cache point c **do**
5: calculate resampling pmf P_r using q^* ▷ Eqs. (15)(17)
6: estimate Q using $\bar{\mathbf{Y}}_{n-1}$
7: normalize target distribution q^*/Q ▷ Eq. (16)
8: **for** each pixel **do**
9: generate one light sub-path and one eye sub-path $\bar{z}$
10: calculate estimators for other strategies using BPT
11: **for** each vertex of $\bar{z}$ with $(t \geq 2)$ **do**
12: find the nearest cache point c_t
13: resample light sub-path $\bar{y}$ from $\bar{\mathbf{Y}}_n$ using cached P_r
14: *//calculate resampling estimator* $\hat{I}_t$
15: generate a full path $\bar{x} = y_0 \dots y_{s-1} z_{t-1} \dots z_0$
16: *//calculate resampling-aware weighting function* w_t
17: $w_t \leftarrow \sum_{i \in \Lambda_{IS}} n_i p_i(\bar{x})$
18: **for** $i \leftarrow 2$ **to** k **do** ▷ $i \in \Lambda_{RIS}$
19: find the nearest cache point c_i for x_{k-i+1}
20: calculate $q(\bar{x})$ using cached $q(x_0 \dots x_{k-i}|c_i)$
21: $w_t \leftarrow w_t + p_{\text{ris}}(\bar{x})$ ▷ Eq. (14)
22: $w_t \leftarrow n_t p_t(\bar{x}) / w_t$ ▷ Eq. (13)
23: calculate $\hat{I}_t$ using w_t and cached P_r ▷ Eq. (6)
24: update pixel intensity using $\hat{I}_t$ and other estimators

Moreover, as shown in the third column ($M = 10^4$), our weighting functions assign more weight to the resampling strategies whose variance decreases as M increases, but the improvement in PCBPT is subtle because the weighting functions for the resampling strategies remain unchanged.

5 IMPLEMENTATION DETAILS

5.1 Rendering Algorithm

Algorithm 1 summarizes our rendering approach. Our method renders an image by iteratively updating the pixel intensities. At each iteration n, we first generate a pre-sampled set $\bar{\mathbf{Y}}_n$ of light sub-paths, which is constructed by tracing M paths from the light sources. Next, a set of cache points is generated on the surface of the scene. For each cache point c, the resampling pmf P_r is calculated (line 5), the normalization factor Q (Eq. (11)) is estimated (line 6), and the conditional pdf in Eq. (16) is calculated (line 7).

Calculation of the resampling pmf P_r. By using the target distribution q^* in Eq. (15), the resampling pmf P_r in Eq. (7) for a given

eye sub-path sample $\bar{Z}_t$ is rewritten as:

$$P_r(\bar{Y}_j|\bar{\mathbf{Y}}) = \frac{q^*(\bar{Y}_j, z_{t-1})/p(\bar{Y}_j)}{\sum_{s\geq 1}\sum_{i=1}^{M} q^*(\bar{Y}_{s,i}, z_{t-1})/p(\bar{Y}_{s,i})}. \quad (17)$$

Since the resampling pmf P_r in Eq. (17) now depends only on the position of the t-th eye sub-path vertex z_{t-1} and the set of pre-sampled light sub-paths, our method calculates the resampling pmf P_r at each cache point c (by substituting c for z_{t-1} in Eq. (17)).

Estimation of the normalization factor Q. We estimate the normalization factor Q at each cache point to normalize the target distribution q^*. One simple way is to use the light sub-paths in the pre-sampled set $\bar{\mathbf{Y}}_n$ at each iteration n, which are used to estimate the contribution function f. However, the error does not decrease linearly in log-scale as shown in Fig. 3. Since the estimate of Q is a random variable, our resampling-aware weighting function w_t is also a random variable. Then the contribution f and the weighting function w_t must be estimated independently, since the expected value of the product of two random variables is equal to the product of the expected values only if those random variables are uncorrelated.

We thus simply use light sub-paths generated in the previous iteration, $\bar{\mathbf{Y}}_{n-1}$, for estimating the normalization factor Q at the current iteration. Since pre-sampled sets $\bar{\mathbf{Y}}_n$ and $\bar{\mathbf{Y}}_{n-1}$ are independently generated, the MAPE for using the light sub-paths in the previous iteration $\bar{\mathbf{Y}}_{n-1}$ converges as shown in Fig. 3(a). However, the positions of the cache points are different between the current and the previous iterations. We thus use the average of Q estimated at the nearby N_q cache points in the previous iteration. The only exception is the first iteration; we estimate the normalization factor Q using the set of pre-sampled light sub-paths in the first iteration $\bar{\mathbf{Y}}_1$. This causes bias, but its influence is quickly reduced as the number of iterations increases.

Normalization of the target distribution q^*. By using the estimate of the normalization factor Q, the target pdf in Eq. (16) at the cache point c is calculated by substituting c for x_s in the conditional pdf as $q(x_0 \dots x_{s-1}|c)p(x_s \dots x_k)$. If the normalization factor Q is severely underestimated, the target pdf q is overestimated, resulting in assigning large weights to strategies with high variance. To address this problem, we clamp the lower bound of the ratio of pdfs $p(x_0 \dots x_{s-1})/q(x_0 \dots x_{s-1}|c)$ by a clamping parameter ϵ (10^{-3} is used in all examples).

Estimation of the pixel intensity I. Our method traces eye sub-paths for each pixel (line 9) to estimate the pixel intensity I. Our method traces one light sub-path that is used for other strategies (e.g., unidirectional path tracing) handled by BPT (line 10). For each vertex $z_{t-1} (t \geq 2)$ of the eye sub-path, nearest cache point c_t is searched and a light sub-path is resampled using the resampling pmf P_r stored at the cache point c_t (lines 12-13). Our method calculates the resampling estimator $\hat{I}_t$ (Eq. (6)) using the resampled light sub-path (lines 14-23). The weighting function w_t (Eq. (13)) in the resampling estimator $\hat{I}_t$ involves the computation of the density p_{ris} in Eq. (14) and the target pdf $q(\bar{x})$ (lines 20-21). Our method approximates the conditional pdf $q(x_0 \dots x_{s-1}|x_s)$ in Eq. (16) with $q(x_0 \dots x_{s-1}|c_t)$, as our method uses the resampling pmf P_r at the cache point c_t.

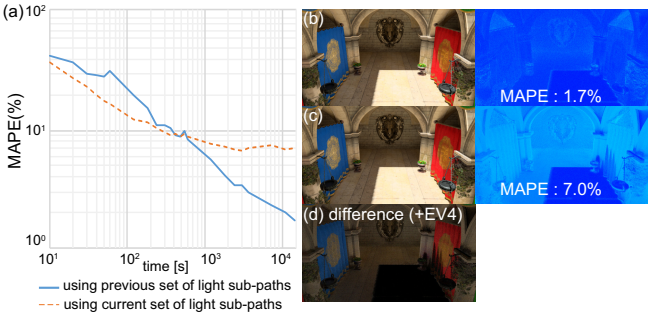


Fig. 3. (a) Log-log mean absolute percentage error (MAPE) plots using different pre-sampled sets of light sub-paths to estimate the normalization factor Q, (b) the set of pre-sampled light sub-paths in the previous iteration $\bar{\mathbf{Y}}_{n-1}$ (solid line) and (c) that in the current iteration $\bar{\mathbf{Y}}_n$ (dot line). (d) The difference (+EV4) between (b) and (c). Using the set of pre-sampled light sub-paths in the current iteration $\bar{\mathbf{Y}}_n$, the MAPE does not decrease linearly in log-log plot (dot line).

Computation of the resampling-aware weighting function w_t. The computation of the weighting function $w_t(\bar{x})$ in Eq. (13) involves the evaluations of the densities $p_i(\bar{x})$ for all $k+2$ strategies of a full light path $\bar{x}$ with length k (line 18). This indicates that the evaluations of the target pdf q are required for $(k-1)$ resampling strategies in $\Lambda_{RIS} = \{2, \dots, k\}$. Our method approximates each target pdf with that evaluated at each nearest cache point. Therefore, our method incurs $(k-1)$ times searches for nearest cache points to compute the weighting function $w_t(\bar{x})$ (line 19). We discuss the computational overhead due to our weighting functions in Sec. 6.2.

5.2 Multiple Cache Points

So far, we have described our resampling scheme for the case in which the closest cache point is used, but this might cause important light sub-paths to be missed due to the difference in the target distribution q^* between the eye sub-path vertex and the cache point. We solve this problem by using the closest N_c cache points and one additional cache point called the virtual cache point. The virtual cache point stores a uniform distribution for resampling in the same way as conservative sampling [Georgiev et al. 2012b]. We first select one cache point from among the (N_c+1) cache points using the pmf P_c (currently we use $P_c(i) = 1/(N_c+1)$), and then resample one light sub-path using the resampling pmf P_r stored at the selected cache point. Since this adds another source of randomness, the resampling estimator $\hat{I}_t$ and the density p_{ris} need slight modifications as follows:

$$\hat{I}_t = \frac{1}{M}\frac{w_t(\bar{Y}\bar{Z}_t)f(\bar{Y}\bar{Z}_t)}{p(\bar{Y})p(\bar{Z}_t)P_r(\bar{Y}|\bar{\mathbf{Y}})} \cdot \frac{p_{\text{ris,i}}(\bar{Y}\bar{Z}_t)}{p_{\text{ris}}(\bar{Y}\bar{Z}_t)},$$

$$p_{\text{ris}}(\bar{x}) = \sum_{i=1}^{N_c+1} P_c(i)p_{\text{ris,i}}(\bar{x}),$$

where $\bar{Y}$ is a light sub-path resampled from the set $\bar{\mathbf{Y}}$, and i is the index of the selected cache point using the pmf P_c. $p_{\text{ris,i}}(\bar{x})$ is calculated by Eq. (14) using the i-th closest cache point.

6 RESULTS AND DISCUSSION

In this section, first, we show comparisons between our method and previous methods (BPT and PCBPT) including combinations with the path guiding method [Vorba et al. 2014] and progressive photon mapping (PPM) [Hachisuka et al. 2008], and then we discuss the limitations of our method. Please see 'comparisons.html' in the supplemental material for the enlarged side-by-side comparisons.

6.1 Comparison to BPT and PCBPT

We compare our method with BPT and PCBPT in Figs. 1 and 4(a)(b), and show comparisons between our method combined with complementary algorithms (i.e., path guiding and PPM) and BPT/PCBPT with those algorithms in Fig. 4(c)(d). All these methods are implemented using our CPU ray tracer and all images are rendered on a PC with an Intel Core i9-7980XE CPU.

We generate cache points in the same way as PCBPT. $0.4\% \times W \times H$ eye sub-paths are generated, where W and H are the width and height of the image, and the vertices of the eye sub-paths are considered as cache points. The resolution of all images is 1280×720. The number of cache points for resampling, N_c, and that for the estimation of the normalization factor Q, N_q, are set to three. The clamping parameter ϵ is set to 10^{-3}. The number of light sub-paths M is 200. These values were determined experimentally as discussed in Sec. 6.1.5. We use the same target distribution of Eq. (15) for our method and PCBPT in all figures (except for Fig. 2), since we found that the target distribution used in PCBPT (i.e., $q^* = f_y f_{yz}$) makes the estimation of the normalization factor Q slightly unstable in our experiments. We show the rendering results of our method and PCBPT using the target distribution of PCBPT in the supplemental material.

Fig. 1 shows a Classroom scene illuminated by an environment map and area light sources on the ceiling. In this scene, the area light sources are located in the classroom and the hallway. Fig. 4(a) shows a Sponza scene illuminated by an environment map with high directionality. In this scene, a fraction of light enters the scene from the top of the building. Fig. 4(b) shows a Door scene illuminated by an area light source. The area light source is located in the next room, and light enters the room through a small slit in the door. The material of the floor is semi-glossy. Fig. 4(c) shows a Bedroom scene, where the generation of full paths is difficult due to the small gaps in the curtains. Fig. 4(d) shows a House scene mainly illuminated from light sources in the house.

Fig. 5 shows the convergence graphs of the mean absolute percentage error (MAPE) for BPT, PCBPT, and our method. According to the paper [Popov et al. 2015], the number of light sub-paths used in PCBPT is set to 100. For a fair comparison to PCBPT, the convergence graphs of our method with $M = 100$ are also shown in Fig. 5. As shown in all the graphs, our method outperforms BPT and PCBPT in terms of reducing the noise and MAPEs. MAPEs for our method with $M = 100$, however, do not decrease monotonically, though it eventually converges. On the other hand, our method with $M = 200$ provides stable noise reduction as shown in the convergence graphs. Details and experiments on different M are discussed in Sec. 6.1.5.

6.1.1 Discussion. As shown in these experiments, our method can greatly reduce the noise compared to BPT and PCBPT. Our method with $M = 200$ converges (a) 3.05, (b) 2.58, (c) 1.48, (d) 2.37, and (e) 1.37 times faster than PCBPT. Although our method outperforms BPT and PCBPT, it also uses the same strategies as BPT (e.g., unidirectional path tracing ($s = 0, t \geq 2$)), which do not benefit from the use of multiple light sub-paths. Therefore, when the contributions corresponding to those strategies are dominant, our method cannot get a performance gain due to the overhead for the high computation required for the weighting functions, which decreases the number of possible iterations in our method for the equal-time comparison. The smaller number of possible iterations is problematic for the case in which all the vertices on the full path are on specular materials or highly glossy materials.

6.1.2 Combination with path guiding. Our method can be combined with path guiding as shown in Fig. 4(c). Eye sub-paths and light sub-paths are generated by using path guiding. The online learning of GMM in the training phase is the same as path guiding, and the pdfs associated with the eye sub-paths and the light sub-paths are replaced with those calculated from GMM. Fig. 4(c) shows equal-time (10 min) comparisons of a Bedroom scene. As shown in Fig. 4(c), our method with path guiding outperforms BPT and PCBPT, which are also combined with path guiding.

6.1.3 Combination with PPM. Our method can be combined with PPM by using the unified sampling methods [Georgiev et al. 2012a; Hachisuka et al. 2012] as shown in Fig. 4(d). Since our method cannot handle connections to vertices on light sub-paths on specular surfaces similar to BPT and PCBPT, combining our method with PPM makes our rendering algorithm more robust. When combined with PPM, the strategies for PPM are added, and the pdf for vertex connection in the weighting function (Eq. (10) in the paper [Georgiev et al. 2012a]) is replaced with p_t in our method. Our method ignores the correlation in PPM and treats PPM as an additional uncorrelated technique. As shown in the insets for BPT and PCBPT, white spots are obvious, which may result from the assignment of large weights to the strategies for PPM, while our weighting function can alleviate the appearance of these white spots. As shown in Figs. 4(c) and (d), our method is a good alternative to BPT when combined with path guiding and PPM.

6.1.4 Comparison with path guiding. Fig. 8 shows equal-time comparisons (1 min for the Door scene and 10 min for the Bedroom scene, and both scenes have difficult visibilities) between our method without path guiding and BPT with path guiding [Vorba et al. 2014] which is also a caching-based solution. While our method yields smaller MAPEs than path guiding in the Door scene, path guiding is more efficient in the Bedroom scene since it includes the specular surfaces. There is a noticeable noise on the specular surface of the closet in the Bedroom scene, where our method has difficulties in handling such materials. However, as shown in Fig. 4(c), this noise can be reduced considerably by combining our method with path guiding.

6.1.5 Parameter selection. We evaluate the effects of the three parameters used in our method: the number of light sub-paths M, the number of resampling light sub-path samples N, and the number of

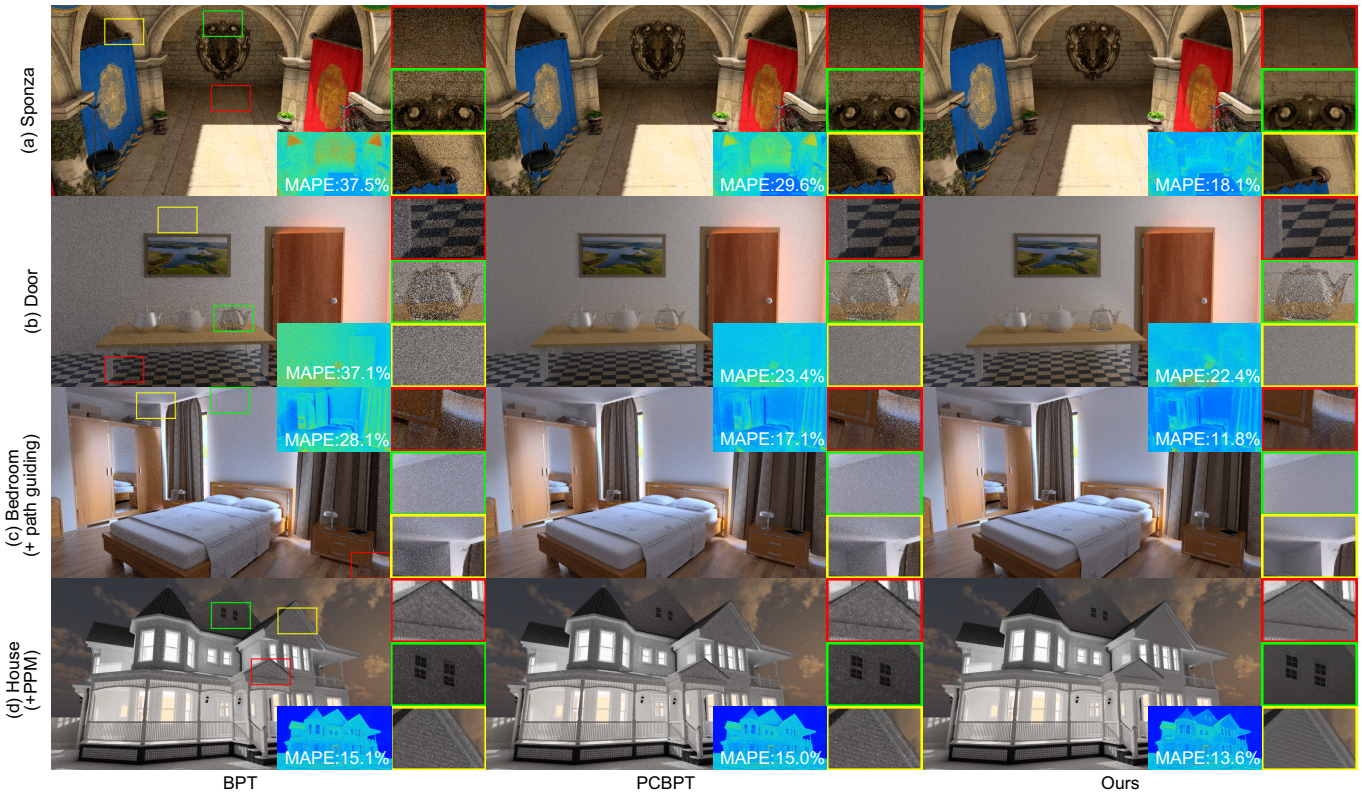


Fig. 4. Equal-time comparison between BPT, PCBPT, and our method for (a) Sponza, (b) Door, (c) Bedroom, and (d) House scenes. Sponza and Door scenes are rendered in 1 min and Bedroom and House scenes are rendered in 10 min. Bedroom scene is rendered using each method with path guiding, and House scene is rendered using each method with PPM. The insets show the absolute percentage errors with false color. Our method can provide superior results than BPT and PCBPT.

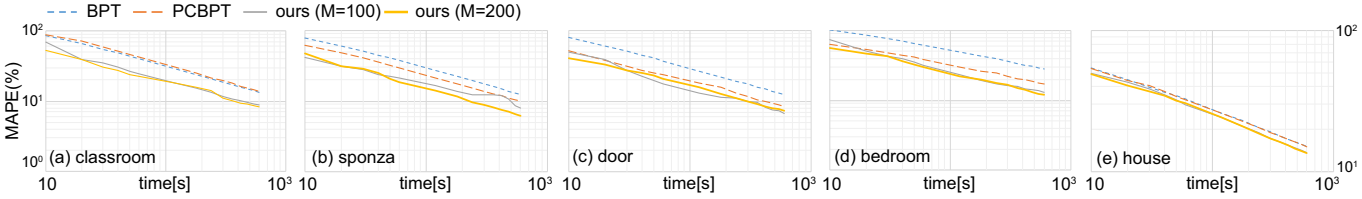


Fig. 5. Convergence graphs of MAPEs for (a) classroom (b) sponza (c) door (d) bedroom and (e) house scenes. Our method with $M = 100$ outperforms BPT and PCBPT, but the convergence graphs do not decrease monotonically, especially pronounced in (b). Our method with $M = 200$ provides stable noise reduction and the convergence graphs approach the straight lines similar to BPT.

nearest cache points N_c.

Number of light sub-paths M: Fig. 6 shows (a) MAPEs and (b) iteration counts in equal-time rendering (10 min) of all the scenes with various M. As M increases, the distribution of the path and p_{ris} approach the target pdf $q(\bar{x})$ that takes into account the connections between eye and light sub-paths, resulting in lower noise. However, it also increases the computational cost of the resampling pmf P_r, which involves evaluating the visibility term, and decreases the possible iteration counts as shown in Fig. 6(b). Fig. 7 shows convergence graphs with various M. For small M, MAPEs do not decrease monotonically and undulation can be seen. This is due to the effect of light sub-paths with high contribution. If a light sub-path with (extremely) high contribution is generated, this affects all the pixels in the image since our method shares light sub-paths with all eye sub-paths. On the other hand, the generation of light sub-paths with high contribution is alleviated for large M and the convergence graphs approach straight lines as shown in Figs. 7(d) and (e) at the cost of high computational overhead to construct the resampling pmf P_r. We found that $M = 200$ achieves a good balance between the stable noise reduction and the overhead of constructing the resampling pmf P_r.

Number of samples N: In Sec. 4, we have selected the number

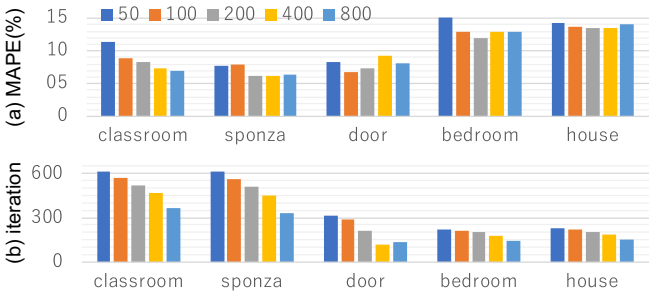


Fig. 6. (a) MAPEs and (b) iteration counts of the five scenes for $M = 50, 100, 200, 400$ and 800. The computational time is fixed to 10 min. Although the best performance depends on the scene, $M = 200$ works fine in most scenes.

of samples $N = 1$ to derive the weighting functions. We compare MAPEs between our weighting functions and those derived by ignoring the variance terms of integrals (without setting N to 1). Fig. 9 shows the graphs of MAPEs and iteration counts in equal-time rendering (10 min) for various N. Although the best performance depends on the scene, small N yields lower variance and large N impairs the reduction in variance due to the small number of possible iterations. In some scenes, $N = 2$ yields smaller MAPEs than $N = 1$ (e.g., in Sponza scene, MAPEs for $N = 1$ and $N = 2$ are 7.97% and 6.78%, respectively). This is because the computational cost to construct resampling pmfs is dominant and that to sample multiple ($N > 1$) light sub-paths at each iteration is much smaller, thus $N = 2$ can sample more light sub-paths in equal-time rendering. However, we used $N = 1$ in all the examples since the difference of MAPEs between $N = 1$ and $N = 2$ is subtle and using larger N definitely reduces the possible iteration counts, which is problematic as discussed in Sec. 6.1.1. This is pronounced in the Bedroom scene (MAPEs for $N = 1$ and $N = 2$ are 12.98% and 14.19%, respectively) where the effects of light paths that our method cannot improve (i.e., all the vertices are on specular materials) are dominant.

Number of nearest cache points N_c: We evaluated the effects of N_c by rendering the Sponza scene in 1 min. MAPEs for $N_c = 1, 3, 5$ are 23.2%, 21.0%, and 22.8%, respectively. The iteration counts for $N_c = 1, 3, 5$ are 71, 57, and 47, respectively. Although our method selects and uses only one cache point from $N_c + 1$ cache points, N_c affects the computation of the density p_{ris}, and the increase in the number of closest cache points incurs a computational overhead, resulting in lower performance.

6.2 Discussion, Limitation, and Future Work

Although our method can accelerate the rendering by caching the resampling pmfs, it also inherits the drawbacks of caching. That is, in regions where the target distribution q^* varies considerably, q^* in the closest cache point is not a good approximation for that in a vertex of the eye sub-path. This is problematic in handling highly glossy materials. Due to this, our method does not include BSDF at the last vertex of the eye sub-path in the target distribution q^*. Since the target distribution q^* also varies considerably due to changes in visibility, noise increases around the boundaries of shadows. Effective cache points may not be obtained due to the sparse distribution of the cache points (e.g., regions far from the camera) even though q^* varies smoothly.

Besides the above drawbacks intrinsic to cache based methods, our weighting functions incur high computational costs compared to those of BPT. Our weighting functions require searching for neighboring cache points not only for vertices on eye sub-paths but also those on light sub-paths. In our current implementation, the computational time for the search takes up to 22% of the total rendering time. To calculate the weighting functions for a path with length k, $N_c(k-1)$ times computations of the pdf q are needed. Fig. 10 shows the failure case. Our method is inferior to BPT for simple scenes such as a Cornell-box due to the relatively high computational cost of calculating the weighting functions.

The limitations of our method come from the use of cache points. In future work, we would like to develop an efficient resampling method without caching, which can avoid costly nearest neighbor searches for cache points, resulting in accelerated computation of the weighting functions.

7 CONCLUSION

We have proposed new weighting functions for BPT with multiple light sub-paths. We interpret probabilistic connections of an eye sub-path and a light sub-path from multiple light sub-paths as resampling importance sampling. This interpretation enables us to derive a novel resampling estimator and a precise formulation of its variance. Our resampling-aware weighting functions are derived by minimizing the upper bound of variance and can appropriately handle the change in the pdfs of light sub-paths due to resampling. We have shown that our weighting functions can significantly reduce the noise compared to BPT and PCBPT.

REFERENCES

Carsten Dachsbacher, Jaroslav Křivánek, Miloš Hašan, Adam Arbree, Bruce Walter, and Jan Novák. 2014. Scalable Realistic Rendering with Many-Light Methods. *Computer Graphics Forum* 33, 1 (2014), 88–104.

Tomáš Davidovič, Jaroslav Křivánek, Miloš Hašan, and Philipp Slusallek. 2014. Progressive Light Transport Simulation on the GPU: Survey and Improvements. *ACM Transactions on Graphics* 33, 3 (2014), 29:1–29:19.

Alejandro Conty Estevez and Christopher Kulla. 2018. Importance Sampling of Many Lights with Adaptive Tree Splitting. *Proc. ACM Comput. Graph. Interact. Tech.* 1, 2, Article 25 (2018), 25:1–25:17 pages.

Iliyan Georgiev, Jaroslav Křivánek, Tomáš Davidovič, and Philipp Slusallek. 2012a. Light Transport Simulation with Vertex Connection and Merging. *ACM Transactions on Graphics* 31, 6 (2012), 192:1–192:10.

Iliyan Georgiev, Jaroslav Křivánek, Stefan Popov, and Philipp Slusallek. 2012b. Importance Caching for Complex Illumination. *Computer Graphics Forum* 31, 2 (2012), 701–710.

Pascal Grittmann, Iliyan Georgiev, Philipp Slusallek, and Jaroslav Křivánek. 2019. Variance-Aware Multiple Importance Sampling. *ACM Transactions on Graphics* 38, 6 (2019), 9.

Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. 2008. Progressive Photon Mapping. *ACM Transactions on Graphics* 27, 5 (2008), 130:1–130:8.

Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. 2012. A Path Space Extension for Robust Light Transport Simulation. *ACM Transactions on Graphics* 31, 6, Article 191 (2012), 10 pages.

Miloš Hašan, Fabio Pellacini, and Kavita Bala. 2007. Matrix Row-column Sampling for the Many-light Problem. *ACM Trans. Graph.* 26, 3 (2007), 26:1–26:10.

Sebastian Herholz, Oskar Elek, Jiří Vorba, Hendrik Lensch, and Jaroslav Křivánek. 2016. Product Importance Sampling for Light Transport Path Guiding. *Computer Graphics Forum* 35, 4 (2016), 67–77.

Ondřej Karlík, Martin Šik, Petr Vévoda, Tomáš Skřivan, and Jaroslav Křivánek. 2019. MIS Compensation: Optimizing Sampling Techniques in Multiple Importance Sampling. *ACM Transactions on Graphics* 38, 6 (2019), 12.

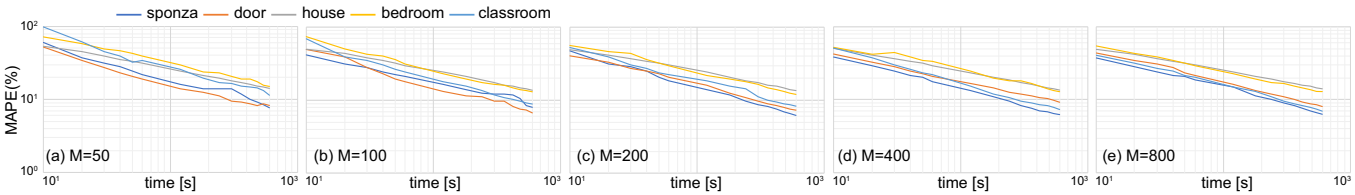


Fig. 7. Convergence graphs of all scenes for $M = 50, 100, 200, 400$ and 800. The use of small M does not provide stable noise reduction and undulation is pronounced in (a) $M = 50$ and (b) $M = 100$. Larger M provides stable noise reduction and the log-log plots approach the straight lines as shown in (d) $M = 400$ and (e) $M = 800$.

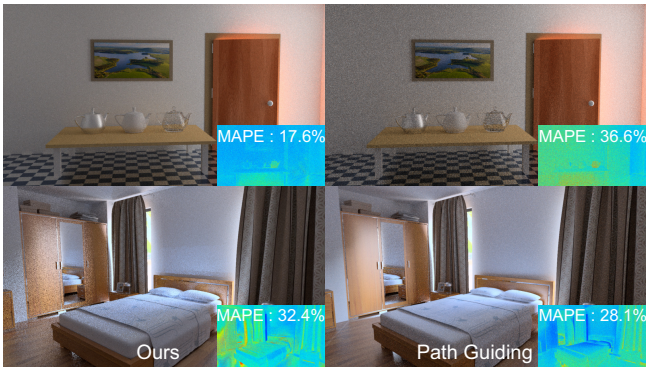


Fig. 8. Equal-time comparison between our method without path guiding (left) and BPT with path guiding (right) [Vorba et al. 2014].

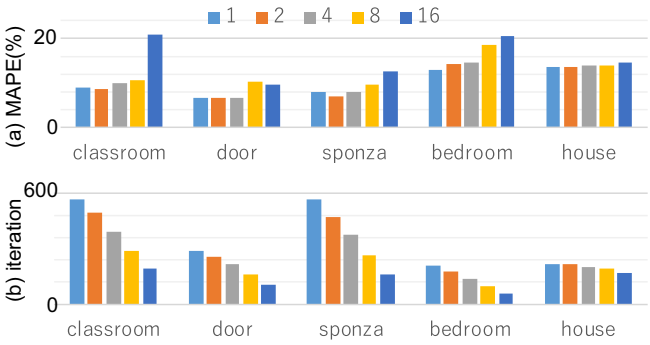


Fig. 9. MAPEs (top) and iteration counts (bottom) of equal-time rendering (10 min) for various N (the number of light sub-path samples at each iteration).

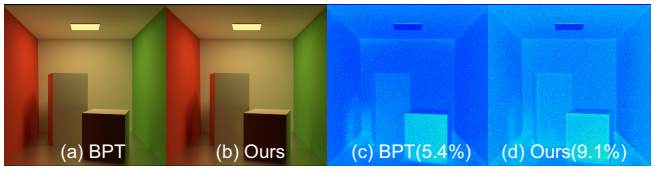


Fig. 10. Failure case. Equal-time comparison (1 min) between (a) BPT and (b) our method for a Cornell-box scene. The figures in (c) and (d) represent MAPE for BPT and our method, respectively. For such a simple scene, the overhead of our method decreases the possible iteration counts (155 iterations in BPT while only 38 iterations in our method) and BPT outperforms our method.

Alexander Keller. 1997. Instant Radiosity. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. 49–56.
Thomas Kollig and Alexander Keller. 2004. Illumination in the Presence of Weak Singularities. In *Monte Carlo and Quasi-Monte Carlo Methods*. 245–257.
Ivo Kondapaneni, Petr Vévoda, Pascal Grittmann, Tomáš Skřivan, Philipp Slusallek, and Jaroslav Křivánek. 2019. Optimal Multiple Importance Sampling. *ACM Transactions on Graphics* 38, 4 (2019), 37:1–37:14.
Eric P. Lafortune and Yves D. Willems. 1993. Bi-Directional Path Tracing. In *Computer-graphics '93*. 145–153.
Thomas Müller, Markus Gross, and Jan Novák. 2017. Practical Path Guiding for Efficient Light-Transport Simulation. *Computer Graphics Forum* 36, 4 (2017), 91–100.
Stefan Popov, Ravi Ramamoorthi, Fredo Durand, and George Drettakis. 2015. Probabilistic Connections for Bidirectional Path Tracing. *Computer Graphics Forum* 34, 4 (2015), 75–86.
Justin F. Talbot. 2005. *Importance Resampling for Global Illumination*. Master's thesis. Bringham Young University.
Justin F. Talbot, David Cline, and Parris Egbert. 2005. Importance Resampling for Global Illumination. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques (EGSR '05)*. 139–146.
Eric Veach. 1997. *Robust Monte Carlo Methods for Light Transport Simulation*. Ph.D. Dissertation. Stanford University.
Eric Veach and Leonidas Guibas. 1994. Bidirectional Estimators for Light Transport. 145–167.
Eric Veach and Leonidas J. Guibas. 1995. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. 419–428.
Petr Vévoda, Ivo Kondapaneni, and Jaroslav Křivánek. 2018. Bayesian Online Regression for Adaptive Direct Illumination Sampling. *ACM Transactions on Graphics* 37, 4 (2018), 125:1–125:12.
Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. 2014. On-line Learning of Parametric Mixture Models for Light Transport Simulation. *ACM Transactions on Graphics* 33, 4 (2014), 101:1–101:11.
Bruce Walter, Adam Arbree, Kavita Bala, and Donald P. Greenberg. 2006. Multidimensional Lightcuts. *ACM Transactions on Graphics* 25, 3 (2006), 1081–1088.
Bruce Walter, Sebastian Fernandez, Adam Arbree, Kavita Bala, Michael Donikian, and Donald P. Greenberg. 2005. Lightcuts: A Scalable Approach to Illumination. *ACM Transactions on Graphics* 24, 3 (2005), 1098–1107.
Bruce Walter, Pramook Khungurn, and Kavita Bala. 2012. Bidirectional Lightcuts. *ACM Transactions on Graphics* 31, 4 (2012), 59:1–59:11.

ACKNOWLEDGMENTS

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