

# SPCBPT: Subspace-based Probabilistic Connections for Bidirectional Path Tracing

FUJIA SU, Peking University, China

SHENG LI\*, School of Computer Science, Peking University, China

GUOPING WANG, School of Computer Science, Peking University, China

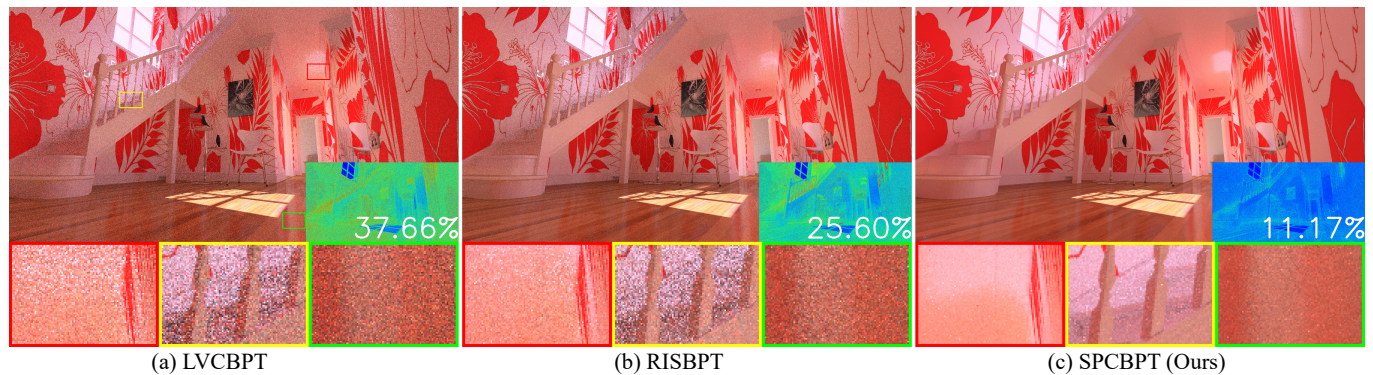


Fig. 1. Equal-time (3 mins) comparisons of the Hallway scene between Light Vertex Cache BDPT (LVCBPT) [Davidović et al. 2014], Resampled Importance Sampling BDPT (RISBPT) [Nabata et al. 2020a] and our approach (SPCBPT). Our approach can significantly improve the rendering quality in terms of mean absolute percentage error (MAPE). We also highlight some local regions in a zoom-in view.

Bidirectional path tracing (BDPT) can be accelerated by selecting appropriate light sub-paths for connection. However, existing algorithms need to perform frequent distribution reconstruction and have expensive overhead. We present a novel approach, SPCBPT, for probabilistic connections that constructs the light selection distribution in sub-path space. Our approach bins the sub-paths into multiple subspaces and keeps the sub-paths in the same subspace of low discrepancy, wherein the light sub-paths can be selected by a subspace-based two-stage sampling method, i.e., first sampling the light subspace and then resampling the light sub-paths within this subspace. The subspace-based distribution is free of reconstruction and provides efficient light selection at a very low cost. We also propose a method that considers the Multiple Importance Sampling (MIS) term in the light selection and thus obtain an MIS-aware distribution that can minimize the upper bound of variance of the combined estimator. Prior methods typically omit this MIS weights term. We evaluate our algorithm using various benchmarks, and the experimental results show that our approach has superior performance and can significantly reduce the noise compared with the state-of-the-art method.

\*corresponding author

Authors' addresses: Fujia Su, sufujia@pku.edu.cn, Peking University, China; Sheng Li, lisheng@pku.edu.cn, School of Computer Science, Peking University, China; Guoping Wang, wgp@pku.edu.cn, School of Computer Science, Peking University, China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2022 Association for Computing Machinery.

0730-0301/2022/7-ART77 \$15.00

<https://doi.org/10.1145/3528223.3530183>

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

Additional Key Words and Phrases: Bidirectional path tracing, subspace, weight, light selection, multiple importance sampling, sub-path, resampling

## ACM Reference Format:

Fujia Su, Sheng Li, and Guoping Wang. 2022. SPCBPT: Subspace-based Probabilistic Connections for Bidirectional Path Tracing. *ACM Trans. Graph.* 41, 4, Article 77 (July 2022), 14 pages. <https://doi.org/10.1145/3528223.3530183>

## 1 INTRODUCTION

Bidirectional path tracing is a classic solution for the light transport simulation [Lafortune and Willems 1993]. It is widely used due to its excellent performance and robustness under various complex lighting conditions [Veach and Guibas 1995a]. BDPT first traces sub-path pairs (eye sub-path and light sub-path) from a light source and a camera, respectively; it then constructs full paths by connecting vertices on the eye sub-paths with those on light sub-paths. Multiple importance sampling (MIS) is used to combine different sampling strategies by weighting samples according to the probability density of each sample [Veach and Guibas 1995b]. Higher weights are assigned to samples of higher probability density to reduce variance.

In the classic BDPT, the renderer generates (or selects) a light sub-path for each traced eye sub-path and then connects these two sub-paths to construct a full path. This light selection problem cannot be handled well by the classic BDPT because the connection is poorly importance sampled, which results in very low efficiency, especially in scenes with complex visibility.

Probabilistic connections for BDPT (PCBPT) [Popov et al. 2015] made an improvement by reusing multiple light sub-paths between different eye sub-paths and selecting the appropriate light sub-path

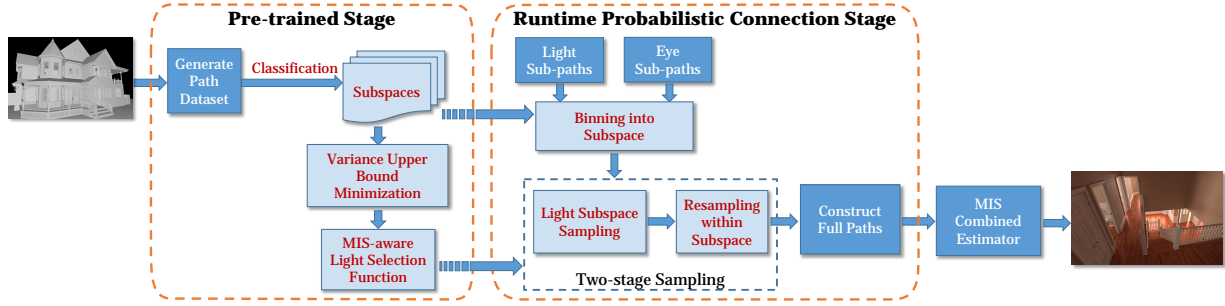


Fig. 2. Overview pipeline of our approach. Our approach finds an approximate solution to minimize the variance upper bound within the classified subspaces in the pre-trained stage, and provides an MIS-aware light selection function for the two-stage sampling in the runtime probabilistic connection stage. Our approach efficiently constructs connections in subspace and present an MIS combined estimator.

regarding the constructed path contribution. Nabata et al. [2020a] proposed resampled importance sampling BDPT (RISBPT) by applying a resampling-aware MIS function to PCBPT and achieved significant noise reduction in various difficult scenarios. However, the candidate light sub-paths used for connection need frequent updates to achieve unbiasedness. As a result, PCBPT and RISBPT need millions of shadow rays to reconstruct the resampling probability mass function (PMF) stored in the cache points when updating the light sub-paths. This expensive overhead limits the number of candidate light sub-paths available and introduces visual artifacts in the early rendering stage. Furthermore, the sampling probabilities ignore multiple importance sampling weights, making PCBPT and RISBPT estimators difficult to achieve optimal.

In this paper, we present an approach to efficiently solve the light selection problem and eliminate the overhead of frequent PMF reconstruction. Different from the previous works that compute the PMF in a subpath-to-subpath manner, we present a novel sampling solution in the sub-path space. By grouping similar eye sub-paths and light sub-paths, we obtain a set of *subspaces*. Here, subspace is defined as a subset of the sub-path space, and sub-paths in the same subspace should be of low discrepancy and share sampling importance. We generate the light sub-paths by first sampling the light subspace and then sampling the light sub-paths within the subspace, called two-stage sampling. Subspace sampling importance is consistent in the rendering and facilitates optimizing the estimator. We collect statistics in a preprocessing pass to determine the sampling importance in subspace. The optimal subspace sampling probability needs to consider multiple importance sampling. However, multiple importance sampling weights depend on the sampling probability. We solve this problem and show that the optimization of sampling importance that minimizes an upper bound of variance is convex, and then use stochastic gradient descent (SGD) to solve for the optimized light selection distribution. Therefore, we obtain an optimized MIS combined estimator based on the proposed MIS-aware light selection, which facilitates efficient sub-path sampling.

We illustrate the overview pipeline of our method in Figure 2. We highlight the novel modules in our algorithm based on the BDPT framework. The subspaces are first specified according to a classification function in the pre-trained stage (preprocessing). We then determine the MIS-aware light selection function by minimizing

the variance upper bound. The selection function is used in runtime sub-paths sampling. In the runtime rendering stage, both light sub-paths and eye sub-paths are binned into the subspaces, and then the two-stage sampling method is used to sample the subspace and sub-path for probabilistic connections. Finally, the full paths are constructed for an MIS combined estimator.

## 2 RELATED WORK

The key of BDPT is to establish the connections between the eye vertices (i.e., eye sub-path) and the light vertices (i.e., light sub-path) [Pajot et al. 2011; Popov et al. 2015; Walter et al. 2012]. Davidović et al. proposed light vertex cache BDPT (LVCBPT), which uses light vertex cache to store and resample light sub-paths.

Importance sampling between bidirectional probabilistic path connections is a key factor affecting efficiency [Georgiev et al. 2012b]. Popov et al. [2015] proposed PCBPT to reuse multiple light sub-paths. It samples the connections between the eye sub-path and a few light sub-paths regarding full path contribution. PCBPT’s importance sampling is a resampling method [Talbot et al. 2005]. After that, Nabata et al. [2020a] proposed RISBPT, which provides MIS weighting functions for PCBPT. In addition to PCBPT, Tokuyoshi and Harada [2019] proposed a hierarchical Russian roulette (HRR) scheme that use BVH to construct connections between eye vertex and light vertices within the BRDF scattering range. Matrix BDPT reorders sub-paths and builds a matrix from light sub-paths and eye sub-paths to improve connection sampling [Chaitanya et al. 2018]. Bitterli et al. [2020] used resampling of direct lighting and achieves significant improvement in real-time.

Many-light methods [Keller 1997; Krivanek et al. 2014] also study the connections between eye sub-path and multiple light sub-paths. Many algorithms select a small portion of the light vertices to connect to improve efficiency [Georgiev et al. 2012b; Hašan and Pellacini 2007; Walter et al. 2006, 2005]. Walter et al. [2012] proposed bidirectional lightcut and introduced MIS into the many-light method. Previous works also find that light selection efficiency can be improved by clustering light sources [Ou and Pellacini 2011; Wang et al. 2021; Wu and Chuang 2013]. Compared to the many-light method, our algorithm focuses on those BDPT scenarios that require special handling of clustering and multiple importance sampling.



Two-stage RISBPT [Nabata et al. 2020b] combines RISBPT with lightcuts and uses lightcuts to sample candidate sub-paths that are likely to contribute highly to the current eye sub-path. However, visibility is ignored when candidates are sampled from the light tree, and the reconstruction of light tree can be expensive. Furthermore, the target probability density function (PDF) of the two-stage algorithm omits the MIS weight, which makes it difficult to obtain a sampling distribution that minimizes the global variance.

Path guiding methods can importance sample paths adaptively according to the scene's illumination distribution [Jensen 1996; Lafortune and Willems 1999]. Vorba et al. [2014] proposed a Gaussian mixture model for path guiding. Müller et al. [2017] proposed an adaptive spatio-directional tree (SD-tree) to represent the incident radiance field based on the spatial binary tree structure of the scene. Other extensions of path guiding include product path guiding [Herholz et al. 2016], using linearly transformed cosines [Diolatzis et al. 2020], using neural networks [Müller et al. 2019, 2020], or guiding the sampling direction in the primary space [Guo et al. 2018]. Variance-aware path guiding [Rath et al. 2020] guides the tracing process in a distribution that can minimize variance. This inspires us to deal with the MIS-aware distribution. MCMC methods importance sample a new path by perturbing the existing one for scenes with highly complex visibility [Jakob and Marschner 2012; Veach and Guibas 1997]. It can also be applied to tracing valuable light sub-paths, but the sampled path may be stuck in a local path space.

### 3 PRELIMINARIES

#### 3.1 Resampled Importance Sampling

Resampled Importance Sampling (RIS) is a sampling technique that can generate samples approximately proportional to any target distribution [Talbot et al. 2005]. To generate samples from target distribution  $q$ , RIS first generates  $M$  candidate samples from a PDF  $p$ , which is easy to sample and can be arbitrarily specified with a constraint that  $p > 0$  whenever  $q > 0$ . Then samples are selected from the candidates in probability proportional to  $q/p$ . If the size of candidates  $M$  is sufficiently large, RIS can generate samples that approximate the target distribution [Nabata et al. 2020a].

#### 3.2 Bidirectional Path Tracing (BDPT)

From the path integral formulation [Veach and Guibas 1995a], the pixel measurement  $I$  is as:

$$I = \int_{\Omega} f(\bar{x}) d\mu(\bar{x}), \quad (1)$$

where  $\Omega$  is the path space, and  $\bar{x} = x_0 x_1 \dots x_k$  is a path of length  $k \geq 1$ ,  $x_0$  and  $x_k$  are on a light source and the camera, respectively.  $d\mu(\bar{x}) = dA(x_0) \dots dA(x_k)$  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  $L_e(x_0, x_1)$  is the radiance emitted  $x_0 \rightarrow x_1$ ,  $W_e$  is the pixel sensitivity,  $\rho$  is the bidirectional scattering distribution function (BSDF), and  $GV$  is the geometry term including the visibility.

Equation 1 is estimated using Monte Carlo integration. Specifically, a full path  $\bar{x}$  is constructed by a light sub-path  $\bar{y}$  and an eye sub-path  $\bar{z}$  as  $\bar{x} = \bar{y}\bar{z}$ . There are  $k+2$  strategies to generate a path of length  $k$ ; each constructs the same full path  $\bar{x}$  by sub-path pair of different lengths. Multiple Importance Sampling (MIS) [Veach and Guibas 1995b] is employed to combine samples from different sampling strategies. As  $n$  strategies are used to generate samples, the combined estimator with MIS is as

$$\tilde{I} = \sum_t \frac{1}{n_t} \sum_i^{n_t} w_t(\bar{X}_{t,i}) \frac{f(\bar{X}_{t,i})}{p_t(\bar{X}_{t,i})}, \quad (2)$$

where  $w_t$  is the weighting function,  $n_t$  the number of samples,  $p_t$  the PDF, and  $\bar{X}_{t,i}$  the  $i$ -th path sample of the  $t$ -th sampling strategy.

Weighting function  $w$  weighs different strategies for each full path  $\bar{x}$ . For unbiased estimation, the sum of the weighting function for each path  $\bar{x}$  should be equal to one (i.e.,  $\sum_t w_t(\bar{x}) = 1$ ) and the weighting function should return zero whenever the PDF returns zero. With these constraints, the weighting function can be designed freely to reduce variance. The balance heuristic aims to minimize the upper bound of variance by assigning each strategy a weight proportional to the sampling density as:

$$w_t(\bar{x}) = \frac{n_t p_t(\bar{x})}{\sum_i n_i p_i(\bar{x})}. \quad (3)$$

#### 3.3 Light Sub-paths Reuse and Probabilistic Connections

BDPT traces an eye sub-path from the camera and a light sub-path from the light source for each pixel. LVCBPT extends BDPT by reusing multiple light sub-paths in different eye sub-paths.

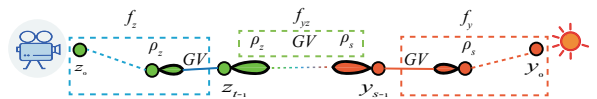
LVCBPT estimates the pixel intensity iteratively. In each rendering iteration, LVCBPT traces multiple light sub-paths in a single pass and stores the light vertices (i.e., the prefixes of the light sub-paths) in the cache, called Light Vertex Cache (LVC). Each prefix sub-path in the LVC can be treated as an individual sub-path. Then, the light sub-path is randomly selected from the LVC to connect with each eye sub-path and eventually constructs the full path.

Sampling the light sub-path from LVC can be interpreted as sampling in the sub-path space in a resampling way, setting the PDF of light sub-path tracing  $p(\bar{y})$  to the target distribution. PCBPT importance samples the cached light sub-paths and decomposes the contribution  $f$  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 [Popov et al. 2015]:

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

$$f_{yz}(\bar{y}, \bar{z}) = \rho(y_{s-2}, y_{s-1}, z_{t-1}) GV(y_{s-1}, z_{t-1}) \rho(y_{s-1}, z_{t-1}, z_{t-2}),$$

$$f_z(\bar{z}) = W_e(z_0, z_1) GV(z_0, z_1) \prod_{i=1}^{t-2} \rho(z_{i-1}, z_i, z_{i+1}) GV(z_i, z_{i+1}).$$



Since  $f_z$  is shared between connections and therefore can be ignored, PCBPT sets  $p(\bar{y}|\bar{z}) = f_{yz}(\bar{y}, \bar{z})f_y(\bar{y})$  as the target distribution for light sub-path resampling and constructs the distributions for resampling in the PMF records, which are distributed across the scene. Nearby eye sub-paths can share a PMF record and its distribution, which avoids repeated construction of the PMF for each eye sub-path. The MIS weighting function of PCBPT can not handle PDF variations due to resampling. Nabata et al. [2020a] proposed RISBPT to address this problem by using resample-aware weighting functions, which significantly improves rendering quality.

However, RISBPT does not involve MIS in the target distribution. It is difficult to resolve an MIS-aware sampling distribution because MIS also needs sampling distribution to determine the weights. Therefore, RISBPT estimator can be further optimized. Moreover, RISBPT needs an extra visibility test to construct PMF records, and this computational overhead will decrease the iteration counts possible in a given time budget [Nabata et al. 2020a] and limit the number of light sub-paths  $M$  used for resampling. The performance of RIS depends heavily on the candidate number  $M$ . In scenes with difficult visibility, the limited  $M$  of RISBPT is insufficient to provide sampling close to the target distribution and is less efficient.

#### 4 OVERVIEW

Our algorithm follows the same probabilistic connections scheme as LVCBPT and PCBPT. So, our goal is to resample the candidate light sub-paths and select the appropriate light sub-path for each eye sub-path to construct connections efficiently. The target PDF for resampling should be easy to evaluate for rendering efficiency, and the resampling distribution should avoid reconstruction as much as possible. Therefore, the key to our approach is to classify the sub-paths into multiple subspaces, thereby keeping the sub-paths in the same subspace of low discrepancy. Then, we can determine the optimal subspaces connection probability by taking multiple importance sampling into account.

We organize the exposition of our approach as follows:

- We first set up the theoretical basis of subspace-based sampling and probabilistic connections (Sec. 5). We start from the optimal distribution of probabilistic connections, and derive the formulation of our subspace-based light selection function, which can be used to approximate the optimal distribution.
- Next, the key issue is to find an effective solution to practical sub-path sampling (Sec. 6). By setting the approximate distribution as the target PDF, we present a subspace-based two-stage sampling method along with an estimator (Sec. 6.1). MIS requires the sampling density of each sampling strategy to compute the MIS weight, and the sampling density depends on the distribution of light selection. Therefore, we propose our MIS weighting function considering subspace sampling (Sec. 6.2). We also involve the MIS weighting function to make the light selection optimal in the subspace. We prove that the minimization of the upper bound of estimated variance is convex regarding subspace sampling distribution, which leads to an MIS-aware sampling distribution and can be resolved by stochastic gradient descent (SGD) (Sec. 6.3). This can also

help resolve the circular dependency problem between MIS weighting function and sampling distribution. Classification of the subspace affects the performance of subspace-based sampling. Therefore, we present a method to divide the sub-path space into subspaces along with a classification function, making sub-paths within a subspace with low discrepancy and the subspace can be indexed quickly (Sec. 6.4).

- Lastly, some technical details for subspace sampling are present in Sec. 7. We run a preprocessing pass and trace a path dataset  $\mathcal{D}$  (Sec. 7.1). Based on this dataset, we classify the subspaces and build decision trees used for fast subspace binning (Sec. 7.2). The subspace-based MIS-aware sampling distribution can be obtained by minimizing the upper bound on the variance of paths in this dataset (Sec. 7.3). In addition, we employ cross-iteration light sub-path reuse to further improve the efficiency (Sec. 7.4).

### 5 SUBSPACE-BASED PROBABILISTIC CONNECTIONS: PRINCIPLE

In this section, we propose our subspace-based light selection function, which is the foundation for our sampling method.

#### 5.1 Optimal Distribution of Probabilistic Connections

We first consider the basic connection principles in probabilistic connection algorithms such as LVCBPT and PCBPT. Eye sub-paths are traced from the camera and connected with light sub-paths resampled from the candidate light sub-paths, and the weighted contribution of each full path is estimated. The sampling strategy is identified by the number of vertices of the eye sub-path; strategy  $t$  refers to generating full paths by connecting eye sub-paths of  $t$  vertices with light sub-paths of arbitrary length. Therefore, the pixel measurement  $I$  in Equation 1 can be reformulated as:

$$I = \sum_t I_t = \sum_t \int_{A^t} I(\bar{z}) d\mu(\bar{z}), \quad (4)$$

$$I(\bar{z}) = \int_{\mathcal{S}} w_t(\bar{y}\bar{z}) f(\bar{y}\bar{z}) d\mu(\bar{y}), \quad (5)$$

where  $A^t$  is the  $t$ -dimensional Cartesian product over the scene surface  $A$ , and  $\mathcal{S} = \cup_t A^t$  is the space of sub-path.  $t$  denotes the number of vertices of eye sub-path  $\bar{z}$  and is used to identify the sampling strategy.

Monte Carlo estimation is used to evaluate  $I(\bar{z})$  for eye sub-path  $\bar{z}$  via connections, and estimation of  $I(\bar{z})$  can be improved by generating the light sub-path  $\bar{y}$  regarding the eye sub-path  $\bar{z}$ . The optimal distribution  $p^*(\bar{y}|\bar{z})$  to estimate  $I(\bar{z})$  should satisfy

$$p^*(\bar{y}|\bar{z}) \propto w_t(\bar{y}\bar{z}) f(\bar{y}\bar{z}) = w_t(\bar{y}\bar{z}) f_y(\bar{y}) f_{yz}(\bar{y}, \bar{z}) f_z(\bar{z}),$$

where  $f_z(\bar{z})$  is ignorable since it is independent of  $\bar{y}$ , therefore,

$$p^*(\bar{y}|\bar{z}) = \gamma(\bar{y}, \bar{z}) \propto w_t(\bar{y}\bar{z}) f_y(\bar{y}) f_{yz}(\bar{y}, \bar{z}). \quad (6)$$

Here we define the optimal light selection function  $\gamma(\bar{y}, \bar{z})$ , a  $\mathcal{S} \times \mathcal{S} \rightarrow \mathcal{R}$  function that returns the optimal sampling density for a given  $(\bar{y}, \bar{z})$  pair. For any given  $\bar{z}$ , we marginalize  $\bar{z}$  in function  $\gamma$  to get an optimal distribution of  $\bar{y}$ .

$I(\bar{z})$  can be estimated efficiently when  $\gamma$  and its marginal distribution are easy to implement. However, complicated scene settings

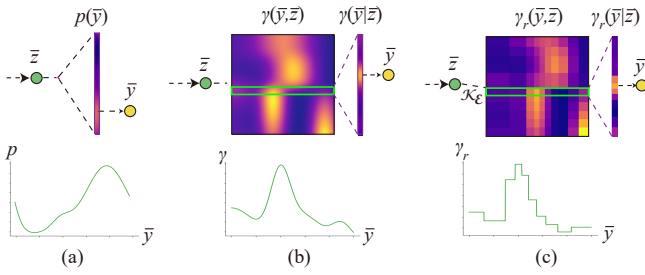


Fig. 3. (a) BDPT generates the light sub-path in a distribution irrelevant to eye sub-path, which may be far from optimal. (b) Optimal light selection distribution can be constructed from  $\gamma$  regarding eye sub-path  $\bar{z}$ . (c) Our approach discretize  $\gamma$  for each subspace-subspace pair to sample light sub-path in a distribution close to the optimal one.

with difficult visibility usually make direct evaluation of  $\gamma$  expensive. Instead, we seek an alternative way to approximate the function  $\gamma$  while maintaining the convenience of evaluation and sampling.

## 5.2 Approximate Formulation of $\gamma$

$\gamma$  is a continuous function defined in  $\mathcal{S} \times \mathcal{S}$  and of much complexity that is difficult to be described by simple continuous functions. We, instead, approximate  $\gamma$  discretely and introduce subspace to get a piecewise representation of  $\mathcal{S}$ . A subspace is a subset of the sub-path space  $\mathcal{S}$ . By dividing  $\mathcal{S}$  into subspaces, we approximately evaluate  $\gamma$  in each subspace-subspace pair separately.

Formally, we introduce the sub-path classification function  $\kappa : \mathcal{S} \rightarrow \mathcal{Z}$ .  $\kappa$  function defines the division of sub-path space  $\mathcal{S}$ , and can be used to bin a sub-path into its associated subspace, and returns with an ID of that subspace. Considering that the distributions of light sub-path and eye sub-path are usually different, we apply different classification functions to the light sub-path (denoted as  $\kappa_{\mathcal{L}}$ ) and the eye sub-path (denoted as  $\kappa_{\mathcal{E}}$ ), respectively. Then we approximate  $\gamma$  by  $\gamma_r$  in a formulation as:

$$\gamma(\bar{y}, \bar{z}) \approx \gamma_r(\bar{y}, \bar{z}) = \Gamma(\kappa_{\mathcal{L}}(\bar{y}), \kappa_{\mathcal{E}}(\bar{z})) f_y(\bar{y}) / Q(\kappa_{\mathcal{L}}(\bar{y})), \quad (7)$$

where  $\Gamma$  is a left stochastic matrix that satisfies  $\sum_i \Gamma(i, j) = 1$  for any  $j \geq 0$ , i. e., the marginal distribution  $\Gamma(i|j)$  is a PMF. We retain the  $f_y(\bar{y})$  term since  $f_y(\bar{y})$  is irrelevant to the eye sub-path  $\bar{z}$  and can be obtained easily during path tracing.  $Q$  is a normalization factor, and  $Q(i)$  is an integral of  $f_y(\bar{y})$  for all  $\bar{y}$  in the light subspace  $i$  as:

$$Q(i) = \int_{\mathcal{S}} f_y(\bar{y}) (\kappa_{\mathcal{L}}(\bar{y}) = i) d\mu(\bar{y}). \quad (8)$$

$Q$  is introduced to ensure that  $\int_{\mathcal{S}} \gamma_r(\bar{y}, \bar{z}) d\mu(\bar{y}) = 1$  holds for every  $\bar{z}$ , i. e.,  $\gamma_r(\bar{y}|\bar{z})$  is a probability density function.

Figure 3 illustrates the key idea of discretization and approximation of the optimal light selection function  $\gamma$ . By dividing  $\mathcal{S}$  into subspaces, we can sample the light sub-path in a distribution close to the optimal one.

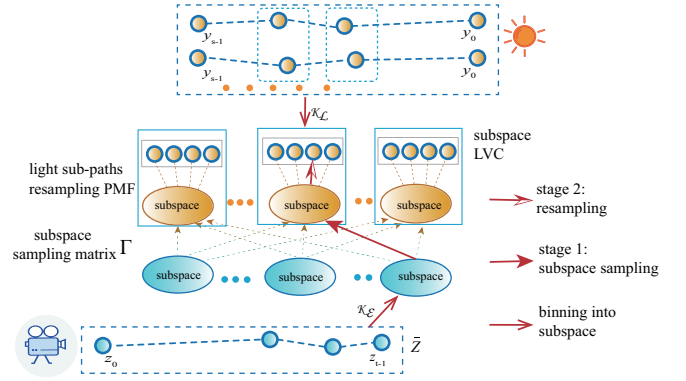


Fig. 4. Two-stage sampling. For an eye sub-path  $\bar{Z}$ , we bin it into its associated subspace by  $\kappa_{\mathcal{E}}$ , use matrix  $\Gamma$  to sample the light subspace regarding the eye subspace, and resample the candidate sub-path within the subspace LVC constructed by  $\kappa_{\mathcal{L}}$ .

## 6 SUBSPACE-BASED PROBABILISTIC CONNECTIONS: METHOD

In this section, we first introduce a two-stage sampling method that efficiently selects the light sub-path in  $\gamma_r$  for our estimator (Sec. 6.1); next, we propose our MIS weighting function (Sec. 6.2) and a method to optimize matrix  $\Gamma$  that can minimize the upper bound of variance (Sec. 6.3); finally, we propose a method to divide space  $\mathcal{S}$  into subspaces and construct  $\kappa$  to keep sub-paths in a subspace of low discrepancy (Sec. 6.4).

### 6.1 Two-stage Sampling and Estimator for $\gamma_r$

To sample from the sub-path space  $\mathcal{S}$  with an approximate function  $\gamma_r$ , we employ a resampling method similar to LVCBPT. In each rendering iteration, we trace  $M$  sub-paths from the light source in the PDF  $p(\bar{y})$ , treat each prefix of the sub-paths as an individual candidate sub-path  $\bar{Y}$ , and construct the LVC. Unlike LVCBPT, we construct an LVC for each light subspace individually rather than a whole for the entire path space  $\mathcal{S}$ . Each light sub-path  $\bar{Y}$  is assigned to the LVC for subspace  $\kappa_{\mathcal{L}}(\bar{Y})$ .

For each eye sub-path  $\bar{Z}$  traced from the camera, we resample the light sub-path  $\bar{Y}$  in two steps:

**Stage 1:** for a given eye sub-path  $\bar{Z}$ , we sample the marginalized PMF  $\Gamma(\kappa_{\mathcal{L}}(\bar{Y})|\kappa_{\mathcal{E}}(\bar{Z}))$  to get the light subspace  $\kappa_{\mathcal{L}}(\bar{Y})$  for the second-stage sampling.

**Stage 2:** for the light subspace  $\kappa_{\mathcal{L}}(\bar{Y})$ , we resample  $\bar{Y}$  in the light subspace with a target PDF proportional to  $f_y$ .

The resampling PMF  $P_r(\bar{Y})$  for the second-stage sampling is constructed as

$$P_r(\bar{Y}) = \frac{f_y(\bar{Y})/p(\bar{Y})}{\sum_{\bar{Y}' \in \text{LVC}(\bar{Y})} f_y(\bar{Y}')/p(\bar{Y}')}. \quad (9)$$

The normalization factor  $Q$  of  $\gamma_r$  is shared in the light subspace and canceled out in the second stage. Resampling does not require normalization of the target distribution [Nabata et al. 2020a], but  $Q$  is necessary for the MIS computation in the next section.



Figure 4 shows the procedure of two-stage sampling. For each eye sub-path  $\bar{Z}$ , we find its associated subspace  $\kappa_{\mathcal{E}}(\bar{Z})$  by the binning process and select the light subspace according to  $\Gamma$ . Finally we resample the light sub-path within the subspace  $\kappa_{\mathcal{L}}(\bar{Y})$ . Considering that  $f_y(\bar{Y})/p(\bar{Y})$  is uncorrelated with the eye sub-path  $\bar{Z}$  in the second-stage sampling, we construct the sampling PMF only once in each iteration without extra visibility test. Therefore, only a little computational overhead is necessary.

Finally, we provide the estimator based on our sampling method. For an eye sub-path  $\bar{Z}$  of  $t$  vertices traced from the camera with probability density  $p(\bar{Z})$ ,  $I_t$  is estimated by  $I(\bar{Z})/p(\bar{Z})$  (Equation 4). The average weighted contribution of the paths constructed by the eye sub-path  $\bar{Z}$  and prefixes of  $M$  light sub-paths makes an unbiased estimation of  $I(\bar{Z})$  (Equation 5). We sample  $N$  light sub-paths  $\{\bar{Y}_i\}$  to connect with  $\bar{Z}$ . For the  $i$ -th light sub-path  $\bar{Y}_i$ , the resampling probability is that the probability for the first stage to sample the light subspace as  $\Gamma(\kappa_{\mathcal{L}}(\bar{Y}_i), \kappa_{\mathcal{E}}(\bar{Z}))$  times the probability for the second stage to sample  $\bar{Y}_i$  as  $P_r(\bar{Y}_i)$ . So the estimator  $\tilde{I}_t$  for strategy  $t$  based on the two-stage sampling is as:

$$\tilde{I}_t = \frac{1}{MN} \sum_{i=1}^N \frac{w_t(\bar{Y}_i, \bar{Z}) f(\bar{Y}_i, \bar{Z})}{p(\bar{Z}) \Gamma(\kappa_{\mathcal{L}}(\bar{Y}_i), \kappa_{\mathcal{E}}(\bar{Z})) P_r(\bar{Y}_i)}, \quad (10)$$

and the combined estimator  $\tilde{I}$  is defined as  $\tilde{I} = \sum_t \tilde{I}_t$  (Equation 4).

## 6.2 MIS Function for Subspace-Based Connection

Probabilistic connections will change the sampling PDF of full path  $\bar{x}$ . We, therefore, derive the MIS weighting function for our subspace-based probabilistic connections.

In our method, the selection of light sub-path  $\bar{y}$  is directed by function  $\gamma_r(\bar{y}, \bar{z})$ , so the target PDF  $p_t(\bar{x})$  for strategy  $t$  is

$$p_t(\bar{x}) = p(\bar{z}) \gamma_r(\bar{y}, \bar{z}), \quad (11)$$

where full path  $\bar{x}$  is constructed by connecting eye sub-path  $\bar{z}$  of  $t$  vertices with light sub-path  $\bar{y}$  as  $\bar{x} = \bar{y}\bar{z}$ .

For a full path of length  $k$ , no connection is involved in strategies  $t \in \{0, k+1\}$ , and classic BDPT can efficiently handle strategy  $t = 1$ . Therefore,  $p_t(\bar{x})$  for strategies  $t \in \{0, 1, k+1\}$  can be resolved in the same way as the classic BDPT. Our method works for strategies  $t \in \{2, 3, \dots, k\}$  only, and the  $p_t(\bar{x})$  of these strategies should be resolved by Equation 11.

We apply the balance heuristic to the weighting function as:

$$w_t(\bar{x}) = \frac{N_t p_t(\bar{x})}{\sum_{i=0}^{k+1} N_i p_i(\bar{x})}, \quad (12)$$

where  $N_t$  is the sample count for strategy  $t$ . We set  $N_t = N$  for strategies  $t \in \{2, 3, \dots, k\}$ , where  $N$  is the number of connections for each eye sub-path, as referred to in Equation 10. Other strategies adopt the same settings as classic BDPT. More discussion about the selection of  $N_t$  to reduce variance of path correlation can be found in Grittmann et al. [2021]; Popov et al. [2015].

Note that the  $p_t(\bar{x})$  in Equation 11 is the target PDF rather than the actual PDF of strategy  $t$  for resampling. The actual PDF for resampling is unavailable in most cases, but a converged target PDF can be achieved as the number of candidate samples  $M$  increases [Talbot et al. 2005]. As long as  $\sum_t w_t(\bar{x}) = 1$  is satisfied,

using an approximate probability density to compute MIS weighting function will not affect the unbiasedness of the estimation, as discussed in Veach and Guibas [1995b].

## 6.3 MIS-Aware $\Gamma$ Minimizing Upper Bound of Variance

Based on the MIS weighting function  $w_t(\bar{y}\bar{z})$  defined in Equation 12, we next evaluate matrix  $\Gamma$  in function  $\gamma_r$ .

Traditionally, MIS is ignored in the light selection distribution, so  $\Gamma$  can be evaluated in terms of full path contribution between subspaces as:

$$\Gamma_{i,j}^{full} \propto \int_{\mathcal{S}_{\mathcal{L}(i)}} \int_{\mathcal{S}_{\mathcal{E}(j)}} f(\bar{y}\bar{z}) d\mu(\bar{z}) d\mu(\bar{y}), \quad (13)$$

where  $\mathcal{S}_{\mathcal{L}(i)}$  and  $\mathcal{S}_{\mathcal{E}(j)}$  refer to the light subspace  $i$  and eye subspace  $j$ , respectively.

$\Gamma^{full}$  in Equation 13 is not optimized due to the absence of the MIS term. A straightforward way to recover the MIS term and obtain an MIS-aware  $\Gamma$  is to integrate the weighted contribution  $w_t(\bar{y}\bar{z})f(\bar{y}\bar{z})$  instead of the full contribution  $f(\bar{y}\bar{z})$  in Equation 13. However,  $w_t$  depends on  $p_t$  (Equation 12) and  $p_t$  depends on  $\gamma_r$  (Equation 11), which in turn depends on  $\Gamma$  (Equation 7). The circular dependency of  $\Gamma \rightarrow w_t \rightarrow p_t \rightarrow \gamma_r \rightarrow \Gamma$  will make it difficult to converge to an optimal  $\Gamma$  in isolated optimization, i.e., starting from a random  $\Gamma$  and updating  $w_t$  and  $\Gamma$  alternately [Rath et al. 2020].

To deal with circular dependency, we analyze the variance resultant from sampling and then propose a solution to an MIS-aware  $\Gamma$  that minimizes the upper bound of variance and proves that the minimization is a convex optimization problem as follows.

Our proof is derived based on two practical assumptions. (i) Different samples of paths for estimation are considered uncorrelated, ignoring the correlation of paths in variance minimization [Veach and Guibas 1995b]. Therefore, covariance of estimator  $\tilde{I}$  equals 0. (ii) Strategy  $t$  generates samples in terms of the target PDF  $p_t(\bar{x})$  in Equation 11, ignoring the difference between the target PDF and actual PDF.

Similar to the balance heuristic, rather than minimizing the variance  $V[\tilde{I}] = E[\tilde{I}^2] - E[\tilde{I}]^2$ , we consider  $E[\tilde{I}^2]$  as the upper bound of  $V[\tilde{I}]$  and minimize the upper bound as

$$\int_{\Omega} \sum_t \frac{w_t^2(\bar{x}) f^2(\bar{x})}{N_t p_t(\bar{x})} d\mu(\bar{x}). \quad (14)$$

Then we substitute  $w_t$  with Equation 12 and obtain:

$$\int_{\Omega} \frac{\sum_t N_t p_t(\bar{x})}{(\sum_t N_t p_t(\bar{x}))^2} f^2(\bar{x}) d\mu(\bar{x}) = \int_{\Omega} \frac{f^2(\bar{x})}{\mathcal{F}(\bar{x})} d\mu(\bar{x}). \quad (15)$$

Here  $\mathcal{F}(\bar{x}) = \sum_t N_t p_t(\bar{x})$ . For  $t \in \{0, 1, k+1\}$ ,  $p_t(\bar{x})$  is uncorrelated to  $\Gamma$  since no subspace sampling is involved in these strategies. Otherwise,  $p_t(\bar{x})$  will be proportional to a specific  $\Gamma_{i,j}$ , according to Equation 11 and Equation 7. Therefore,  $\mathcal{F}$  is computed as

$$\mathcal{F}(\bar{x}) = \sum_t N_t p_t(\bar{x}) = g_0(\bar{x}) + \sum_{i,j} g_{i,j}(\bar{x}) \Gamma_{i,j}, \quad (16)$$

where  $g_0(\bar{x})$  refers to the PDF sum of those strategies handled by classic BDPT and irrelevant to  $\Gamma$ . The gradient  $g_{i,j}(\bar{x}) = \frac{d\mathcal{F}(\bar{x})}{d\Gamma_{i,j}}$  can be obtained by substituting Equation 11 and Equation 7 into  $p_t(\bar{x})$ .

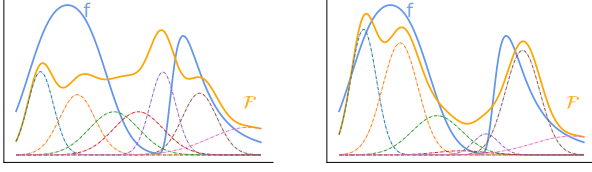


Fig. 5.  $\mathcal{F}$  (orange curve) is the mixture of a serial of pdfs  $g_{i,j}$  (dashed curves) and  $\Gamma$  is the mixture coefficient to describe the ratio of the mixture.  $\Gamma$  should be optimized to minimize  $\int f(\bar{x})^2 / \mathcal{F}(\bar{x}) d\mu(\bar{x})$  and makes  $\mathcal{F}$  close to the target function  $f$  (blue curve).

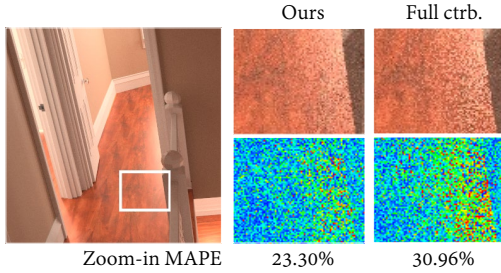


Fig. 6. Full-contribution-based  $\Gamma^{full}$  vs. our MIS-aware  $\Gamma$ . Our  $\Gamma$  shows advantage in reducing variance.

According to Equation 16,  $\mathcal{F}$  is a mixture function of  $g_0$  and  $g_{i,j}$ , and  $\Gamma$  is the mixture coefficient that defines the ratio of the mixture, as shown in Figure 5. Equation 15 is exactly a convex function of  $\Gamma$ , which has been proven in Douc et al. [2007]; Sbert et al. [2016]. Therefore, finding the optimized  $\Gamma$  becomes a convex optimization problem. We employ stochastic gradient descent (SGD) [Kingma and Ba 2014] to find the optimal numerical solution to  $\Gamma$ , which will be explained in detail in Sec. 7.3.

With the MIS-aware  $\Gamma$  discussed above, the MIS term  $w$  is combined with  $\Gamma$  to form the light selection function  $\gamma_r$ . As shown in Figure 6, our MIS-aware  $\Gamma$  achieves robust performance in complex scenes and outperforms the full-contribution-based  $\Gamma^{full}$ , labeled as "Full ctrb.".

#### 6.4 Classification Function

The classification function  $\kappa$  divides space  $\mathcal{S}$  into a set of subspaces and is the basis of our approach. Direct illumination is essential for rendering, and partitioning and clustering of the light sources have been widely discussed in previous works [Vévoda et al. 2018; Walter et al. 2005]. Therefore,  $\kappa$  for sub-paths of the light source can be easily constructed. However, for general sub-paths,  $\mathcal{S}$  is the union of spaces covering arbitrary dimensions. The number of subspaces is usually limited due to  $O(N^2)$  overhead of the subspace sampling matrix  $\Gamma$ . It is impractical to divide sub-path space  $\mathcal{S}$  by simple segmentation. Therefore, we adopt the centroid-distance scheme to define the classification function.

Using a centroid sub-path set  $\{\bar{C}_i\}$  to denote the subspace centers and distance function  $d$  to evaluate the distance between sub-paths, the subspace of a sub-path depends on its closest centroid sub-path.

Therefore, we define the centroid-based  $\kappa$  as

$$\kappa(\bar{y}) = \arg \min_i d(\bar{C}_i, \bar{y}). \quad (17)$$

In practice, we sample the centroid sub-paths set  $\{\bar{C}_i\}$  from the prefix/suffix of high-contribution paths to adaptively improve the approximate estimate of  $\gamma_r$  of important paths. Formally, we run a preprocessing pass to trace multiple full paths  $\{\bar{X}\}$ . When sampling the centroids for light/eye subspaces, we treat each prefix/suffix sub-path of  $\bar{X}$  as a candidate sub-path with a sampling probability proportional to  $f(\bar{X})/p(\bar{X})$ . The centroids can then be resampled from the candidates sub-paths.

We next specify the distance function  $d$  in Equation 17 to measure the difference between sub-paths with respect to  $\gamma$ . As defined in Equation 6,  $\gamma(\bar{y}, \bar{z})$  is proportional to  $w_t(\bar{y}\bar{z})$ ,  $f_{yz}(\bar{y}, \bar{z})$ , and  $f_y(\bar{y})$ . In our method, the  $f_y(\bar{y})$  term can be handled by the second-stage sampling (Sec. 6.1). Because all vertices of a sub-path have effect on  $w$ , we leave the difference of  $w$  aside, similar to Popov et al. [2015], and keep the  $f_{yz}$  similarity within subspace.  $f_{yz}(\bar{y}, \bar{z})$  depends on the position, normal, and incident direction of the last vertex of a sub-path. Therefore, the distance function  $d$  between sub-paths is measured as:

$$d = d_s^2 + \sigma_s^2(1 - \cos\theta_n) + k_d\sigma_s^2(1 - \cos\theta_d), \quad (18)$$

where  $d_s$  is spatial distance,  $\theta_n$  is angle between normals, and  $\theta_d$  is angle between incident, of the last vertices of two sub-paths, respectively;  $\sigma_s^2$  measures the spatial scale of the scene, and we trace a set of full paths and determine  $\sigma_s^2$  by the spatial difference for vertices in the paths;  $k_d$  weighs the difference in the incident direction. In practice,  $k_d$  is set to a small value (generally  $0 \sim 0.05$ ) since position and normal play more important roles in most cases.

Centroid-distance computation is expensive because a set of centroids with  $N_s$  elements has  $O(N_s)$  complexity to compute  $\kappa$ . In practice, we generate a set of sub-paths and bin them into subspaces using the centroid-based  $\kappa$ . We then use the labeled sub-paths as the training set to grow a decision tree to fit centroid-based  $\kappa$ . The label of a node in the decision tree is determined by the subspace with the most labeled sub-paths, or the same as its parent node if there is no sub-path in the node. We split the node when less than 99% of the sub-paths in the node are correctly classified. When a node splits, we randomly select an attribute from position, normal, and incident direction, and compute its bounds with sub-paths included. We set the center of the bounds as the decision threshold.

$\kappa$  is implemented using the decision tree with low computational overhead. Figure 7 shows the procedure of building a decision tree, and we also visualize the classification results accordingly, using Cornell Box as an example and ignoring the incident difference.

## 7 TECHNICAL DETAILS AND ALGORITHM

### 7.1 Pre-traced Path Dataset $\mathcal{D}$

Statistics of the path space are required to determine the relevant terms in our function, including the classification function  $\kappa$ , subspace sampling matrix  $\Gamma$ , and normalization factor  $Q$ . We, therefore, construct a full path dataset  $\mathcal{D}$  as the path space in the pre-trained stage.  $\mathcal{D}$  stores the necessary information of each full path  $\bar{X}$ , including  $f(\bar{X})$ ,  $p(\bar{X})$ , positions of each vertex  $X_i \in \bar{X}$ ,  $p(\bar{Z})$  for each

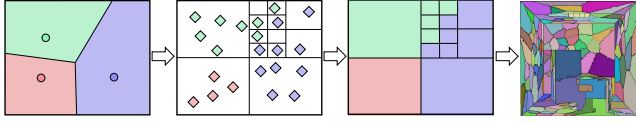


Fig. 7. Decision tree is used to fit the centroid-based  $\kappa$ . We trace multiple sub-path samples (marked with diamonds) and label each sub-path by its closest centroid (marked with circle). The decision tree is trained based on the labeled sub-paths. We visualize the result of light sub-path classification by decision tree in Cornell Box.

suffix eye sub-path  $\bar{Z}$ , and  $f_y(\bar{Y})$  for each prefix light sub-path  $\bar{Y}$ . We run BDPT to generate  $\mathcal{D}$ , and other path tracing and sampling methods are also workable. Those paths that can not be improved by probabilistic connections will be discarded.

The optimization should be performed based on the relative luminance of each pixel [Rath et al. 2020]. Therefore, we use the relative contribution  $f(\bar{X})/\bar{I}_{\bar{X}}$  for training instead of the absolute contribution  $f(\bar{X})$ , where  $\bar{I}_{\bar{X}}$  is the pixel's luminance corresponding to path  $\bar{X}$ , which can be estimated from dataset  $\mathcal{D}$ . Because the paths in  $\mathcal{D}$  is usually insufficient to accurately estimate each pixel, we use a lower-resolution downsampling strategy when estimating  $\bar{I}_{\bar{X}}$  (generally set to  $0.1 \times$  the image width and height).

## 7.2 Classification Function $\kappa$

**Light source division:** Since direct illumination is usually essential for the rendering, we give around 20% of the subspace budget  $N_S$  to divide the light source and make a specific classification function  $\kappa_{\mathcal{L}}$  for light source sub-paths. In our implementation, the light source is evenly divided, and the light source sub-paths are binned into the subspace based on the UV coordinate of the light surface or environment map, which is sufficient for robust performance. For complex light source settings, the centroid-based method discussed in Section 6.4 is still available.

**Centroid-based  $\kappa$  and decision tree:** We sample the centroid sub-paths from the prefix/suffix of the full paths stored in  $\mathcal{D}$ . The probability of a sub-path being sampled is proportional to the contribution of the corresponding full path. For  $d$  in Equation 18, we set  $\sigma_s^2 = \max(\sigma_x^2, \sigma_y^2, \sigma_z^2)$ , where  $\sigma_x^2$ ,  $\sigma_y^2$  and  $\sigma_z^2$  are variances of the path vertices in  $\mathcal{D}$  in Cartesian coordinate, respectively. In practice, a large  $k_d$  setting in  $d$  can improve the rendering of glossy material, while the normal and position are more important in most cases. Therefore, we set  $k_d$  to 0.05 as default. Two decision trees are trained as practical  $\kappa_{\mathcal{L}}$  and  $\kappa_{\mathcal{E}}$ , respectively, using the prefix/suffix sub-paths stored in  $\mathcal{D}$  and labeled by the centroid-based  $\kappa$  as the training set.

## 7.3 Estimation of $Q$ and $\Gamma$

Once  $\kappa_{\mathcal{L}}$  is specified, we trace  $M' \gg M$  light sub-paths from the light source and classify each light sub-path by  $\kappa_{\mathcal{L}}$ . The normalization term  $Q$  for  $\gamma_r$  is evaluated as

$$Q(i) = \frac{1}{M'} \sum_j \{f_y(\bar{Y}_j)/p(\bar{Y}_j) | \kappa_{\mathcal{L}}(\bar{Y}_j) = i\}. \quad (19)$$

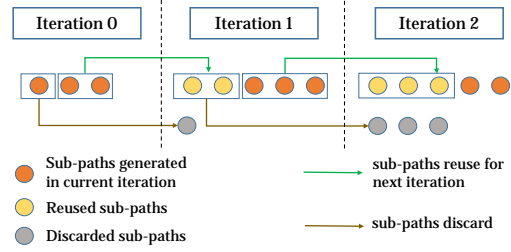


Fig. 8. Cross-iteration reuse of the light sub-paths. The subspace LVC retains the latest  $M_0$  sub-paths for resampling, making it possible for a light sub-path to be reused in successive iterations.

$\Gamma$  can be obtained by minimizing the upper bound of variance for all the paths stored in  $\mathcal{D}$ . We thereby define the loss function as

$$Loss(\Gamma) = \sum_{\bar{X} \in \mathcal{D}} \frac{f^2(\bar{X})}{p(\bar{X})\mathcal{F}(\bar{X}|\Gamma)},$$

where  $p(\bar{X})$  is the PDF used to sample  $\bar{X}$  in  $\mathcal{D}$ , and  $\mathcal{F}$  is the mixture function as discussed in Sec. 6.3.

We minimize the loss function by gradient descent. Similar to Vorba et al. [2019], to keep the constraint  $\Gamma \geq 0$ , we represent  $\Gamma$  as a sigmoid function applied to an  $\mathcal{R}$  matrix  $\Theta$  and normalized by columns, i.e.,  $\Gamma_{i,j} = \text{Sigmoid}(\Theta_{i,j}) / \sum_{i'} \text{Sigmoid}(\Theta_{i',j})$ . We optimize  $\Theta$  instead of  $\Gamma$  directly, and an Adam optimizer [Kingma and Ba 2014] is employed to minimize the loss function, where  $\Theta$  gradient can be computed by back propagation.

Generally, an interpolation of  $\Gamma$  with a 10% uniform distribution as conservative sampling [Georgiev et al. 2012b] is necessary. We compute the gradient of  $\Gamma$  knowing the interpolation.

## 7.4 Cross-iteration Light Sub-path Reuse

We employ a two-stage method to sample light sub-paths in  $\gamma_r$  (Sec. 6.1), wherein the second-stage resampling requires a large number of candidate light sub-paths for sampling in a distribution that approximates  $f_y(\bar{y})$ . However, in the highly complex scenes, even a fairly large  $M$  can not guarantee that each subspace gets sufficient sub-paths.

We solve this by retaining the light sub-paths in difficult-to-sample subspace for more than one iteration. As shown in Figure 8, for a light subspace, we maintain the LVC as a first-in-first-out queue, keeping at most  $M_0$  candidate light sub-paths. At each iteration, we push the newly traced light sub-paths into the associated subspace's LVC and pop the old sub-paths (i.e., discard) when there are more than  $M_0$  sub-paths remaining. Therefore, in difficult subspaces, a light sub-path can be retained and reused for multiple iterations, ensuring that there are sufficient candidate sub-paths for resampling.

We denote the number of sub-paths retained in the LVC as  $M_r(\bar{Y})$  and the number of sub-paths generated from subspace  $\kappa_{\mathcal{L}}(\bar{Y})$  as  $M(\bar{Y})$ , including those retained and discarded sub-paths. Because the order of sub-paths is random during the stochastic generation, retaining sub-paths can be treated as sampling  $M_r(\bar{Y})$  sub-paths from  $M(\bar{Y})$  sub-paths. The probability for a generated sub-path  $\bar{Y}$



to be retained is  $\frac{M_r(\bar{Y})}{M(\bar{Y})}$ . In this case,  $P_r(\bar{Y})$  is computed as

$$P_r(\bar{Y}) = \frac{M_r(\bar{Y})}{M(\bar{Y})} \frac{f_y(\bar{Y})/p(\bar{Y})}{\sum_{\bar{Y}_i \in LVC(\bar{Y})}^{M_r(\bar{Y})} f_y(\bar{Y}_i)/p(\bar{Y}_i)}. \quad (20)$$

In  $j$ -th iteration, there are in total  $jM$  sub-paths traced from the light source. Therefore, we reformulate the estimator in Equation 10 as:

$$\tilde{I}_t = \frac{1}{jMN} \sum_{i=1}^N \frac{w_t(\bar{Y}_i \bar{Z}) f(\bar{Y}_i \bar{Z})}{p(\bar{Z}) \Gamma(\kappa_{\mathcal{L}}(\bar{Y}_i), \kappa_{\mathcal{E}}(\bar{Z})) P_r(\bar{Y}_i)}. \quad (21)$$

Empirically,  $M_0 = 200$  is sufficient to provide a fair resampling (see experimental result in Sec. 8.5.3).

## 7.5 Algorithm

---

### Algorithm 1: SPCBPT

---

```

pre-trace paths by BDPT and construct the path dataset  $\mathcal{D}$ ;
construct  $\kappa$  from path dataset  $\mathcal{D}$ ;
estimate  $Q$  by tracing  $M'$  light sub-paths from light sources;
estimate  $\Gamma$  in given  $\kappa, Q, \mathcal{D}$  by Adam optimizer;
for  $n = 1$  to  $maxIteration$  do
  generate candidate light sub-paths by tracing  $M$  sub-paths from
  light sources;
  update the sub-paths retained in subspace LVCs according to  $\kappa_{\mathcal{L}}$ ;
  reconstruct resampling distribution for the second-stage sampling;
  for each pixel do
    trace one eye sub-path from camera;
    deal with strategies  $t \in \{0, 1, k + 1\}$  using BDPT;
    for each valid eye sub-path  $\bar{Z}$  with  $t \geq 2$  vertices do
      for  $i = 1$  to  $N$  do
        sample light subspace  $\kappa_{\mathcal{L}}(\bar{Y})$  from PMF
         $\Gamma(\kappa_{\mathcal{L}}(\bar{Y}) | \kappa_{\mathcal{E}}(\bar{Z}))$ ;
        resample light sub-path  $\bar{Y}$  from subspace  $\kappa_{\mathcal{L}}(\bar{Y})$ ;
        perform visibility test;
        if pass visibility test then
          estimate  $\tilde{I}_t$  for the full path  $\bar{Y}\bar{Z}$ ;
        end
      end
    end
    update pixel intensity;
  end
end

```

---

We present the pseudocode in Algorithm 1.  $\kappa, Q$  and  $\Gamma$  are determined in the preprocessing stage. The cross-iteration reuse is applied to the light sub-path tracing pass at the beginning of each iteration. We compute the MIS weighting function using recursive MIS (RMIS) [V. Antwerpen 2011], and we provide the details in the supplementary material.

Our method is different from PCBPT and RISBPT [Nabata et al. 2020a; Popov et al. 2015]; given that  $\Gamma, Q$ , and  $\kappa$  are pre-trained in the preprocessing, no additional visibility test is required to construct the light selection distribution in the runtime stage (see algorithm 1), and the intersection overhead can be significantly saved. Those

terms involved in the contribution of a connection (like visibility, MIS weight, etc.) are covered by the first-stage sampling using  $\Gamma$ , which is consistent when the light sub-paths changes.

## 8 EXPERIMENTS AND RESULTS

We present the experimental results of our algorithm in different scenarios. We make comparisons with LVCBPT [Davidovič et al. 2014], which has GPU acceleration of BDPT as the baseline, and the state-of-the-art method RISBPT [Nabata et al. 2020a] on GPU. Mean absolute percentage error (MAPE) is used as the metric for comparison. We also discuss the effect of using different parameters in our algorithm, as well as the applicable scenarios and limitations.

### 8.1 Experimental Setting

**8.1.1 Renderer Setting.** All the algorithms are implemented based on the OptiX architecture [Parker et al. 2010], and run on an NVIDIA GeForce RTX 2080 SUPER GPU with an Intel Core i7-10700 CPU on a Windows system. All images are rendered in a resolution of  $1920 \times 1000$ . RISBPT adopts the same parameter configuration as recommended in [Nabata et al. 2020a]. In each iteration, PCBPT traces  $M = 200$  light sub-paths for resampling while both LVCBPT and our algorithm set  $M = 100,000$ . For each traced eye sub-path, the number of light sub-paths resampled for connection  $N = 3$  is set for all algorithms. We enable all the strategies involved in BDPT, including unidirectional path tracing and light tracing (i.e., strategies  $s = 0$  and  $t = 1$ ).

Our approach traces 1,000,000 full paths to construct the dataset  $\mathcal{D}$  in the preprocessing stage, trace  $M' = 2,000,000$  light sub-paths to estimate  $Q$ .  $\Gamma$  is optimized by an Adam optimizer with  $\alpha = 0.1$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 1e-8$ . Each batch we compute the gradient for 10,000 full path and run 2 epochs to figure out  $\Gamma$ . The number of subspace  $N_S = 1,000$  is set for light sub-path and eye sub-path. We use the above settings for the experiments unless otherwise stated.

**8.1.2 Scene Settings.** Eight benchmarks, including Bedroom, Door, Garden, Hallway, House, Kitchen, Sponza, and Diningroom, are tested. These benchmarks cover a variety of scenes, illumination conditions, and surface materials. Bedroom is illuminated by an environment map and the curtains mask most of the light transport. Door is illuminated by the area light located in the next room and features difficult shadowing. Garden and Hallway are illuminated by an environment map with directionality; Garden features an outdoor environment while Hallway features an interior one. The glossy wood floor in Hallway would reflect sunlight to the ceiling and make caustics. House is an interior scene lit by two large-area lights outside. Garden, Hallway, and House are complex scenes and sampling paths with distinct characteristics is a challenge. Kitchen contains rich glossy materials, and the glass windows prevent straightforward next-event estimation (NEE) because they are in front of the light source, so path guiding is necessary to handle this scene (see Sec. 8.4). Sponza features a sharp irradiance change that comes from the strong directionality of the environment map. Diningroom is illuminated by cold indoor light and a warm environment map and features tone differences. The reference images are rendered by LVCBPT in 150K iterations for Bedroom, Hallway and Diningroom, and 50K iterations for other scenes.

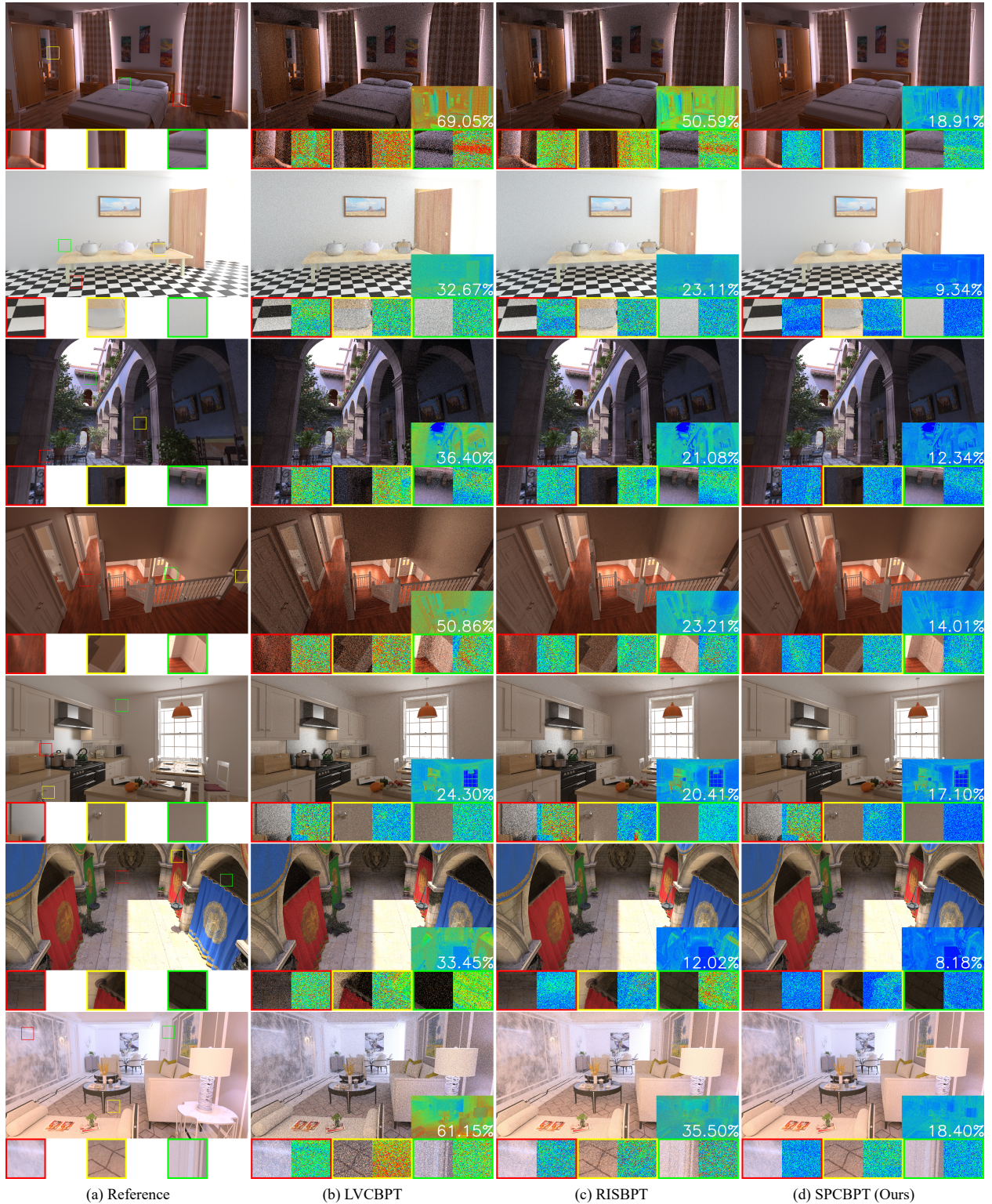


Fig. 9. Equal-time (1 min) comparisons: LVCBPT [Davidović et al. 2014], RISBPT [Nabata et al. 2020a], and our approach. Our preprocessing time has been counted in the total time budget. Our approach obtains higher quality in terms of MAPE and the zoom-in regions. Seven benchmarks include Bedroom, Door, Garden, House, Kitchen, Sponza, and Diningroom (from top to bottom), covering various types of scenes with different illumination conditions and materials.



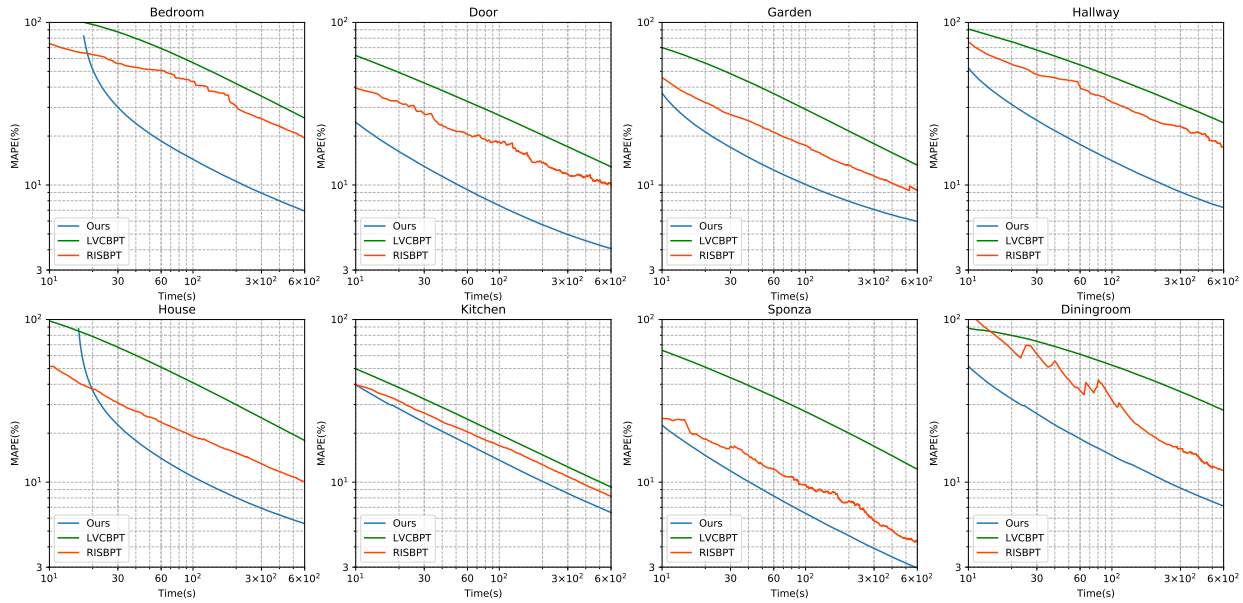


Fig. 10. Convergence over time (10 mins) in terms of MAPE: LVCBPT [Davidović et al. 2014], RISBPT [Nabata et al. 2020a], and our approach. Eight benchmarks, including seven scenarios corresponding to Figure 9 and a Hallway scene corresponding to Figure 1. Although the extra time cost used for our approach’s preprocessing has been included, our algorithm outperforms others with superior performance and exhibits a stable convergence tendency.

**8.1.3 Overhead.** Our training-based method needs additional time and storage overhead. Table 1 provides the preprocessing time cost for each scene, which is used to construct the training dataset for the specific scene and to resolve subspace information including  $\kappa$  and  $\Gamma$ . Construction of  $\kappa$  with optimization of  $\Gamma$  takes less than 2s for each test scene. Meanwhile, at most 20 seconds are required to construct the training dataset  $\mathcal{D}$  by using BDPT, especially in scenes with difficult visibility.

The extra runtime storage for decision tree and  $\Gamma$  of each scene does not exceed 100MB in our experiment. In the preprocessing stage, dataset  $\mathcal{D}$  requires at most 512MB memory (depending on the average path length of the scene) to store the path information. However, we can discard  $\mathcal{D}$  once we obtain  $\kappa$ ,  $Q$ , and  $\Gamma$ .

## 8.2 Performance Evaluation

We demonstrate the performance of different methods within the same time budget in Figure 1 and Figure 9. We also highlight the zoom-in regions using error visualization with heat map. Our algorithm can significantly reduce the noise and obtain superior quality in terms of MAPE in all test scenarios. Note that our algorithm’s additional preprocessing time cost has been included in all experiments. Although the preprocessing takes a particular time cost (a few seconds), the benefits gained are considerable in terms of the significant improvement of performance.

In Table 1, we show the statistics of time cost and iterations required to reach the same MAPE (12%). We highlight that our approach obtains higher efficiency over RISBPT with several times speedup, e.g., over 10× in Hallway and Bedroom. Moreover, our approach uses fewer iterations than RISBPT, and takes less time for each iteration. In all, our approach reduces both the time cost and the number of iterations significantly in all test benchmarks due

to our powerful two-stage sampling and the MIS-aware  $\Gamma$ , which minimizes the rendering variance.

Figure 10 shows the convergence over time (10 mins) in terms of MAPE in all scenarios. Our approach outperforms others in terms of convergence speed and exhibits a stable convergence tendency. From Figure 10, RISBPT’s overall performance is better than LVCBPT, but the overhead of PMF record generation and KD-tree query decreases the possible iteration counts, as stated in Nabata et al. [2020a]. In different scenarios, our approach shows a consistent advantage over other methods except at the early stages of iterations. Our approach can sample probabilistic connections efficiently both in simple scenarios and complex scenarios (e.g., Hallway with very complicated light path) with lower computational overhead.

## 8.3 Comparison and Analysis

Unlike RISBPT, our approach considers MIS weights when constructing and sampling from light sub-path selection distribution. In addition, using thousands of shadow rays for each light sub-path at each iteration to construct the PMF cache is unnecessary, allowing us to resample more light sub-paths ( $M = 100k \gg 200$ ).

We make further comparisons with RISBPT, and Table 2 shows the results of MAPE and time cost with the same 100 iterations. The rendering quality in terms of MAPE reflects how well different methods can handle the probabilistic connections. Our approach achieves better performance in all scenes, especially in complex scenes, where RISBPT would suffer from the limited resampled light sub-paths number  $M = 200$ . The powerful two-stage sampling allows our algorithm to use a massive  $M$  setting and make an outstanding rendering in various scene settings.

Kitchen may be an exceptional case, where  $M = 200$  seems sufficient to capture the illumination condition. In this case, the subpath-to-subpath style importance sampling of RISBPT would provide



Table 1. Time cost (measured in seconds) and iterations to reach the same MAPE (all scenes are 12%) with different methods testing on eight benchmarks. Our method shows higher efficiency over the state-of-the-art method (RISBPT) with several times speedup (esp. over 10× in Hallway and Bedroom). The preprocessing time used for our algorithm has been included, which is 16.58s for Bedroom, 3.41s for Door, 6.90s for Garden, 6.49s for Hallway, 15.5s for House, 3.03s for Kitchen, 3.21s for Sponza, and 5.43s for Diningroom.

Scene	Bedroom		Door		Garden		Hallway		House		Kitchen		Sponza		Diningroom	
	Time	Itr.	Time	Itr.	Time	Itr.	Time	Itr.	Time	Itr.	Time	Itr.	Time	Itr.	Time	Itr.
LVCBPT	3182.7	6075	724.2	2328	756.0	2023	3295.8	4498	1471.0	4741	325.6	562	599.6	1852	4207.5	7068
RISBPT	2136.8	1407	273.4	269	255.7	357	1983.8	766	364.2	550	232.9	165	60.7	94	569.8	268
Ours	<b>147.48</b>	<b>171</b>	<b>35.6</b>	<b>59</b>	<b>64.5</b>	<b>124</b>	<b>146.6</b>	<b>132</b>	<b>80.6</b>	<b>150</b>	<b>134.5</b>	<b>150</b>	<b>28.8</b>	<b>68</b>	<b>160.9</b>	<b>187</b>

Table 2. Comparison of time cost and accuracy (MAPE) after 100 iterations between RISBPT [Nabata et al. 2020a] and our method.

Scene	Method	RISBPT		Ours	
		Time (s)	MAPE	Time (s)	MAPE
Bedroom		149.43	36.95%	95.12	14.92%
Door		100.89	18.34%	58.34	9.47%
Garden		70.99	19.72%	53.18	13.01%
Hallway		252.94	23.45%	112.27	13.42%
House		65.83	22.47%	58.70	14.19%
Kitchen		140.63	14.87%	90.98	14.23%
Sponza		64.54	11.48%	41.05	9.94%
Diningroom		211.68	18.18%	88.44	15.48%

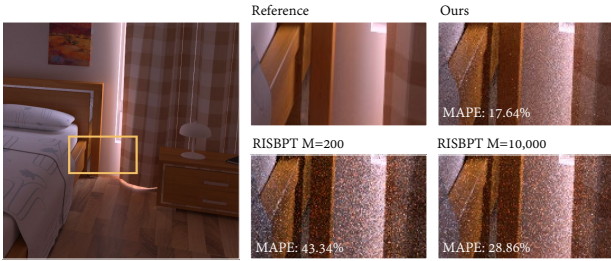


Fig. 11. Our approach outperforms RISBPT with different settings ( $M = 200$  and  $M = 10,000$ , respectively), all using 64 iterations.

fine-grained guidance to the probabilistic connections, so our algorithm achieves only a slight advantage over RISBPT in Kitchen. In addition, we investigate the performance of using different  $M$  for RISBPT. since Bedroom has many difficult visibility, a large  $M$  will be helpful. Figure 11 shows the zoom-in image for comparison, where in each iteration  $M = 10,000$  light sub-paths are traced for RISBPT to resample, which is 50× the recommended setting. In such a situation, RISBPT takes extra seconds to construct the PMF distribution in each iteration and obtains an improvement in MAPE 43.34% → 28.86%. However, RISBPT still falls far behind our approach using the same number of iterations, not to mention the extra computational costs required. An increment of  $M$  will not significantly improve RISBPT’s overall performance.

To sum up, our algorithm provides stable sampling effectiveness for paths that have difficulty sampling in previous algorithms.



Fig. 12. Equal-time (1 min) comparisons: LVCBPT + path guiding (upper left), RISBPT + path guiding (upper right), our approach (bottom left), and our approach + path guiding (bottom right).

Our approach achieves better performance than RISBPT in various scenes, whether measured in equal time or equal iterations.

#### 8.4 Combination with Path Guiding

We integrate the path guiding method based on the SD-tree [Müller et al. 2017] with our approach. Following the same setup, we use 1 min to pre-train the SD-tree and then trace eye light sub-paths and light sub-paths with guidance.

We test the performance in Kitchen, where the glass window prevents the application of NEE. Figure 12 shows equal-time (1 min) results with four settings: LVCBPT with path guiding, RISBPT with path guiding, and our method without/with path guiding. Our approach works well with the path guiding algorithm and benefits from the guided paths by improved quality in MAPE 22.59% → 17.1%. Even though all methods combine with path guiding, our method still outperforms LVCBPT and RISBPT.

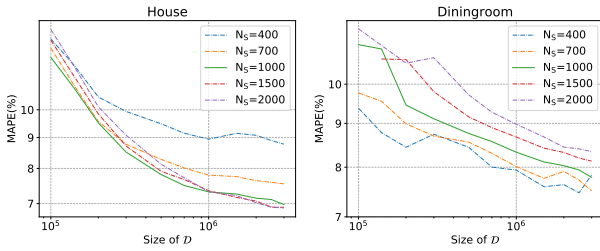
#### 8.5 Parameters

We study the reliability of our approach by tuning key parameters that may affect the performance. These parameters include the number of samples  $N$  used for a single eye vertex to sample and build connections, the effect of light sub-path reuse, the size of pre-traced dataset  $\mathcal{D}$ , and the classification subspace number  $N_S$ .

**8.5.1 Number of Samples  $N$ .** Table 3 shows the performance using different  $N$ , the number of light sub-paths for an eye sub-path to connect. Obviously, a higher  $N$  can improve the performance of each iteration and decrease the iterations needed to achieve the same MAPE, but it needs more visibility tests per iteration and

Table 3. Using different  $N$ , the time cost and iterations to reach MAPE=8%.

N	Scene		Hallway		Sponza	
	Itr.	Time (s)	Itr.	Time (s)	Itr.	Time (s)
1	932	499.40	335	76.50		
2	570	462.61	204	63.36		
3	408	443.35	158	63.50		
4	352	474.17	134	64.17		
8	264	641.60	97	75.94		

Fig. 13. Different settings: number of subspace ( $N_S$ ) and size of dataset ( $\mathcal{D}$ ) used in House and Diningroom, all MAPE obtained after 512 iterations. The horizontal axis indicates the scale of  $\mathcal{D}$ .

slows down the speed. The best performance of our algorithm can be obtained when  $N = 2\sim 4$ , so we use  $N = 3$  as default.

**8.5.2 Size of Path Dataset  $\mathcal{D}$  and Number of Subspace  $N_S$ .** Figure 13 shows the equal iteration performance with different size of  $\mathcal{D}$  and  $N_S$  used in House and Diningroom, where we focus on the rendering quality regardless of the preprocessing time.

The approximation function  $\gamma_r$  can approximate the ideal function  $\gamma$  better as  $N_S$  increases, but it requires a larger  $\mathcal{D}$  to train  $\Gamma$  to avoid overfitting. In relatively simple scenes like Diningroom, a small  $N_S$  is sufficient, while  $\mathcal{D}$  size is more important for the render performance. In contrary, House requires more subspaces to capture the complex light transport.

In the experiments, setting  $N_S$  to 1000 and  $\mathcal{D}$  to 1,000,000 works well even for complex scenes, since the paths with high contributions are usually concentrated in a few regions rather than evenly distributed in the scene. When the time budget is limited, i.e., in a time-critical condition, the number of pre-traced paths can be appropriately reduced. Our approach still provides a powerful improvement in cases where  $\mathcal{D}$  is small.

**8.5.3 Effect of Light Sub-path Cross-iteration Reuse.** Light sub-path cross-iteration reuse can improve the performance of the second-stage resampling. Figure 14 shows the performance comparison of  $M = 100,000$  with/without cross-iteration reuse and  $M = 1,000,000$  without cross-iteration reuse. Compared to simply increasing  $M$ , cross-iteration reuse makes better use of difficult-to-sample light sub-paths. It provides better resampling in complex scenes such as House without the extra overhead of sub-path tracing. For scenes with easy visibility (Kitchen), only reusing  $M_0 = 200$  sub-paths lowers the rendering slightly, but the difference is not noticeable (14.35% vs. 14.12% after 128 iterations). This demonstrates that our reuse of light sub-paths benefits those scenes with difficult visibility.

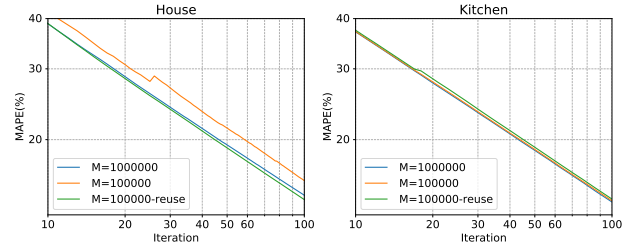
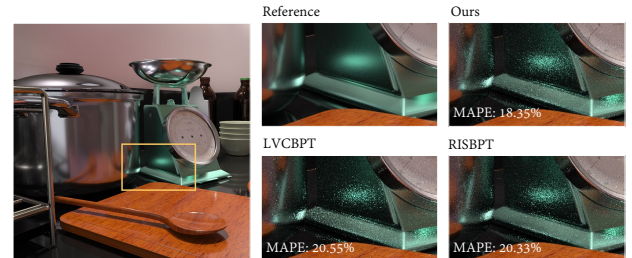
Fig. 14. Convergence over iterations with three different settings:  $M = 100,000$  with/without sub-path cross-iteration reuse,  $M = 1,000,000$  without cross-iteration reuse, in House and Kitchen.

Fig. 15. Highly glossy scene rendered within 1 min. The improvement of our approach over other methods is relatively low. None of them can handle this scene very well.

## 9 CONCLUSION, LIMITATION, AND FUTURE WORK

Based on our formulation of subspace for probabilistic connections, we present a feasible subspace-based method for importance sampling from a large number of candidate light sub-paths. By first sampling the light subspace and then resampling the light sub-path from the light subspace, our algorithm can significantly improve the rendering quality in the complex scene with little computation overhead. Our work for the probabilistic connections has no specific restrictions on the way the sub-path is generated. Therefore, any specific light path generation method like path guiding (see Sec. 8.4) can be available for our algorithm.

In the pre-processing stage, our approach requires a large number of paths to train the approximate function  $\gamma_r$ . However, even pre-tracing these paths for training remains a challenge in a scene with difficult visibility. Adaptive sampling strategies like MCMC [Veach and Guibas 1997] may be a solution to this problem and need further investigation. Online training strategies can be an alternative to solving the function  $\gamma_r$  instead of pre-training and deserve future investigation. Our proposed MIS-aware  $\Gamma$  is based on the balance heuristic. The combination of our method with other MIS weighting functions like correlation-aware MIS [Grittmann et al. 2021] should be discussed further.

In this paper,  $\gamma$  is evaluated approximately by dividing  $\mathcal{S}$  into different subspaces. However, the way of light subspace division may not applied to all eye subspaces. A coarse-to-fine division may provide a better approximation, but the sampling and MIS computation for such a division need further investigation.

Because of the limitation of BDPT, some difficult path configurations are still hard to handle, e.g., highly glossy material (see

Figure 15) or caustics caused by perfect specular bounce. Our approach also can not handle this type of scene very well. Generally, path guiding is necessary to help render the glossy surface smoothly. The VCM/UPS framework [Georgiev et al. 2012a; Hachisuka et al. 2012] provides approach to combine BDPT with PPM to solve the caustics paths and can also be combined with our algorithm. Recent works on PPM have achieved great improvement by reducing the radius of photon collection [Hachisuka and Jensen 2009; Lin et al. 2020] and extending the form of photon into a high-dimensional photon surface [Deng et al. 2019]. A combination of our algorithm with these methods to further improve the efficiency will be an interesting future work.

## ACKNOWLEDGMENTS

We thank all the anonymous reviewers for helpful suggestions. This work is supported by the National Key R&D Program of China (No. 2021YFF0500901) and NSFC of China (No. 62172013). We also thank the following people for providing test scenes: Wig42 (Hallway), SlykDrako (Bedroom), Miika Aittala et al. (Door), Marios Papas et al. (Sponza), Guillermo et al. (Garden), Jay-Artist (Kitchen), MrChimp2313 (House), Davilion (Glossy material scene).

## REFERENCES

- B. Bitterli, C. Wyman, M. Pharr, P. Shirley, A. Lefohn, and W. Jarosz. 2020. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Trans. Graph. (TOG)* 39, 4 (07 2020), 148:1–17.
- C. R. A. Chaitanya, L. Belcour, T. Hachisuka, S. Premoze, J. Pantaleoni, and D. Nowrouzezahrai. 2018. Matrix Bidirectional Path Tracing. In *Eurographics Symposium on Rendering - Experimental Ideas & Implementations*. The Eurographics Association, 23–32.
- T. Davidović, J. Krivanek, M. Hašan, and P. Slusallek. 2014. Progressive Light Transport Simulation on the GPU. *ACM Trans. Graph. (TOG)* 33, 3 (05 2014), 29:1–19.
- X. Deng, S. Jiao, B. Bitterli, and W. Jarosz. 2019. Photon surfaces for robust, unbiased volumetric density estimation. *ACM Trans. Graph. (TOG)* 38, 4 (07 2019), 46:1–12.
- S. Diolatzis, A. Gruson, W. Jakob, D. Nowrouzezahrai, and G. Drettakis. 2020. Practical Product Path Guiding Using Linearly Transformed Cosines. *Computer Graphics Forum* 39, 4 (07 2020), 23–33.
- R. Douc, A. Guillin, J. M. Marin, and C. P. Robert. 2007. Minimum variance importance sampling via Population Monte Carlo. *ESAIM: Probability and Statistics* 11 (2007), 427–447. <https://doi.org/10.1051/ps:2007028>
- I. Georgiev, J. Krivanek, T. Davidović, and P. Slusallek. 2012a. Light Transport Simulation with Vertex Connection and Merging. *ACM Trans. Graph. (TOG)* 31, 6 (11 2012), 192:1–192:10.
- I. Georgiev, J. Krivanek, S. Popov, and P. Slusallek. 2012b. Importance Caching for Complex Illumination. *Computer Graphics Forum* 31, 2 (05 2012), 701–710.
- P. Grittmann, I. Georgiev, and P. Slusallek. 2021. Correlation-Aware Multiple Importance Sampling for Bidirectional Rendering Algorithms. *Computer Graphics Forum* 40, 2 (05 2021), 231–238.
- J. Guo, P. Bauszat, J. Bikker, and E. Eisemann. 2018. Primary sample space path guiding. In *Eurographics Symposium on Rendering*, Vol. 2018. The Eurographics Association, 73–82.
- T. Hachisuka and H. W. Jensen. 2009. Stochastic Progressive Photon Mapping. *ACM Trans. Graph. (TOG)* 28, 5 (12 2009), 141:1–8.
- T. Hachisuka, J. Pantaleoni, and H. W. Jensen. 2012. A Path Space Extension for Robust Light Transport Simulation. *ACM Trans. Graph. (TOG)* 31, 6 (11 2012), 191:1–10.
- M. Hašan and F. Pellacini. 2007. Matrix Row-Column Sampling for the Many-Light Problem. *ACM Trans. Graph. (TOG)* 26, 3 (08 2007), 26:1–10.
- S. Herholz, O. Elek, J. Vorba, H. Lensch, and J. Krivanek. 2016. Product Importance Sampling for Light Transport Path Guiding. *Computer Graphics Forum* 35, 4 (06 2016), 67–77.
- W. Jakob and S. Marschner. 2012. Manifold Exploration: A Markov Chain Monte Carlo Technique for Rendering Scenes with Difficult Specular Transport. *ACM Trans. Graph. (TOG)* 31, 4 (07 2012), 58:1–13.
- H. w. Jensen. 1996. Importance Driven Path Tracing Using the Photon Map. *Eurographics Rendering Workshop* (09 1996), 326–335.
- A. Keller. 1997. Instant radiosity. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. 49–56.
- D. Kingma and J. Ba. 2014. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations* (12 2014).
- J. Krivanek, M. Hasan, A. Arbree, C. Dachsbacher, A. Keller, and B. Walter. 2014. Scalable Realistic Rendering with Many-Light Methods. *Computer Graphics Forum* 33, 1 (02 2014), 88–104.
- E. P. Lafortune and Y. D. Willems. 1993. Bi-Directional Path Tracing. In *Proceedings of Compugraphics*. 145–153.
- E. P. Lafortune and Y. D. Willems. 1999. A 5D Tree to Reduce the Variance of Monte Carlo Ray Tracing. In *Rendering Techniques*.
- Z. Lin, S. Li, X. Zeng, C. Zhang, J. Jia, G. Wang, and D. Manocha. 2020. CPPM: chi-squared progressive photon mapping. *ACM Trans. Graph. (TOG)* 39, 6 (11 2020), 240:1–12.
- T. Müller, B. McWilliams, F. Rousselle, M. Gross, and J. Novák. 2019. Neural Importance Sampling. *ACM Trans. Graph. (TOG)* 38, 5 (10 2019), 145:1–19.
- T. Müller, F. Rousselle, A. Keller, and J. Novák. 2020. Neural Control Variates. *ACM Trans. Graph. (TOG)* 39, 6 (11 2020), 243:1–19.
- T. Müller, M. Gross, and J. Novák. 2017. Practical Path Guiding for Efficient Light-Transport Simulation. *Computer Graphics Forum* 36, 4 (07 2017), 91–100.
- K. Nabata, K. Iwasaki, and Y. Dobashi. 2020a. Resampling-aware Weighting Functions for Bidirectional Path Tracing Using Multiple Light Sub-Paths. *ACM Trans. Graph. (TOG)* 39, 2 (03 2020), 15:1–11.
- K. Nabata, K. Iwasaki, and Y. Dobashi. 2020b. Two-stage Resampling for Bidirectional Path Tracing with Multiple Light Sub-paths. *Computer Graphics Forum* 39, 7 (10 2020), 219–230.
- J. Ou and F. Pellacini. 2011. LightSlice: matrix slice sampling for the many-lights problem. *ACM Trans. Graph. (TOG)* 30, 6 (12 2011), 179:1–8.
- A. Pajot, L. Barthe, M. Paulin, and P. Poulin. 2011. Combinatorial bidirectional path-tracing for efficient hybrid CPU/GPU rendering. *Computer Graphics Forum* 30, 2 (2011), 315–324.
- S. G. Parker, J. Bigler, A. Dietrich, H. Friedrich, J. Hoberock, D. Luebke, D. McAllister, M. McGuire, K. Morley, A. Robison, et al. 2010. Optix: a general purpose ray tracing engine. *ACM Trans. Graph. (TOG)* 29, 4 (07 2010), 66:1–13.
- S. Popov, R. Ramamoorthi, F. Durand, and G. Drettakis. 2015. Probabilistic Connections for Bidirectional Path Tracing. *Computer Graphics Forum* 34, 4 (07 2015), 75–86.
- A. Rath, P. Grittmann, Sebastian H., P. Vévoda, P. Slusallek, and J. Křivánek. 2020. Variance-Aware Path Guiding. *ACM Trans. Graph. (TOG)* 39, 4 (07 2020), 151:1–12.
- M. Sbert, V. Havran, and L. Szirmay-Kalos. 2016. Variance Analysis of Multi-sample and One-sample Multiple Importance Sampling. *Computer Graphics Forum* 35, 7 (10 2016), 451–460.
- J. Talbot, D. Cline, and P. Egbert. 2005. Importance Resampling for Global Illumination. *Eurographics Symposium on Rendering*, 139–146.
- Y. Tokuyoshi and T. Harada. 2019. Hierarchical russian roulette for vertex connections. *ACM Trans. Graph. (TOG)* 38, 4 (07 2019), 36:1–12.
- D. V. Antwerpen. 2011. *Recursive MIS Computation for Streaming BDPT on the GPU*. Technical Report.
- E. Veach and L. Guibas. 1995a. Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques*. 145–167.
- E. Veach and L. Guibas. 1995b. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. 419–428.
- E. Veach and L. J. Guibas. 1997. Metropolis light transport. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. 65–76.
- P. Vévoda, I. Kondapaneni, and J. Křivánek. 2018. Bayesian Online Regression for Adaptive Direct Illumination Sampling. *ACM Trans. Graph. (TOG)* 37, 4 (07 2018), 125:1–12.
- J. Vorba, J. Hanika, S. Herholz, T. Müller, J. Křivánek, and A. Keller. 2019. Path Guiding in Production. In *ACM SIGGRAPH 2019 Courses*. 18:41–45.
- J. Vorba, O. Karlik, M. Šik, T. Ritschel, and J. Krivanek. 2014. On-line Learning of Parametric Mixture Models for Light Transport Simulation. *ACM Trans. Graph. (TOG)* 33, 4 (07 2014), 101:1–11.
- B. Walter, A. Arbree, K. Bala, and D. Greenberg. 2006. Multidimensional Lightcuts. *ACM Trans. Graph. (TOG)* 25 (07 2006), 1081–1088.
- B. Walter, S. Fernandez, A. Arbree, K. Bala, M. Donikian, and D. Greenberg. 2005. Lightcuts: a scalable approach to illumination. *ACM Trans. Graph. (TOG)* 24, 3 (07 2005), 1098–1107.
- B. Walter, P. Khungurn, and K. Bala. 2012. Bidirectional lightcuts. *ACM Trans. Graph. (TOG)* 31, 4 (07 2012), 59:1–11.
- Y. Wang, Y. Wu, T. Li, and Y. Chuang. 2021. Learning to Cluster for Rendering with Many Lights. *ACM Trans. Graph. (TOG)* 40, 6 (12 2021), 277:1–10.
- Y. Wu and Y. Chuang. 2013. VisibilityCluster: Average Directional Visibility for Many-Light Rendering. *IEEE transactions on visualization and computer graphics* 19 (09 2013), 1566–1578.