Vision, Modeling, and Visualization (2020)
J. Krüger, M. Niessner, and J. Stückler (Eds.)

Portal-Based Path Perturbation for Metropolis Light Transport

H. Otsu[1] and J. Hanika[1] and C. Dachsbacher[1]

[1]Karlsruhe Institute of Technology, Germany

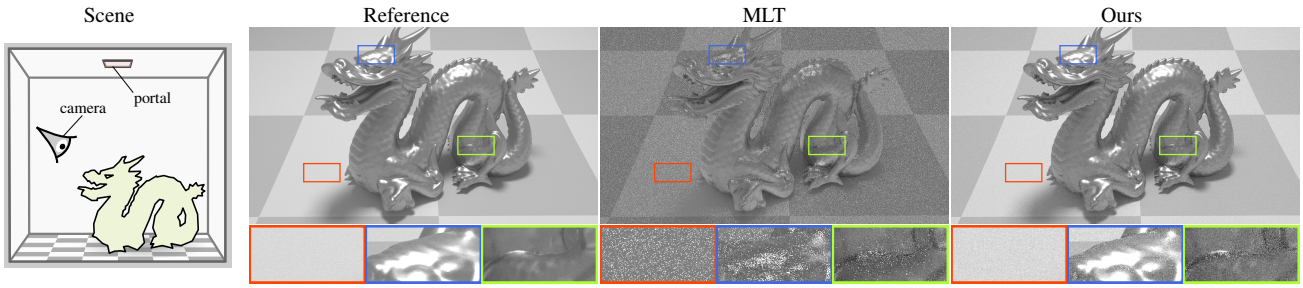


Figure 1: *Equal-time comparisons (30 minutes) of the* DRAGON IN BOX *scene in which the objects are located inside a box with a small opening surrounded by the scene geometry, illuminated by an environment light. We propose a novel path mutation technique for Metropolis light transport [VG97] utilizing the information of portals. The existing path mutation technique (center) cannot efficiently render this scene, since the edge perturbation is constrained on the path vertices and many attempts for the perturbation result in rejections due to the visibility change. Our approach (right) relaxes this constraint and instead fixes a point on the path edge allowing the perturbation to rotate the edge around this point. This enables better exploration of the Markov chain.*

Abstract

Light transport simulation in scenes with difficult visibility still remains a challenging problem. Markov chain Monte Carlo (MCMC) rendering is often employed for such configurations. It generates a sequence of correlated light transport paths by iteratively mutating the current state, a path, to another. Since the proposed path is correlated to the current path, MCMC can explore regions of the path space, also with difficult visibility, once they have been found. To improve the efficiency of the exploration, we propose a path mutation strategy making use of the concept of portals. Portals are user-defined objects in the scene to guide the sampling of the difficult visibility, which have been employed in the context of non-MCMC rendering. Our mutation strategy perturbs a path edge around the intersection point of the edge and the portal, instead of perturbing the edge by moving a path vertex as in the ordinary path mutation strategies. This reduces the probability for the proposed path being rejected due to changes in visibility.

CCS Concepts

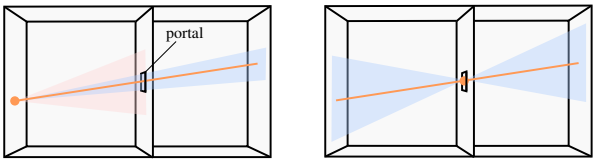
• ***Computing methodologies*** → ***Ray tracing;***

1. Introduction

Light transport simulation using Monte Carlo methods has become the de-facto standard tool for photorealistic rendering. The simulation is based on a stochastic generation (sampling) of light transport paths which connect the light sources and the sensor through scattering events in the scene. The ultimate goal is to find a general and robust sampling approach capable of rendering all scenes.

In this paper, we focus on one computing light transport in challenging geometric configurations. For instance, scenes where light transport paths are passing through a small opening in the geometry, such as an indoor scene illuminated only by light sources visible from the window, or scenes where a light source and the sensor are located in different parts of the scene. Developing a robust and efficient approach to handle this kind of difficult visibility is still an open problem.

Often such scenes are rendered with Markov chain Monte Carlo (MCMC) methods [ŠK20], a large family of methods originating from Metropolis light transport (MLT) by Veach and Guibas [VG97]. MCMC methods generate correlated sequences of paths (in a Markov chain) by iteratively mutating the current path to the next path, so that the generated paths are distributed according to a user-defined target distribution. Since the proposed path can depend on the current path, a mutation strategy can be designed to sample the proposed path in spatial proximity of the current path. This implies once a path with difficult geometric configuration is found, MCMC can explore this part of the state space without the need of finding it again via random sampling.



(a) existing path perturbation (b) portal-based path perturbation

Figure 2: *(a) Existing path perturbation and (b) Our portal-based perturbation. Previous work is limited to small step sizes (blue region) since the mutation rotates a path edge around a fixed vertex (a, orange point). As a result, the perturbed paths are likely to be rejected due to the visibility change (red region). Our proposed portal-based perturbation mutates the path by rotating the edge around the intersection point with a portal (b, orange point), which results less paths being rejected due to occlusion and thus in better exploration.*

Unfortunately this means that for small geometric openings such as small windows or even key holes, the perturbed paths can only deviate minimally from the current path when using existing mutation strategies [VG97, JM12, KHD14]. This is because they perturb path edges by keeping one vertex fixed (Fig. 2, left). For instance, Veach's lens perturbation in MLT perturbs the primary edge from the sensor fixing the point on the sensor, that is, it rotates the edge around the vertex on the sensor. Such mutation strategies cannot efficiently handle the case where the path edge is passing through a small opening surrounded by the geometry, because the rotation is *locked* by the position of the path vertex. This problem also occurs even when the mutation is adapted to geometry [OHHD18], as long as we stick to the rotation around a path vertex.

We propose a novel path mutation technique to address this problem, which enables perturbations based on arbitrary point on the path edge. Specifically, our mutation technique rotates the path edge around the intersection point of the edge and the *portal* (Fig. 2, right). A portal is a user-defined object typically introduced to guide the sampling through difficult visibility. Portals have been employed in the context of rasterization and Monte Carlo path tracing [Jon71, LD05, BNJ15]. Our approach introduces portal mutations into MCMC methods and can achieve better exploration than previous mutation strategies by allowing the use of larger step sizes and by increasing the acceptance rates. Furthermore, it can easily be combined with other path mutation strategies which is important to handle a large variety of scene configurations. We demonstrate that our approach shows better performance for configurations where existing mutation techniques fail to efficiently mutate the path.

In summary, our main contributions are:

- A path mutation technique based on an edge perturbation around a point on the edge,
- The use of portals in the context of MCMC rendering, and
- An analysis of the advantages and the limitations of portal-based path mutation technique.

2. Related Work

2.1. MCMC Rendering

MCMC methods have been introduced to rendering by Veach [VG97] who named the method Metropolis light transport (MLT). It is based on the seminal Metropolis-Hastings (MH) algorithm [MRR*53, Has70]. The original MLT method takes a light transport path as the state of a Markov chain, and the mutation strategies generate new paths by altering the path vertices. Since the mutations operate directly on paths, they are called *path space* algorithms.

Various specialized path mutation strategies have been proposed since then. Jakob and Marschner [JM12] proposed a mutation strategy that can efficiently handle specular or highly glossy interactions. Kaplanyan et al. [KHD14] and Hanika et al. [HKD15] extended the approach by using a parameterization of transport paths using half-vectors. More recently, Otsu et al. [OHHD18] proposed a mutation technique which adapts to the geometry around a path by estimating the opening cone angle around a path edge. These approaches, however, rely on edge perturbation around a path vertex, which inherently suffers from a poor exploration in difficult geometric configurations. To alleviate the issue, our approach instead perturbs the edge by rotating it around an arbitrary point along this edge, simultaneously moving both end vertices of the edge.

Kelemen et al. [KSKAC02] proposed the so-called primary sample space MLT (PSSMLT) method, where the mutation is performed in the space of random numbers which are used to sample a path. This space of random numbers is called the *primary sample space* (PSS). Arbitrary path sampling techniques can be used to map random vectors to paths, e.g. path tracing [Kaj86] or bidirectional path tracing (BDPT) [LW93, VG95]. Hachisuka et al. [HKD14] augmented the PSS by one extra dimension so that the state can differentiate bidirectional sampling strategies. Various other MCMC rendering also depend on the primary sample space paradigm. For instance, Li et al. [LLR*15] and Luan et al. [LZBG20] utilize analytic derivatives to apply anisotropic mutation kernels in PSS, Rioux-Lavoie et al. [RLLG*20] proposed an adaptive mutation technique based on delayed rejection in PSS.

Some approaches attempted to improve the (image space) stratification of MCMC, for instance, via improved scalar contribution functions [HH10], distributing the contribution over the image plane [CTE05], designing mutation kernels adapting to a target importance function [SKS17], selectively initiating the MCMC process only for difficult light transport [BJ19], or separating MCMC processes into overlapping tiles [GWH20].

In this paper we only focus on path space mutation techniques. Approaches based on PSS are less efficient in handling difficult visibility than path space techniques [OKH*17], since they do not explicitly maintain valid paths connecting a light source and a camera. Thus they always need to retrace the path from scratch for every mutation. Note that although our approach works in path space, it can still be combined with PSS mutation strategies by utilizing recent work about merging the two state spaces [OKH*17, Pan17, BJNJ17].

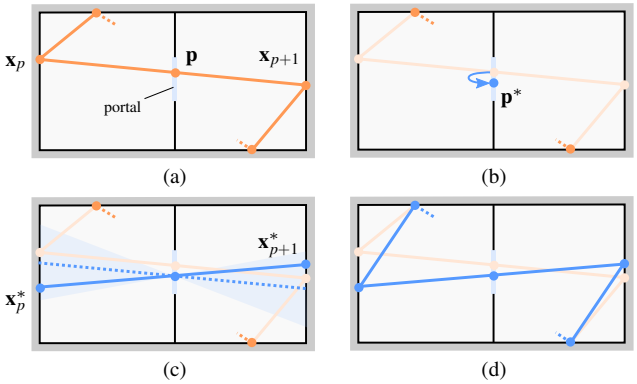


Figure 3: *Our portal-based path perturbation proceeds in four steps: (a) A path edge $\mathbf{x}_p\mathbf{x}_{p+1}$ intersects a portal at the point $\mathbf{p}$ and is selected for perturbation. (b) The position on the portal is perturbed and a new position $\mathbf{p}^*$ is obtained. (c) A new direction is sampled based on the current direction of the edge, obtaining the intersected points $\mathbf{x}^*_p$ and $\mathbf{x}^*_{p+1}$. (d) The proposed path is completed by connecting the new vertices with their neighbor vertices.*

2.2. Portals in Rendering

Portals are a classic concept employed early in computer graphics by Jones [Jon71] for hidden-line removal. The name *portal* has initially been introduced by Airey [Air90] and is now widely acknowledged in the graphics community. Later, portals were actively used in the context of occlusion culling for rasterization [TS91, LG95, AL99, LD05]. These approaches focus on determining the primary visibility seen through the portal given a viewpoint, such as a part of the room seen through a door.

Applications of portals in the context of Monte Carlo rendering have been developed as next event estimation techniques for path tracing [Kaj86]. Since a portal is typically modeled as a polygonal shape in the scene, importance sampling techniques for projected polygons/triangles [Arv95, UnFK13] can be directly used to sample a direction towards a portal. Bitterli et al. [BNJ15] developed a joint importance sampling technique for an environment map visible through a portal. More recently and concurrently, Ogaki [Oga20] introduced the generalized portals, e.g., defined by arbitrary polygon meshes or by textures, in the context of Monte Carlo rendering. Our approach for the first time leverages portals to design a path mutation strategy in the context of MCMC rendering.

3. Background

3.1. Path Integral

Light transport is often described by the path integral formulation [Vea98] where the pixel intensity I_j for the j-the pixel is defined as

$$I_j = \int_{\mathcal{P}} f_j(\bar{x})\,\mathrm{d}\mu(\bar{x}). \tag{1}$$

Here the integral is defined over the measurement contribution function $f_j(\bar{x})$ with respect to the product area measure μ. The domain $\mathcal{P}$ of the integral is the *path space* representing the set of all possible paths of arbitrary lengths. More specifically, the path space can be written as $\mathcal{P} = \cup_{k=2}^{\infty}\mathcal{P}_k$, where $\mathcal{P}_k$ is a set of paths of length $k-1$. An element of the path space is called a *path*. A path $\bar{x} \in \mathcal{P}$ is defined as a sequence of points $(\mathbf{x}_1, \ldots, \mathbf{x}_k)$. Unlike Veach's original formulation, we use a convention that the path starts with a point on the sensor. That is, $\mathbf{x}_1$ is the point on the sensor, and $\mathbf{x}_k$ is the point on the light, and all other vertices are the locations where the scattering events occur. In the following discussion, for brevity, we will omit the dependency to the pixel index j from the terms.

3.2. Metropolis Light Transport

Our approach is formulated as a mutation technique for Metropolis light transport. MLT generates a sequence of correlated samples according to the Metropolis-Hastings algorithm [MRR*53, Has70]. The path space $\mathcal{P}$ is used as the state space and the mutation strategies operate on this space. If we consider the target distribution π, the generated chain of samples is distributed according to π/b, where $b = \int_{\mathcal{P}} \pi(\bar{x})\,\mathrm{d}\mu(\bar{x})$ is the normalization factor, which is estimated with independent MC sampling techniques. As a target distribution π, we take the luminance of the measurement contribution function f, that is, $\pi(\bar{x}) := l(f(\bar{x}))$, where l is the luminance function.

Given the current path $\bar{x}_i$, at each step the MH algorithm first proposes a tentative path $\bar{x}^*$ by sampling from the transition kernel T, i.e., $\bar{x}^* \sim T(\cdot \mid \bar{x}_i)$. The proposed path is probabilistically either accepted as the new state $\bar{x}_{i+1}$, or rejected and the current state remains same, according to the *acceptance probability*. That is,

$$\bar{x}_{i+1} = \begin{cases} \bar{x}^* & \text{with probability } \min(1, a) \\ \bar{x}_i & \text{otherwise,} \end{cases} \tag{2}$$

where

$$a = a(\bar{x}^* \mid \bar{x}_i) = \frac{\pi(\bar{x}^*)T(\bar{x}_i \mid \bar{x}^*)}{\pi(\bar{x}_i)T(\bar{x}^* \mid \bar{x}_i)} = \frac{l \circ R(\bar{x}_i \mid \bar{x}^*)}{l \circ R(\bar{x}^* \mid \bar{x}_i)}, \tag{3}$$

where the function $R(\bar{y} \mid \bar{x}) := \frac{f(\bar{y})}{T(\bar{y}|\bar{x})}$ is a convenience function.

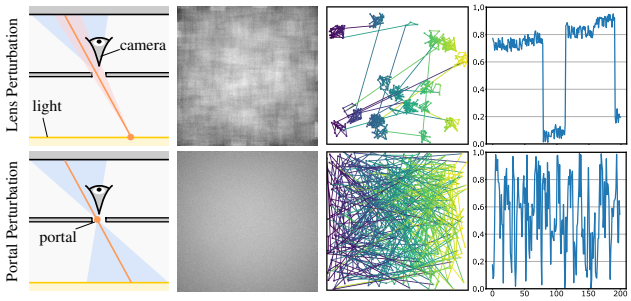
Given the initial state, this process generates a sequence of samples $\{\bar{x}_i\}_{1,\ldots,N}$ by N number of mutations, which can be used to estimate the path integral as

$$I \approx \hat{I} = \frac{1}{N}\sum_{i=1}^{N} \frac{f(\bar{x}_i)}{\pi(\bar{x}_i)/b} = \frac{b}{N}\sum_{i=1}^{N} \frac{f(\bar{x}_i)}{\pi(\bar{x}_i)}. \tag{4}$$

3.3. Path Perturbation

Veach and Guibas [VG97] proposed several path perturbation techniques, such as lens perturbation or caustic perturbation, in order to support local exploration of a specific path configuration.

Unlike the independent case in bidirectional path tracing [VG95], the path perturbation allows us to keep a part of the the path structure, and the direction can be taken from a conditional distribution given the current direction. In general, most path perturbations are a three step process: a) selecting a path edge, b) perturbing subpaths, c) reconnecting perturbed subpaths.



(a) Scene (b) Rendering (c) Trajectory (d) Traceplot (y axis)

Figure 4: *Illustrative example of the proposed approach. (a) The scene consists of three planes: a diffuse reflector illuminated by an area light, through a small rectangular hole on the blocker plane. (b) Equal-time renderings using two techniques: lens perturbation (existing approach, top) and portal-based perturbation (proposed approach, bottom). Lens perturbation clearly suffers from correlation artifacts compared to the proposed portal-based perturbation. (c) Sample trajectory on the image plane consisting of 500 points, where each point is taken per 500 mutations. (d) Traceplot of the image-space samples for the y axis, up to 200 points. The width and height of the images are normalized to* $[0,1]$.

Given a path $\bar{x} = \mathbf{x}_1 \cdots \mathbf{x}_k$, in step a), the path perturbation strategy deterministically selects a connecting edge. This process corresponds to selecting the strategy index (s,t) of bidirectional path sampling, where s and t are the number of vertices in light/eye subpaths, respectively. The connecting edge separates the path into an eye subpath $\bar{z} = \mathbf{z}_1 \cdots \mathbf{z}_t$ and a light subpath $\bar{y} = \mathbf{y}_1 \cdots \mathbf{y}_s$, where $\bar{x} = \mathbf{z}_1 \cdots \mathbf{z}_t \mathbf{y}_s \cdots \mathbf{y}_1$. We note that a subpath can be empty, which corresponds the case where $s = 0$ or $t = 0$.

Step b) perturbs both subpaths separately into $\bar{z}^* = \mathbf{z}_1^* \cdots \mathbf{z}_t^*$ and $\bar{y}^* = \mathbf{y}_1^* \cdots \mathbf{y}_s^*$ by sampling from the transition kernels: $\bar{z}^* \sim T(\cdot \mid \bar{z}^*)$ and $\bar{y}^* \sim T(\cdot \mid \bar{y}^*)$. Note that the path perturbation does not change the path length, so the proposed subpaths also have the same number of vertices as the original subpaths. Also note that some of the path vertices can remain the same as the original vertices. After the perturbation of the subpaths, in step c), the endpoints of the subpaths are connected and the proposed path is completed: $\bar{x}^* = \mathbf{z}_1^* \cdots \mathbf{z}_t^* \mathbf{y}_s^* \cdots \mathbf{y}_1^*$. If the connecting vertices are not endpoints, visibility must be checked to construct a valid path.

The transition kernels for the subpaths are composed of (multiple) mutation kernels for path directions. For instance, lens perturbation perturbs the primary edge from the camera. In this paper, following the suggestion by Veach and Guibas, we use the truncated reciprocal distribution to mutate directions. It has the following PDF in solid angle measure:

$$p_\sigma(\omega(\theta,\phi) \mid \omega_0) = \begin{cases} \frac{1}{C} \cdot \frac{1}{\theta \sin\theta} & \text{if } \theta \in [r_{\min}, r_{\max}] \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where C is the normalization constant, and the perturbed direction ω is parameterized by the spherical coordinates (θ,ϕ) in the local frame centered around the current direction ω_0.

3.4. Discussion

Path perturbations can be seen as correlated bidirectional path sampling, where each subpath must be sampled by tracing the path similarly to bidirectional path tracing, sequentially by perturbing the directions and extending the edges. This means that the perturbation of the path only consists of successively perturbing outgoing directions from vertices. As we discussed in the introduction, this common type of perturbation might be severely constrained by surrounding geometries around an edge, since the center of rotation (path vertices) can be distant from the opening.

4. Portal-Based Path Perturbation

4.1. Overview

We propose a path perturbation technique enabling edge perturbations with portals (Fig. 3). The main idea is to utilize a point on the edge as the center of rotation, allowing the edge to pivot around a point on the portal. Compared to existing path perturbation techniques which use a path vertex as a center of rotation, our approach is less likely to be rejected because of occlusion, which results in better exploration of the path space.

In this section, we will first discuss the representation of portals (Sec. 4.2). We will then explain the process of sampling a proposal path (Sec. 4.3), followed by the computation of the acceptance probability (Sec. 4.4), where we introduce the idea of considering the mutation process in an extended path space to properly account for transition probabilities. We will also discuss topics related to the implementation of the approach (Sec 4.5).

Illustrative Example. Fig. 4 shows an illustrative example to highlight which kind of configurations benefit from our approach. Here we consider a simple scene with three parallel planes (Fig. 4 (a)). The plane on the bottom is an area light, and the plane on the top is a diffuse reflector. The other plane is located in the middle between these planes, and has a hole in the center. The middle plane is large enough to block the light transport between the reflector and the light source, constraining the light to pass only through the small hole. The camera is facing toward the reflector.

We compare equal-time renderings with two different approaches: 1) lens perturbation [VG97] as a representative example of existing path perturbation techniques, and 2) our portal-based perturbation. Both approaches are configured with the same mutation size, combined with the bidirectional mutation to guarantee global exploration with the same selection probability. The portal is set up to cover the hole in the middle plane. Fig. 4 (b) shows that the lens perturbation severely suffers from correlation artifacts (top), compared to the portal-based perturbation (bottom). Although the primary edge from the camera can perturb relatively strongly (top, blue region), the mutation with the lens perturbation is constrained by the connecting edge (top, red region). On the other hand, the portal-based perturbation can freely move the second edge since the perturbation rotates the edge around the point on the portal (bottom, blue region). This behavior is also underlined by the sample trajectory on the image plane (Fig. 4 (c)), and the traceplot of the samples (Fig. 4 (d)). We can observe that the lens perturbation cannot freely explore in image space and the Markov chain tends to be stuck in small regions.

4.2. Portal Representation

We define portals as rectangular quad meshes in the scene, although technically we can assign arbitrary-shaped meshes as portals. Of course, the portal meshes are separated from the other geometries when it comes to ray intersection tests for visibility computation. Portal meshes are only intersected in the path proposal and acceptance ratio computation. We use a ray-quad intersection algorithm similar to the Möller–Trumbore algorithm [MT97].

Note that our approach only requires intersection tests against portals and does not depend on any other types of queries such as sampling points on the portals' surfaces. This means we do not need a special acceleration structure (similar to light hierarchies for next event estimation) for selecting and sampling a point on a portal mesh.

4.3. Path Proposal

Given a current path $\bar{x} = \mathbf{x}_1 \cdots \mathbf{x}_k$, the path proposal process starts by finding the initial path edge which intersects a portal in the scene. If a path intersects multiple portals, the edge closest to the sensor is selected. We denote the selected edge by $\mathbf{x}_p\mathbf{x}_{p+1}$, where p ($p \geq 1$) is the index of the vertex closer to the sensor, and the point on the portal by $\mathbf{p}$.

The perturbation of the selected edge proceeds as follows (also illustrated in Fig. 3): First we perturb the point on the portal $\mathbf{p}^* \sim T_A(\cdot \mid \mathbf{p})$, where T_A is the mutation kernel according to the area measure. The selection of the kernel is discussed later. Next, a new ray direction ω^* from $\mathbf{p}^*$ is sampled. The direction is sampled from the truncated reciprocal distribution similar to the lens perturbation: $\omega^* \sim T_{\sigma^\perp}(\cdot \mid \omega_{\mathbf{x}_p \to \mathbf{x}_{p+1}})$, where the base direction is taken from the current edge $\mathbf{x}_p\mathbf{x}_{p+1}$.

Next we compute the intersection with the two rays $(\mathbf{p}^*, -\omega^*)$ and $(\mathbf{p}^*, \omega^*)$ with the scene, which are assigned as the perturbed vertices $\mathbf{x}_p^*$ and $\mathbf{x}_{p+1}^*$ respectively. Unless $\mathbf{x}_p^*$ is on the sensor ($p \geq 2$), we check the visibility between $\mathbf{x}_{p-1}$ and $\mathbf{x}_p^*$. Similarly, we check the visibility between $\mathbf{x}_{p+1}^*$ and $\mathbf{x}_{p+2}$, unless $\mathbf{x}_{p+1}^*$ is on the light source ($p \leq k-2$). In either case, if any occlusion is found, then the proposal is immediately rejected. Otherwise, the final proposed path is then constructed as $\bar{x}^* = \bar{z}'\mathbf{x}_p^*\mathbf{x}_{p+1}^*\bar{y}'$, where

$$\bar{z}' = \begin{cases} \mathbf{x}_1 \cdots \mathbf{x}_{p-1} & p \geq 2 \\ \varnothing & \text{otherwise,} \end{cases} \quad \bar{y}' = \begin{cases} \mathbf{x}_{p+2} \cdots \mathbf{x}_k & p \leq k-2 \\ \varnothing & \text{otherwise.} \end{cases} \tag{6}$$

4.4. Acceptance Probability

Computing the Jacobian: A Problem. We next derive the acceptance probability for the mutation strategy introduced in the previous section. To compute this probability, we need to compute the transition probability in the same measure in which the original integral is defined. Our mutation strategy uses two mutation kernels: one for perturbing a point on the surface T_A and one for perturbing the direction $T_{\sigma^\perp}$, which must be converted to the area measure around the points $\mathbf{x}_p^*$ and $\mathbf{x}_{p+1}^*$. The conversion is done by multiplying the Jacobian:

$$T(\mathbf{x}_p^*\mathbf{x}_{p+1}^* \mid \mathbf{x}_p\mathbf{x}_{p+1}) = T_A(\mathbf{p}^*)T_{\sigma^\perp}(\omega^*)\left|\frac{d\sigma^\perp dA_*}{dA_p dA_{p+1}}\right|, \tag{7}$$

where we explicitly denote the area measure around each point by index, e.g., for $\mathbf{p}^*$ by A_*. For brevity, in the following discussion we denote the transition probabilities by $T_A(\mathbf{p}^*) \equiv T_A(\mathbf{p}^* \mid \mathbf{p})$ and by $T_{\sigma^\perp}(\omega^*) \equiv T_{\sigma^\perp}(\omega^* \mid \omega_{\mathbf{x}_p \to \mathbf{x}_{p+1}})$.

However, computing the Jacobian $\left|\frac{d\sigma^\perp dA_*}{dA_p dA_{p+1}}\right|$ is not obvious, since the sampling process changes both endpoints of the edge $\mathbf{x}_p^*\mathbf{x}_{p+1}^*$ simultaneously, and depends on the point on the portal which is also a sampled quantity. In other words, it is not obvious to how to interpret the change of the density according to the area measures around $\mathbf{x}_p^*$ and $\mathbf{x}_{p+1}^*$ through the mapping.

Extended Path. To work around this issue, we propose to use an *extended path*. We define the extended path $\bar{x}_+$ as a path which is augmented by the vertex on the portal. In our case, $\bar{x}_+ = \bar{z}'\mathbf{x}_p\mathbf{p}\mathbf{x}_{p+1}\bar{y}'$ since the portal always intersects the edge $\mathbf{x}_p\mathbf{x}_{p+1}$. We can naturally consider the aforementioned mutation strategy as a strategy applied to the extended path to perturb the vertices $\mathbf{x}_p\mathbf{p}\mathbf{x}_{p+1}$. Since the portal itself has no specific interaction, we can consider the scattering event at the vertex $\mathbf{p}$ being *pass-through* scattering, which does not change the incoming ray direction. This scattering can be modeled by the following BSDF:

$$f_s(\mathbf{x}, \omega_o, \omega_i) = \delta_{\sigma^\perp}(H(\omega_o, \omega_i)), \tag{8}$$

where $\delta_{\sigma^\perp}$ is the Dirac delta function defined in projected solid angle measure: $\int_\Omega \delta_{\sigma^\perp}(H(\omega_o, \omega))d\sigma^\perp(\omega) = 1$. $H\left(\omega, \omega'\right) = \omega + \omega'$ is the unnormalized half vector. Note that the idea behind pass-through scattering is conceptually the same as a null scattering being discussed in the rendering of heterogeneous volumes [MGJ19].

The mapping between the original path $\bar{x}$ and the extended path $\bar{x}^+$ is a bijection. To compute the acceptance probability, we can therefore use the extended path instead of the original path. Similar discussions have been conducted for the mapping between primary sample space and path space [OKH*17, Pan17, BJNJ17].

Mutating the Extended Path. The mutation strategy for the extended path can be described by the following steps: 1) sampling of the point $\mathbf{p}^*$ on the portal, 2) perturbing the direction ω^+ and finding $\mathbf{x}_{p+1}^*$, 3) sampling direction ω^- according to the BSDF at $\mathbf{p}^*$ as ω^+ being the incoming direction: $\omega^- \sim p_{\sigma^\perp}(\cdot \mid \omega^+) = \delta_{\sigma^\perp}(H(\omega^+, \cdot))$, and finding $\mathbf{x}_p^*$. By definition, it is simply a deterministic operation to change the sign of the direction: $\omega^- = -\omega^+$. 4) Lastly the proposed path is assembled: $\bar{x}_+^* = \bar{z}'\mathbf{x}_p^*\mathbf{p}^*\mathbf{x}_{p+1}^*\bar{y}'$. The mutation on the original path handles the third step implicitly.

Therefore, by a change of variables, the transition probability of the path vertices $\mathbf{x}_p\mathbf{p}\mathbf{x}_{p+1}$ can be written as

$$T(\mathbf{x}_p^*\mathbf{p}^*\mathbf{x}_{p+1}^* \mid \mathbf{x}_p\mathbf{p}\mathbf{x}_{p+1}) = T_A(\mathbf{p}^*) \cdot T_{\sigma^\perp}(\omega^+) \underbrace{\left|\frac{d\sigma^\perp}{dA_{p+1}}\right|}_{G(\mathbf{p}^*, \mathbf{x}_{p+1}^*)} \cdot \underbrace{p_{\sigma^\perp}(\omega^- \mid \omega^+)}_{\delta_{\sigma^\perp}(H(\omega^+, \omega^-))} \underbrace{\left|\frac{d\sigma^\perp}{dA_p}\right|}_{G(\mathbf{p}^*, \mathbf{x}_p^*)}, \tag{9}$$

where the Jacobians in the equation are geometry factors [Vea98]. Therefore, the acceptance probability can be computed by the following R function:

$$R(\bar{x}^*_+ \mid \bar{x}_+) = \frac{f(\bar{x}^*)}{G(\mathbf{x}^*_p, \mathbf{x}^*_{p+1})} \cdot \frac{G(\mathbf{p}^*, \mathbf{x}^*_p) f_s(\mathbf{x}^*_p, \mathbf{p}^*, \mathbf{x}^*_{p+1}) G(\mathbf{p}^*, \mathbf{x}^*_{p+1})}{T(\mathbf{x}^*_p \mathbf{p}^* \mathbf{x}^*_{p+1} \mid \mathbf{x}_p \mathbf{p} \mathbf{x}_{p+1})} = \frac{f(\bar{x}^*)}{T_A(\mathbf{p}^*) T_{\sigma^\perp}(\omega^+) G(\mathbf{x}^*_p, \mathbf{x}^*_{p+1})}, \tag{10}$$

where $f(\bar{x}^*)$ is the measurement contribution function for the *original* path. Note how the function takes a simple form thanks to some terms canceling out. Here we use the identity

$$\frac{f(\bar{x}_+)}{f(\bar{x})} = \frac{G(\mathbf{p}, \mathbf{x}_p) f_s(\mathbf{x}_p, \mathbf{p}, \mathbf{x}_{p+1}) G(\mathbf{p}, \mathbf{x}_{p+1})}{G(\mathbf{x}_p, \mathbf{x}_{p+1})}. \tag{11}$$

Eq. 10 can be further simplified if the transition kernels are symmetric. If so, the equation is simply written as $R(\bar{x}^*_+ \mid \bar{x}) = f(\bar{x}^*)/G(\mathbf{x}^*_p, \mathbf{x}^*_{p+1})$. Note that Eq. 10 implies the Jacobian $\left| \frac{d\sigma^\perp dA_*}{dA_p dA_{p+1}} \right|$ in Eq. 7 is in fact equal to $G(\mathbf{x}^*_p, \mathbf{x}^*_{p+1})$.

4.5. Implementation

Our approach can be implemented easily in any renderer supporting path space MLT. In this section, we will discuss implementation details.

Perturbation on the Portal. In our implementation, we used a *fixed* strategy for the proposal distribution T_A, which does not change the current intersected position on the portal at all. That is, the proposal distribution is $T_A(\mathbf{p}^* \mid \mathbf{p}) = \delta_A(\mathbf{p}^* - \mathbf{p})$. This strategy is simple to implement and parameter-free. The change of the position on the portal relies on other combined mutation strategies. An exploration and analysis of different design choices for locally perturbing the position on the portal is an interesting venue for future work.

Optimizing Directional Perturbation. The directional perturbation is defined so that the direction in the same face as the direction of the secondary edge $\mathbf{x}_p \mathbf{x}_{p+1}$, that is, $(\omega^* \cdot \omega_{\mathbf{x}_p \to \mathbf{x}_{p+1}}) > 0$. This assumption improves the performance of the mutation because it can lower the possibility of the connection failure between $\mathbf{x}^*_{p+1}$ and $\mathbf{x}_{p+2}$ or between $\mathbf{x}^*_p$ and $\mathbf{x}_{p-1}$.

Early Rejection. Our implementation of the mutation techniques uses *early rejections* of proposals to avoid unnecessary computation when it is obvious that the proposal will be rejected irrespective of the continuation of the execution.

Our approach can also utilize typical early rejections in the proposal, for instance when the perturbed direction does not intersect any scene geometry. Also, if the endpoint is associated with degenerate geometry (e.g., pinhole camera), the intersection tests from the point $\mathbf{p}^*$ against the degenerated geometry always fails. This means in this case the mutation is always rejected. We thus utilize this information to limit the range of p to avoid this case.

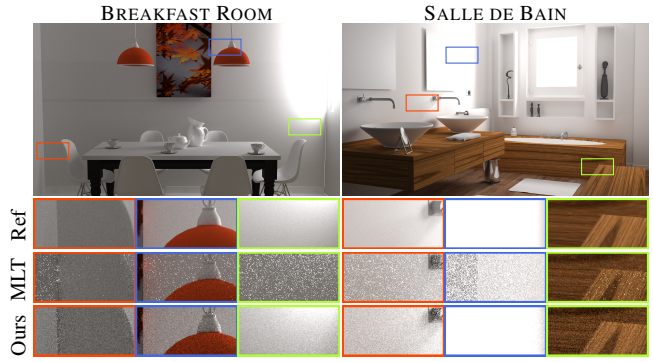


Figure 5: *Equal-time comparisons (30 minutes) of indoor scenes illuminated by an environment light through portals. We compare two approaches: MLT (middle, bidirectional mutation + lens perturbation) and ours (bottom, bidirectional mutation + portal/lens perturbations). Both are configured so that the global exploration (bidirectional mutation) and the local exploration (portal/lens perturbations) have equal strategy selection probabilities. Since lens perturbations cannot efficiently handle the edge directly connected to the environment light, there are severe fireflies caused by the poor exploration of the Markov chain. Our approach efficiently mutates this edge via portal perturbation resulting in better exploration. Our supplemental material includes full-image comparisons.*

Selecting Strategies. MLT can combine multiple mutation strategies in a single execution. Given a current state, a set of mutation strategies are selected according to the *suitability* of the mutation, and the mutation strategy for the current step is selected probabilistically according to the weights. If the path can be mutated by a strategy, the suitability is one. If not, the suitability is zero. In our case, the suitability is assigned to one when the path intersects with portals. If the path does not intersect any portals, the reverse transition probability $T(\bar{x} \mid \bar{x}^*)$ is always zero, since there is no way to sample the path without intersecting a portal with the technique.

5. Results

Setup. We implemented our proposed approach as a mutation strategy of MLT in an open source rendering framework [Ots19]. All experiments with time measurements have been carried out on a machine with an Intel Core i7-6700 CPU at 3.4 GHz using 8 threads. The reference images are rendered with more than 4 hours of computation time with bidirectional path tracing using a machine with 8-sockets Intel Xeon E7-8867 v3 CPU at 2.5 GHz using 256 threads. We configured the maximum path length to 9 for all scenes and used the same parameters for the truncated reciprocal distribution $r_{\min} = 2^{-8}$ and $r_{\max} = 2^{-4}$. Our supplemental material includes comparisons with an interactive viewer.

Approaches. In our experiments we compared the standard *MLT* and *our* approach; both are based on the same implementation of MLT and only differ in the combinations of the mutation strategies. Further, both combine the bidirectional mutation for global exploration and different path perturbation techniques for local exploration. We assigned equal strategy selection probabilities for the bidirectional mutation and the local path perturbation techniques.

As discussed in Sec. 4.5, the actual selection probability is obtained according to the suitabilities of the strategies for the current state. As local path perturbation strategy, *MLT* uses lens perturbation, which we selected to be representative of an existing path perturbation technique. On the other hand, *Ours* combines lens perturbation *and* our proposed portal perturbation with equal selection probabilities. That is, half of the mutations that would be done using lens perturbations in *MLT* are replaced by our approach. We combined two perturbation techniques because the portal perturbation does not explicitly handle the mutation of primary rays, which is crucial to image space exploration.

Equal-Time Comparisons. Fig. 1 and Fig. 5 show equal-time comparisons between *MLT* and *Ours*; all images have been rendered in 30 minutes. We conducted our experiments with three scenes, all of which are interior scenes illuminated by an environment light through a small opening space surrounded by scene geometry, which we configured as portals. Although we focus on the scene with an environment light, our approach is not limited to this lighting model. An example with the portal illuminated by indirect lighting is exhibited in the discussion (Sec. 6).

In all scenes, *MLT* has problems handling the direct illumination because the lens perturbation cannot mutate the edge directly connected to the environment light. As a result, the approach generates severe noise which is apparent as fireflies. Our portal perturbation can efficiently handle this case, which results in superior rendering performance.

Fig. 6 shows the pixel-wise error distribution using the relative root mean square error (rRMSE) as an error metric. In addition to the error distribution, we also calculated the rRMSE for the entire image, which is shown in the top-left corner. This figure also shows the effectiveness of our approach.

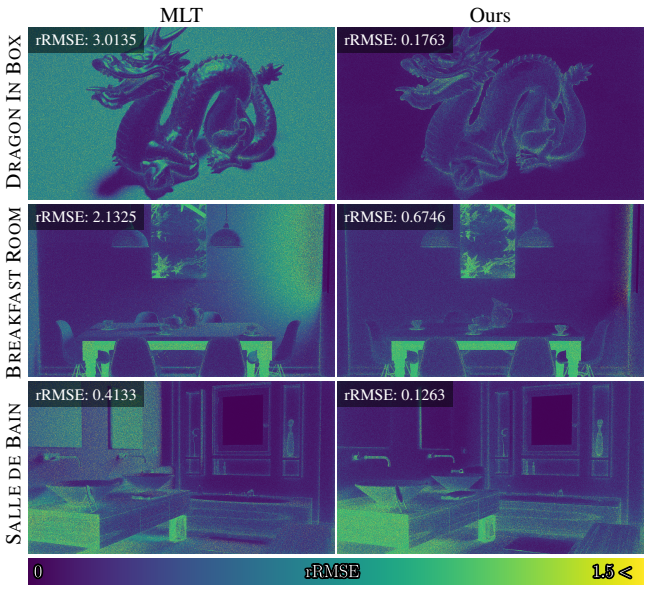


Figure 6: *Comparison of the error distributions in three scenes (*Dragon In Box*,* Breakfast Room*, and* Salle de Bain*). As error metric we used the relative root mean square error (rRMSE). The images show the pixel-wise rRMSE to visualize the error distributions; the single value rRMSE for the entire image is shown in the top-left corner. We can observe that our approach achieves lower error, especially in parts of the scenes directly illuminated from the light passing through the portal.*

6. Discussion

Comparison with Bitterli et al. Fig. 7 shows a comparison with the approach by Bitterli et al. [BNJ15] (PortalPT). The approach is implemented as a next event estimation of path tracing [Kaj86] and combined with ordinary BRDF sampling via multiple importance sampling [VG95]. We chose two scenes to highlight the characteristics of the two approaches. Both scenes are illuminated with a constant-luminance environment light, and differ in the geometry blocking the illumination from the interior (Fig. 7, left).

PortalPT performs better than our approach for direct lighting configuration (Fig. 7, top), where the object is directly illuminated from the environment light through the single portal; this is due to the simplicity of the base algorithm (path tracing). On the other hand, our approach can handle more general and difficult scene configurations, e.g. where the object is illuminated *indirectly* from illuminated surfaces using two portals (Fig. 7, bottom). The portal above the object is covered by another blocking geometry and thus the light passing through the portal can only stem from indirectly illuminated surfaces. We configured another portal on the opening of the covering geometry. In this case, our approach performs better than PortalPT, since their approach only considers the edge directly connected to the environment light. Although PortalPT can sample the last edge connected to the environment light efficiently, it is only effective if the path passes through the another small opening connected to the room containing the object and the camera, which results in lower effective samples.

Also note that our approach is agnostic to the light models (e.g., area light, directional light), while PortalPT is limited to environment lights. PortalPT needs to precompute the warped environment map for each portal, which limits the applicability to large number of portals due to the memory requirement. Also, selecting portals from a query point entails the similar difficulties as many light rendering [CEK18], since selecting non-occluded portals is not trivial.

Limitations. Fig. 8 depicts a limitation of our approach with the Ajar Door scene. The scene is only illuminated by the light passing through an ajar door. Here we defined the portal to cover the extent of the door frame. In this configuration our portal perturbation does not pay off, because the visible part of the scene from the sensor is mostly *indirectly* illuminated by the light passing through the portal. This means in this case the portal perturbation cannot mutate the primary edges and cannot achieve image space exploration, which must be processed by other mutation strategies. The same behavior can also be observed in the indirectly illuminated parts in Fig. 5. For instance, *MLT* is superior to *Ours* in the back of the chair in the Breakfast Room scene.

Technically, decorrelating the indirect part of the path can still alleviate correlation artifacts in the image. Specifically for this scene,

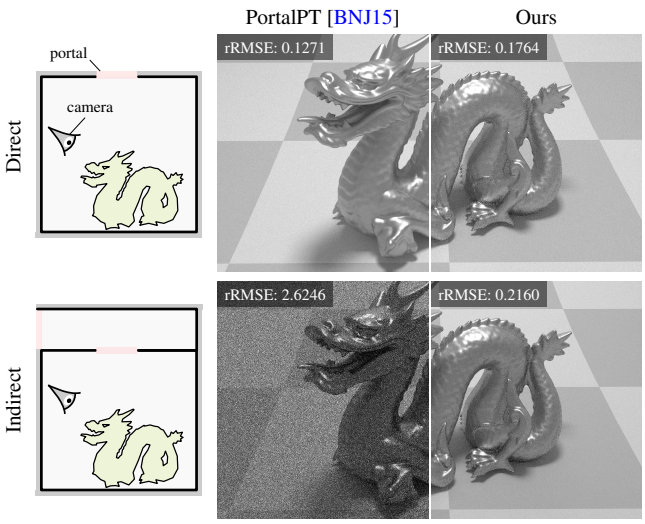


Figure 7: *Equal-time comparison (30 minutes) to the approach by Bitterli et al. [BNJ15] (PortalPT). PortalPT is tailored for the scenes where the object is illuminated directly from the environment light through the portal (top). Note how our approach can efficiently render the scene with more general and difficult lighting configurations where the object is illuminated by the indirectly illuminated surfaces through the portal (bottom); PortalPT cannot efficiently find the edge passing through the portal connecting the rooms.*

however, the correlation artifacts are almost unnoticeable even only with lens perturbation, which implies there is only a small room for the portal perturbation to further benefit the performance by improving the correlations.

Future Work. Our portal-based perturbation cannot mutate purely specular interactions directly connected to the sensor, since the connection to the previous vertex on the eye subpath would always generate a path with zero contribution due to the delta function of the specular interaction. To resolve this issue, it would be an interesting future work to combine our approach with those connecting vertices over specular interactions [JM12, ZGJ20].

Our portal perturbation needs to be used with lens perturbation to improve the image space exploration. To further improve the exploration performance, it would be interesting to develop a mutation technique combining both approaches as a single mutation technique, where the primary ray and the path edge through portals are perturbed simultaneously.

If a path intersects multiple portals, our approach deterministically selects the closest portal to the eye vertex (Sec. 4.3). This is based on the heuristic that decorrelating the path segment closer to the camera is more effective, based on the observation that using lens perturbation alone already gives moderately good results in many scenes. Our approach, however, naturally extends to select different portals, which would be an good venue for the future work.

Our approach uses rectangular quad meshes as portals and we used only a small number of portals (1 or 2) in the experiments. Using the arbitrary-shaped portals (e.g., a mesh well fit to the opening between geometries), might improve the exploration performance since conservative placement of portals can result in many occluded paths. The extension and analysis of the performance, for instance, when the number of portals is large or the portal is associated to complex geometries (such as in [Oga20]), would be an interesting future work.

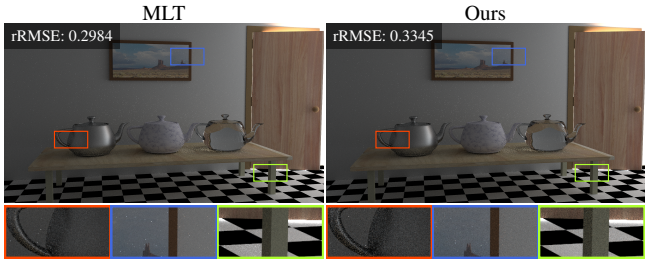


Figure 8: *Equal-time comparisons (30 minutes) of the* AJAR DOOR *scene which is only illuminated by the light coming through the ajar door. We set up a portal on the door frame connecting the two rooms. Unfortunately, rendering does not benefit from our portal perturbation in this scene, since the visible region of the scene is mostly illuminated indirectly by the light passing through the portal, which causes poor exploration on the image plane.*

A combination with the geometry-adaptive approach by Otsu et al. [OHHD18] to find an opening cone angle around a path edge would also be an interesting future work, since this cone centered around a path edge can be considered as a generalized portal. This could provide information to design adaptive portal-based perturbation techniques.

Currently the information of portals must be provided by users. To alleviate the burden of this additional requirement, it would be interesting if the shape and the placement of portals is generated automatically according to the scene geometry, or according to the information obtained through rendering.

7. Conclusion

We presented a novel path mutation strategy for Metropolis light transport, which perturbs a path edge around the intersection point of the path edge and a portal. Unlike existing path mutation techniques where the edge perturbation is constrained on the path vertices, our approach can perturb the edge itself while avoiding high rejection probabilities due to geometric occlusions. We demonstrated that our approach can improve the exploration performance of the Markov chain, especially in scene configurations where existing approaches fail to mutate paths efficiently.

Acknowledgements

We would like to thank the reviewers for their insightful comments. The AJAR DOOR scene was originally modeled by Miika Aittala, Samuli Laine, and Jaakko Lehtinen. We thank blendswap.com artist Wig42 for the BREAKFAST ROOM scene and nacimus for the SALLE DE BAIN scene. The model in the DRAGON IN BOX scene is courtesy of the Stanford Computer Graphics Laboratory. This work was funded by DFG grant DA 1200/8-1, project 405788923.

References

[Air90] AIREY J. M.: *Increasing Update Rates in the Building Walkthrough System with Automatic Model-Space Subdivision and Potentially Visible Set Calculations*. PhD thesis, 1990. 3

[AL99] ALIAGA D., LASTRA A.: Architectural walkthroughs using portal textures. *Proc. IEEE Visualization* (03 1999), 355–362. 3

[Arv95] ARVO J.: Stratified sampling of spherical triangles. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (1995), SIGGRAPH '95, p. 437–438. 3

[BJ19] BITTERLI B., JAROSZ W.: Selectively Metropolised Monte Carlo light transport simulation. 2

[BJNJ17] BITTERLI B., JAKOB W., NOVÁK J., JAROSZ W.: Reversible jump metropolis light transport using inverse mappings. *ACM Transactions on Graphics 37*, 1 (2017). 2, 5

[BNJ15] BITTERLI B., NOVÁK J., JAROSZ W.: Portal-masked environment map sampling. *Computer Graphics Forum (Proc. Eurographics Symposium on Rendering) 34*, 4 (2015). 2, 3, 7, 8

[CEK18] CONTY ESTEVEZ A., KULLA C.: Importance sampling of many lights with adaptive tree splitting. *Proc. ACM Comput. Graph. Interact. Tech. 1*, 2 (2018). 7

[CTE05] CLINE D., TALBOT J., EGBERT P.: Energy redistribution path tracing. *ACM Transactions on Graphics (Proc. SIGGRAPH) 24*, 3 (2005), 1186–1195. 2

[GWH20] GRUSON A., WEST R., HACHISUKA T.: Stratified Markov Chain Monte Carlo Light Transport. *Computer Graphics Forum (Proc. of Eurographics)* (2020). 2

[Has70] HASTINGS W. K.: Monte Carlo sampling methods using Markov chains and their applications. *Biometrika 57*, 1 (1970), 97–109. 2, 3

[HH10] HOBEROCK J., HART J. C.: Arbitrary importance functions for metropolis light transport. *Computer Graphics Forum 29*, 6 (2010), 1993–2003. 2

[HKD14] HACHISUKA T., KAPLANYAN A. S., DACHSBACHER C.: Multiplexed Metropolis light transport. *ACM Transactions on Graphics (Proc. SIGGRAPH) 33*, 4 (2014). 2

[HKD15] HANIKA J., KAPLANYAN A. S., DACHSBACHER C.: Improved half vector space light transport. *Computer Graphics Forum (Proc. Eurographics Symposium on Rendering) 34*, 4 (2015), 65–74. 2

[JM12] JAKOB W., MARSCHNER S.: Manifold exploration: a Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics (Proc. SIGGRAPH) 31*, 4 (2012). 2, 8

[Jon71] JONES C. B.: A new approach to the 'hidden line' problem. *The Computer Journal 14*, 3 (1971), 232–237. 2, 3

[Kaj86] KAJIYA J. T.: The rendering equation. *Computer Graphics (Proceedings of SIGGRAPH '86) 20*, 4 (1986), 143–150. 2, 3, 7

[KHD14] KAPLANYAN A., HANIKA J., DACHSBACHER C.: The natural-constraint representation of the path space for efficient light transport simulation. *ACM Transactions on Graphics (Proc. SIGGRAPH) 33*, 4 (2014), 1–13. 2

[KSKAC02] KELEMEN C., SZIRMAY-KALOS L., ANTAL G., CSONKA F.: A simple and robust mutation strategy for the Metropolis light transport algorithm. *Computer Graphics Forum 21*, 3 (2002), 531–540. 2

[LD05] LOWE N., DATTA A.: A new technique for rendering complex portals. *IEEE Transactions on Visualization and Computer Graphics 11* (02 2005), 81–90. 2, 3

[LG95] LUEBKE D., GEORGES C.: Portals and mirrors: Simple, fast evaluation of potentially visible sets. In *Symposium on Interactive 3D Graphics* (1995), pp. 105–106. 3

[LLR*15] LI T.-M., LEHTINEN J., RAMAMOORTHI R., JAKOB W., DURAND F.: Anisotropic Gaussian mutations for Metropolis light transport through Hessian-Hamiltonian dynamics. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 34*, 6 (2015), 209:1–209:13. 2

[LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bi-directional path tracing. In *Compugraphics '93* (1993), pp. 145–153. 2

[LZBG20] LUAN F., ZHAO S., BALA K., GKIOULEKAS I.: Langevin monte carlo rendering with gradient-based adaptation. *ACM Transactions on Graphics (Proc. SIGGRAPH) 39*, 4 (2020). 2

[MGJ19] MILLER B., GEORGIEV I., JAROSZ W.: A null-scattering path integral formulation of light transport. *ACM Transactions on Graphics (Proc. SIGGRAPH) 38*, 4 (2019). 5

[MRR*53] METROPOLIS N., ROSENBLUTH A. W., ROSENBLUTH M. N., TELLER A. H., TELLER E.: Equation of state calculations by fast computing machines. 1087–1092. 2, 3

[MT97] MÖLLER T., TRUMBORE B.: Fast, minimum storage ray-triangle intersection. *Journal of Graphics Tools 2*, 1 (1997), 21–28. 5

[Oga20] OGAKI S.: Generalized light portals. *Proc. ACM Comput. Graph. Interact. Tech. 3*, 2 (2020). 3, 8

[OHHD18] OTSU H., HANIKA J., HACHISUKA T., DACHSBACHER C.: Geometry-aware metropolis light transport. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 37*, 6 (2018), 278:1–278:11. 2, 8

[OKH*17] OTSU H., KAPLANYAN A. S., HANIKA J., DACHSBACHER C., HACHISUKA T.: Fusing state spaces for markov chain monte carlo rendering. *ACM Transactions on Graphics (Proc. SIGGRAPH) 36*, 4 (2017). 2, 5

[Ots19] OTSU H.: Lightmetrica – research-oriented renderer (version 3), 2019. https://github.com/lightmetrica/lightmetrica-v3. 6

[Pan17] PANTALEONI J.: Charted metropolis light transport. *ACM Transactions on Graphics (Proc. SIGGRAPH) 36*, 4 (2017). 2, 5

[RLLG*20] RIOUX-LAVOIE D., LITALIEN J., GRUSON A., HACHISUKA T., NOWROUZEZAHRAI D.: Delayed rejection Metropolis light transport. *ACM Transactions on Graphics 39*, 3 (2020). 2

[ŠK20] ŠIK M., KŘIVÁNEK J.: Survey of markov chain monte carlo methods in light transport simulation. *IEEE Transactions on Visualization and Computer Graphics 26*, 4 (2020), 1821–1840. 1

[SKS17] SZIRMAY-KALOS L., SZÉCSI L.: Improved stratification for metropolis light transport. *Computers & Graphics 68* (2017), 11 – 20. 2

[TS91] TELLER S., SEQUIN C.: Visibility preprocessing for interactive walkthroughs. *Computer Graphics (Proc. SIGGRAPH) 25*, 4 (1991), 61–68. 3

[UnFK13] UREÑA C., FAJARDO M., KING A.: An area-preserving parametrization for spherical rectangles. *Computer Graphics Forum (Proc. Eurographics Symposium on Rendering) 32* (2013), 59–66. 3

[Vea98] VEACH E.: *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford University, USA, 1998. AAI9837162. 3, 6

[VG95] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for Monte Carlo rendering. *Proc. SIGGRAPH '95* (1995), 419–428. 2, 3, 7

[VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. In *SIGGRAPH '97* (1997), pp. 65–76. 1, 2, 3, 4

[ZGJ20] ZELTNER T., GEORGIEV I., JAKOB W.: Specular manifold sampling for rendering high-frequency caustics and glints. *ACM Transactions on Graphics (Proc. SIGGRAPH) 39*, 4 (2020). 8