# Ensemble Metropolis Light Transport

THOMAS BASHFORD-ROGERS, University of the West of England, UK
LUÍS PAULO SANTOS, Universidade do Minho/INESC TEC, Portugal
DEMETRIS MARNERIDES and KURT DEBATTISTA, University of Warwick, UK

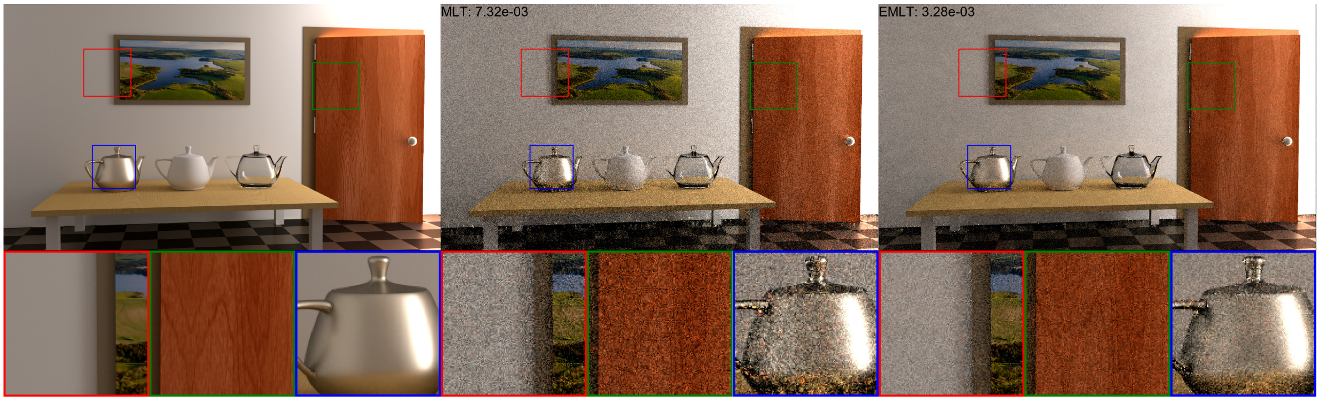


Fig. 1. The DOOR AJAR scene showing the reference on the left, Metropolis Light Transport in the centre, and Ensemble Metropolis Light Transport on the right, rendered at an equal sample count. The use of the ensemble to propose anisotropic transition kernels allows sampling to be adapted to the scene geometry and lighting information, leading to variance reduction as can be seen by lower MSE values shown at the top of each image.

This article proposes a **Markov Chain Monte Carlo** (**MCMC**) rendering algorithm based on a family of guided transition kernels. The kernels exploit properties of ensembles of light transport paths, which are distributed according to the lighting in the scene, and utilize this information to make informed decisions for guiding local path sampling. Critically, our approach does not require caching distributions in world space, saving time and memory, yet it is able to make guided sampling decisions based on whole paths. We show how this can be implemented efficiently by organizing the paths in each ensemble and designing transition kernels for MCMC rendering based on a carefully chosen subset of paths from the ensemble. This algorithm is easy to parallelize and leads to improvements in variance when rendering a variety of scenes.

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

Additional Key Words and Phrases: Light transport, MCMC, ensemble

**ACM Reference format:**
Thomas Bashford-Rogers, Luís Paulo Santos, Demetris Marnerides, and Kurt Debattista. 2021. Ensemble Metropolis Light Transport. *ACM Trans. Graph.* 41, 1, Article 5 (December 2021), 15 pages.
https://doi.org/10.1145/3472294

## 1 INTRODUCTION

Accurately rendering photorealistic imagery requires computing extremely large numbers of light paths in a virtual environment. While most research has been applied to traditional Monte Carlo estimators for rendering, **Markov Chain Monte Carlo** (**MCMC**) methods, such as **Metropolis Light Transport** (**MLT**) [Veach and Guibas 1997], have shown impressive capabilities for computing light transport efficiently, even in complicated scenes.

MCMC algorithms such as MLT generate a chain of paths that follow the distribution of the lighting in the scene. Each new path is generated by applying a transition kernel to the previous path, and probabilistically replacing the previous path with the new path. In the original application of MCMC to graphics [Veach and Guibas 1997], transition kernels were designed to either locally explore regions around an existing path, or to globally explore path space. While these transition kernels have been improved to consider local geometric or lighting information [Li et al. 2015; Otsu et al. 2018], the use of non-local information capturing a wider range of lighting can also be used to guide transition kernels. Such non-local information can be captured by path guiding methods, for example [Müller et al. 2017]; however, this comes at a precomputation and memory cost, and the cached distributions of lighting may not clearly map to transition kernels. Another approach is to exploit the fact that multiple light paths generated by MCMC

Authors' addresses: T. B.-Rogers, University of the West of England, Coldharbour Ln, Bristol BS16 1QY, UK; L. P. Santos, Universidade do Minho/INESC TEC, Campus de Gualtar, Braga 4710-057, Portugal; D. Marnerides and K. Debattista, University of Warwick, Gibbet Hill Road, Coventry CV4 7AL, UK.

0730-0301/2021/12-ART5 $15.00
https://doi.org/10.1145/3472294

algorithms will be distributed proportional to the lighting in the scene, and as such can be used to generate guided transition kernels.

This article proposes such a method that uses ensembles of light paths to guide mutations of existing paths. We name this approach **Ensemble Metropolis Light Transport** (**EMLT**). Crucially, these guided transition kernels do not need to be based on caching distributions in world space; they only require moderately sized ensembles in the low tens of thousands of paths, and can be combined to build families of mutation strategies. To summarize, the main contributions of this work are as follows:

- The introduction of EMLT, a method that guides sampling based on a complementary ensemble of transport paths.
- A family of adaptive and anisotropic proposal distributions for path mutations based on ensemble sampling.
- The use of a carefully chosen subset of paths from the ensemble to create guided transition kernels.
- Results showing improvement of EMLT over traditional approaches in a range of real-world scenes.

## 2 BACKGROUND AND RELATED WORK

This section introduces the relevant background theory of light transport, path guiding methods which exploit information about the radiance or importance distribution in the scene to reduce variance, and MCMC methods which can efficiently compute images in challenging scenes.

### 2.1 Light Transport

The path integral form of the rendering equation [Kajiya 1986] is given by [Hachisuka et al. 2014; Veach and Guibas 1997]

$$I_j = \int_{\mathcal{P}} h_j(\overline{x}) f(\overline{x}) d_\mu(\overline{x}), \qquad (1)$$

and states that the intensity $I_j$ at a pixel $j$ consists of the contribution $f(\overline{x})$ of light paths $\overline{x}$ weighted by a pixel filter $h_j(\overline{x})$. The domain of integration is the union of all possible path lengths $\mathcal{P} = \bigcup_{k=2}^{\infty} \mathcal{P}(k)$ where $\mathcal{P}(k)$ are all paths of length $k$. In this work, path vertices $x_0..x_k$ lie on the scene manifold $\mathcal{M}$, i.e., integration is with respect to the product area measure $\mu$, and are indexed starting from the light source $x_0$. The contribution of a path of length $k$ is defined as $f(\overline{x}) = Le(\overline{x}_0)G(\overline{x}_0 \leftrightarrow \overline{x}_1) \prod_{j=1}^{k-1} fr(\overline{x}_{j-1} \rightarrow \overline{x}_j \rightarrow \overline{x}_{j+1})G(\overline{x}_j \leftrightarrow \overline{x}_{j+1})$, where $Le$ is the emitted radiance, $G$ is the geometry term, and $fr$ is the **Bidirectional Reflectance Distribution Function (BRDF)**.

There are multiple ways of solving Equation (1), almost all relying on Monte Carlo estimation:

$$I_j \approx \frac{1}{N} \sum_{i=1}^{N} \frac{h_j(\overline{x}(i)) f(\overline{x}(i))}{p(\overline{x}(i))}, \qquad (2)$$

where $p(\overline{x}(i))$ denotes the **probability density function** (**pdf**) of sampling the $i$th path and is a product of probability densities for sampling each vertex to build up the path. This typically consists of sampling the sensor, lens, BRDFs and light sources. Ideally $p(\overline{x}(i)) \propto h_j(\overline{x}(i)) f(\overline{x}(i))$; however, this is typically not possible in practice. Therefore, distributions which approximate some components of $h_j(\overline{x}(i)) f(\overline{x}(i))$ are used (see Christensen and Jarosz [2016] for a survey of these methods).

### 2.2 Path Guiding

Most rendering techniques generate light paths incrementally by sampling the next vertex in a path given the previous vertex. Path guiding approaches build on traditional BRDF and cosine sampling to include information about incoming illumination or importance when generating samples. Most techniques cache a distribution that represents the incoming radiance or importance at a sparse set of locations in a scene, and query locations at runtime using a spatial data structure. Examples include 5D spatio-directional Trees [Lafortune and Willems 1995; Müller et al. 2017], 7D distributions [Pantaleoni 2020], use of various basis functions to store radiance at discrete points in the scene [Bashford-Rogers et al. 2012; Diolatzis et al. 2020; Herholz et al. 2016; Hey and Purgathofer 2002; Jensen 1995; Ruppert et al. 2020; Vorba et al. 2014], or using machine learning methods [Bako et al. 2019; Dahm and Keller 2017].

Path guiding has also been applied to sample partial or complete light paths. Approaches such as neural importance sampling [Guo et al. 2018; Müller et al. 2018; Zheng and Zwicker 2019] have learned a warping in **Primary Sample Space** (**PSS**) [Kelemen et al. 2002] which encodes the illumination distribution in PSS based on a small set of paths traced before rendering. However, as these approaches are designed to generate full paths, they face the curse of dimensionality and are more effective in lower dimensional scenarios such as importance sampling one bounce indirect lighting. A related approach to whole path importance sampling was proposed by Reibold et al. [2018] that selectively stores and samples distributions for high contribution paths which were unlikely to be sampled through BRDF sampling.

EMLT exploits information about the lighting distribution in the scene, and can use any of the distributions commonly used for path guiding to generate samples. However, our method does not require a spatial cache, and builds distributions on the fly from a small set of paths from a complementary ensemble (see Section 3).

### 2.3 MCMC

MCMC [Hastings 1970; Metropolis et al. 1953] techniques provide another approach to sample a space and were initially applied to rendering as MLT [Veach and Guibas 1997]. In MLT, sampling starts from an initial path $\overline{x}$, and then proposes a new path $\overline{x'}$ from a transition kernel $T(\overline{x} \rightarrow \overline{x'})$. The transition kernels have to satisfy certain properties in order for the chain to explore the state space: ergodicity meaning all states will be visited by the chain in a finite time, and aperiodicity meaning states will not get stuck in a loop. At the limit, these states are distributed according to a target distribution, $\frac{f}{b}$, where $b = \int_{\mathcal{P}} h_j(\overline{x}) f(\overline{x}) d_\mu(\overline{x})$ is a normalization constant. A scalar contribution function, $f^* : \mathbb{R}^S \mapsto \mathbb{R}$, is defined where $S$ are the spectra or color channels associated with evaluating $f(\overline{x})$. Then, based on the detailed balance condition $f^*(\overline{x})T(\overline{x} \rightarrow \overline{x'})a(\overline{x} \rightarrow \overline{x'}) = f^*(\overline{x'})T(\overline{x'} \rightarrow \overline{x})a(\overline{x'} \rightarrow \overline{x})$, the new state $\overline{x'}$ is probabilistically chosen to replace the previous state based on calculating an acceptance probability:

$$a(\overline{x} \rightarrow \overline{x'}) = min\left(1, \frac{f^*(\overline{x'})T(\overline{x'} \rightarrow \overline{x})}{f^*(\overline{x})T(\overline{x} \rightarrow \overline{x'})}\right). \tag{3}$$

This leads to a chain of light paths, each dependent only on the previously sampled light path, which explore the path space. In rendering, the scalar contribution function is typically chosen to be the luminance of the contribution of the path, but other functions can be chosen (see Gruson et al. [2016]; Hoberock and Hart [2010]). $b$ is typically estimated by a separate Monte Carlo estimator, and the initial state of the chain is generated through resampling a path from a small set of paths computed at startup. Then, the resulting Monte Carlo estimator is given by

$$I_j \approx \frac{b}{N} \sum_{i=1}^{N} \frac{h_j(\overline{x}(i)) f(\overline{x}(i))}{f(\overline{x}(i))} = \frac{b}{N} \sum_{i=1}^{N} h_j(\overline{x}(i)), \tag{4}$$

meaning that samples will be distributed according to the integrand. Equation (4) is typically evaluated over the image plane which allows for information about light transport to be shared between pixels, resulting in a significantly more efficient estimator. Many strategies can be designed such that many terms in the numerator and denominator in Equation (3) cancel, something which is especially important to remove the weak singularity in the geometry term. The variance reduction properties of this method also depends on the ability of the transition kernel to explore the state space.

Veach and Guibas [1997] proposed a series of transition kernels which were chosen to reduce variance for different types of light transport. Bidirectional mutations were designed to ensure ergodicity through deleting a randomly chosen series of vertices from $\overline{x}$, and replacing them with vertices generated through sampling the same pdfs used in a standard Monte Carlo estimator, i.e., BRDF and light source sampling. The remaining strategies, known as perturbations, were designed to explore sub spaces of path space given the state of the path. Lens perturbations explored image space by perturbing the position of the path vertex on the image plane, tracing a path through any specular interactions until a non-specular vertex is reached, then deterministically connecting to the unchanged light subpath. This connection leaves a geometry term associated with the deterministic connection when evaluating Equation (3). The caustic perturbation was designed to explore caustics through perturbing the outgoing direction for the first vertex on the caustic subpath, following the chain of specular interactions, and then deterministically connecting to the camera. The multi-chain perturbation explores specular-diffuse-specular paths through combining a perturbation on the lens with directional perturbations at each non-specular surface before deterministically connecting to the remaining light subpath.

There have been several extensions to the original MLT algorithm which have added or improved mutation strategies such as perturbations in participating media [Pauly et al. 2000], improved sampling of specular chains [Jakob and Marschner 2012], and perturbations in half vector space [Kaplanyan et al. 2014]. Kelemen et al. [2002] introduced mutations to light paths in PSS (PSSMLT). These mutations were improved by Hachisuka et al. [2014] who combined PSSMLT with Multiple Importance Sampling [Veach and Guibas 1995], Bitterli and Jarosz [2019] who detected and perturbed high variance paths in PSS, the use of delayed rejection by Rioux-Lavoie et al. [2020], the use of Hamiltonian Monte Carlo applied to rendering by Li et al. [2015] who used anisotropic Gaussian kernels generated from a path gradient, and Luan et al. [2020] who used the Metropolis-adjusted Langevin algorithm also based on the gradients of the path. Integration in both path space and PSS have been proposed [Bitterli et al. 2018; Otsu et al. 2017; Pantaleoni 2017] which allows path space mutations to be combined with PSS mutations. For further information, Šik and Křivánek [2018] provide a detailed survey of MCMC methods in rendering.

Closer to our work, adaptive perturbation sizes based on scene geometry were proposed by Otsu et al. [2018], which used cone tracing to estimate how large a perturbation could be based on the surrounding geometry of a path. This was applied starting at the camera, followed specular bounces if any, then traced one extra path vertex to form a perturbed path.

Other methods mutate a set of paths but do not directly use these to adapt transition kernels. Energy Redistribution Path Tracing [Cline et al. 2005] combined Path Tracing and MLT by creating many short chains whenever a path would be better explored by MCMC methods than standard Monte Carlo. Segovia et al. [2007] used Multiple-Try MCMC to generate paths for Instant Radiosity [Keller 1997], and Nimier-David et al. [2019] also proposed a Multiple-Try MCMC method suitable for vectorized instructions.

## 2.4 Ensemble MCMC Methods

The use of multiple paths have been used in rendering to reduce variance and better explore path space. These methods have largely focused on variants of parallel tempering, also known as Replica Exchange Monte Carlo [Swendsen and Wang 1986]. This uses multiple Markov Chains to explore different spaces, and uses a detailed balance preserving transition to swap chains between spaces. This was introduced to graphics by Kitaoka et al. [2009], and improved by Šik and Křivánek [2016] and Otsu et al. [2013]. These approaches have also been applied to progressive photon mapping [Hachisuka and Jensen 2011], the combination VCM/UPS with MCMC [Šik et al. 2016], and with stratified MCMC on the image plane [Gruson et al. 2020]. Hachisuka et al. [2014] also used a pool of chains of different lengths to sample path lengths proportional to their contribution.

Another related approach is to use **Population Monte Carlo** (**PMC**) [Cappé et al. 2004; Fan et al. 2007; Lai et al. 2007]. This iteratively and adaptively samples and resamples a population of paths proportional to their contribution and guides future samples, typically by adapting the parameters of distributions or kernels used to generate samples. While this is related to our approach, it is not trivial to combine PMC with MCMC methods without biasing the result, and it is also not clear how this approach can be applied when computing high dimensional integrals.

One approach outside of the graphics literature which is closely related to our work is **Affine Invariant Sampling** (**AIS**) [Goodman and Weare 2010]. This work considered a pool or ensemble of walkers $\in \mathbb{R}^N$, and used the states of all other walkers to guide perturbations for each walker. The authors proposed three perturbations: stretch moves which shift a walker's position toward or away from a randomly sampled walker in the ensemble, a walk move which samples a subset of walkers and builds a Gaussian transition kernel, and a replacement move which aims to reconstruct the whole space and sample from the reconstruction. These techniques were shown to efficiently guide the sampling, especially in the case of complicated distributions. This was further extended by Foreman-Mackey et al. [2013] which proposed a parallel approach to using the ensemble of walkers. We also use a similar approach of partitioning the walkers into two pools, and using one pool to guide sampling in the complementary pool.

## 3 ENSEMBLE METROPOLIS LIGHT TRANSPORT

Our approach is based on an ensemble of paths which capture global information of the distribution of lighting in a scene to guide sampling for each path in the ensemble. We first define an ensemble of chains containing $O$ paths:

$$\mathbf{X} = \{\overline{x}^1, \overline{x}^2, .., \overline{x}^O\}. \tag{5}$$

This ensemble can be considered to be in $\mathcal{P}^O$. Similar to the argument in Goodman and Weare [2010], if we consider a product density using the ensemble $F(\mathbf{X}) = f(\overline{x}^1) f(\overline{x}^2)..f(\overline{x}^O)$, then any MCMC algorithm which preserves this density is valid. Such a strategy is to update each path in the ensemble conditioned on the other paths in the ensemble, i.e., following partial resampling [Liu 2008], if when updating path $\overline{x}^i$ the remaining paths in the ensemble $\{\overline{x}^1, .., \overline{x}^{i-1}, \overline{x}^{i+1}, .., \overline{x}^O\}$ remain fixed, then the update of the $i'$th path of the ensemble preserves the joint distribution $F$. This also allows the other paths in the ensemble to guide the sampling of each path of the ensemble.

This implies updating each path in series, as each update relies on fixing the states of all other paths in the ensemble. However, paths can be updated in parallel for all $\overline{x}^i \in \mathbf{X}$ by defining a complementary ensemble, $\mathbf{Y} = \{\overline{y}^1, \overline{y}^2, .., \overline{y}^O\}$, to guide sampling for each path in $\mathbf{X}$ [Foreman-Mackey et al. 2013]. Therefore, each path in $\mathbf{X}$ can be processed in parallel using $\mathbf{Y}$ as guidance for sampling, i.e., the transition kernel takes the form $T(\overline{x}^i \to \overline{x'}^i | \mathbf{Y})$. This transition kernel can be written as the product of multiple sampling events; in the case of light transport this corresponds to progressively sampling a subpath:

$$T(\overline{x}^i \to \overline{x'}^i | \mathbf{Y}) = \prod_{j=1}^{k} K(\overline{x}^i_j \to \overline{x'}^i_j | \mathbf{Y}), \tag{6}$$

where $K(\overline{x}^i_j \to \overline{x'}^i_j | \mathbf{Y})$ is a transition kernel for the $j'$th sampling event of $k$ events. Specifically, this is the transition kernel associated with perturbing the direction of a path vertex conditioned on the set of paths from the complementary ensemble. This transition kernel can be applied to one or more path vertices, producing a perturbation to a light path. The acceptance probability for updating each path in the ensemble is therefore computed as

$$a(\overline{x}^i \to \overline{x'}^i) = min\left(1, \frac{f^*(\overline{x'}^i) T(\overline{x'}^i \to \overline{x}^i | \mathbf{Y})}{f^*(\overline{x}^i) T(\overline{x}^i \to \overline{x'}^i | \mathbf{Y})}\right). \tag{7}$$

**ALGORITHM 1:** The EMLT algorithm. Two ensembles of paths **X** and **Y** are input, and during rendering, paths from the ensemble **X** are processed in parallel $it$ times, and the guided transitions kernels based on **Y** are used to propose new paths. After all paths in **X** are processed, **X** and **Y** are swapped.

```
   Input: X and Y
1  while rendering do
2      ParFor $\overline{x}^i \in \mathbf{X}$
3          for $it$ iterations do                                  ▷ See Section 3.4
4              $\overline{x'}^i \sim T(\overline{x}^i \to \overline{x'}^i | \mathbf{Y})$     ▷ See Section 3.3
5              $a \leftarrow a(\overline{x}^i \to \overline{x'}^i | \mathbf{Y})$               ▷ Equation (7)
6              Accumulate to Image
7              if $\xi < a$ then
8                  $\overline{x}^i \leftarrow \overline{x'}^i$
9              end
10         end
11     end
12     Swap X and Y
13 end
```

Once all the paths of the ensemble **X** have been updated, this is referred to as an iteration, the ensembles are swapped $\mathbf{X} \leftrightarrow \mathbf{Y}$ and paths of **Y** are updated based on using **X** as path guidance: $T(\overline{y}^i \to \overline{y'}^i | \mathbf{X})$. However, without loss of generality we refer to **Y** as the complementary ensemble in the remainder of the text. Algorithm 1 summarizes the EMLT algorithm. Firstly, the two ensembles **X** and **Y** are initialized, then during rendering each path is processed in parallel $it$ times (lines 2 and 3) using the proposed guided transition kernels (lines 4–9). When all paths in an ensemble are processed, then ensembles are swapped (line 12), and the process repeats.

The use of ensembles for guiding sampling of paths could be applied to either path space or PSS. One possibility is to apply the use of ensembles to PSS through a strategy which directly perturbs a point in PSS based on other points in the ensemble, similar to AIS. However, due to the difference in the number of random numbers required to sample paths, it is not clear how walkers of different dimensionalities could be used to create any of the transition kernels proposed by Goodman and Weare [2010]. Secondly, interpolating between points in high dimensions, which is the result of applying AIS to PSS, is unlikely to lead to usable paths, especially if there are small regions in PSS containing valid light transport paths. However, the alternative of applying this to path space is also not trivial as samples are no longer in $\mathbb{R}^N$, and the strategies outlined in Goodman and Weare [2010] are not immediately applicable. Our proposed transition kernels are designed to be suitable for path space, but are also constructed to inherit the advantages of using an ensemble to guide sampling.

This then allows scope for a wide range of new guided transition kernels which are conditioned on the complementary ensemble. While the entire complementary ensemble could be used to create transition kernels, this would be prohibitively expensive when the complementary ensemble is large. An alternative, and significantly faster, approach that we propose in this article is to use a carefully chosen subset of the paths in the complementary ensemble. These paths should be similar, both in interaction types and spatial proximity, such that they can still produce valid guided transition kernels. Section 3.1 describes how to efficiently find and weight the subset of paths from the complementary ensemble, then Section 3.2 describes how guided transition kernels can be constructed from this subset of paths. Finally, Section 3.3 describes how these guided transition kernels can be combined into path perturbations.

## 3.1 Complementary Ensemble

Before describing the transition kernels, we first describe two aspects of using the complementary ensemble for sampling. The first, explained in Section 3.1.1, is how to select paths from the complementary ensemble for sampling. This is important as many of the guided perturbations require paths to be sampled that maintain the same number of path vertices with the same interaction types as the original path.

The second aspect deals with the similarity of paths sampled from the complementary ensemble to the original path. This is required as although paths with the same length and interaction types may be sampled from the ensemble, paths which are similar to the original path are likely to lead to better proposal distributions than those further away. Effective use of this similarity between paths is what allows our approach to avoid a spatial cache. Section 3.1.2 describes an approach for measuring similarity between paths.

*3.1.1 Finding Paths.* As discussed previously, the proposed guided perturbations rely on a subset of $M$ paths from $\mathbf{Y}$: $\Upsilon = \{\overline{v}^1..\overline{v}^M\}$. These $M$ paths are located in $\mathbf{Y}$ based on similar properties to a base path $\overline{x}^i$, such as identical length or the same Heckbert notation interaction types [Heckbert 1990]. This is motivated by two observations: (i) perturbations guided by similar paths, rather than all paths, are likely to explore similar regions of path space leading to higher acceptance probabilities and (ii) perturbations are more likely to succeed since they rely on preserving interaction types.

Therefore, the set of paths in $\Upsilon$ is deterministically selected from the paths in $\mathbf{Y}$ with similar properties to $\overline{x}^i$. This is facilitated by a tree data structure over path lengths and interaction types which can be queried in $O(1)$ time to find a subset of $\mathbf{Y}$ which matches the desired properties. This is built at the start of the rendering process, or at the end of each iteration, and please see the supplementary material for more details about the construction and traversal of this data structure. If the selection of paths forming $\Upsilon$ was probabilistic and dependent on $\overline{x}^i$, then the probability of sampling the set $\Upsilon$ given $\overline{x}^i$ would have to be computed taking into account all paths with similar properties in $\mathbf{Y}$ which would be prohibitively slow. By performing a deterministic selection, in our case based on a counter which is stored with the ensemble and updated each step, this has the effect of having a minimal impact on performance with the additional benefit that the computation of the acceptance probabilities is significantly simplified as the probability of sampling $\Upsilon$ is not required.

*3.1.2 Measuring Similarity.* The set of paths returned from querying the ensemble, $\Upsilon$, may have similar properties to the current light path $\overline{x}^i$. However, while some of the vertices in the paths returned may have similar positions in world space to $\overline{x}^i$, others may not. When developing guided transition kernels, it is useful to have a measure of how similar light paths, or vertices within light paths, are to each other. For instance, some transition kernels can benefit from calculating weights for each vertex from $\Upsilon$ as this is likely to provide a good estimate of nearby lighting.

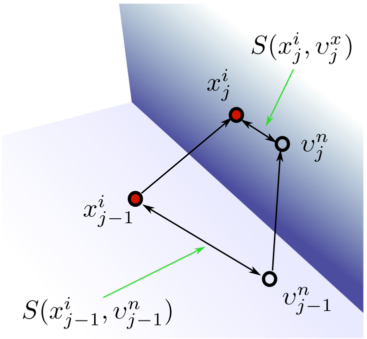


Fig. 2. Measuring similarity between a base path (red circles) and a path from $v^n \in \Upsilon$ (empty circles). This is a product of the similarity between pairs of path vertices with the same index $j$: $S(x_j^i, v_j^n)$.

The use of entire light paths in MCMC methods widens the range of methods to measure similarity. While Chaitanya et al. [2018] proposed an effective heuristic of total path length, i.e., the sum of distances between path vertices, we typically do not need to consider the whole path. We develop a heuristic based on the world space position of a set of vertices from $\Upsilon$, and vertices in the current light path $\overline{x}^i$. Other attributes, such as normals, albedo, or surface roughness could be considered, but we found that using the world space position was effective for computing similarity.

Specifically, given the $j$'th vertex from $\overline{x}^i$, $x_j^i$, and the previous vertex, $x_{j-1}^i$, the similarity value can be computed for all paths in $\Upsilon$ by computing the distance to $v_j^n$ and $v_{j-1}^n$, $n \in [1..M]$. We define the difference between the world position of two vertices as $d(x_j^i, v_j^n) = max(|x_j^i - v_j^n|^2, \epsilon)^{-1}$ where $\epsilon$ is a small positive constant (we use $\epsilon = 0.0001$). From this, we define a normalized similarity value as

$$S\left(x_j^i, v_j^n\right) = \frac{2}{1 + e^{-d\left(x_j^i, v_j^n\right)}} - 1. \tag{8}$$

This scaled sigmoid leads to a larger value when vertices are similar, and smaller the further apart they become. See Figure 2 for an illustration of similarity computation. The similarity of multiple vertices starting at the $j$'th position in the path to the $k$'th position can be computed as

$$S\left(x^i, v^n, j, k\right) = \prod_{l=j}^{k} S\left(x_l^i, v_l^n\right). \tag{9}$$

## 3.2 Guided Transition Kernels

We first describe guided transition kernels for a single vertex, and then describe how full perturbation strategies can be built from these individual strategies in the following section. All of these methods require information gathered from the set returned from querying the tree structure $\Upsilon$.

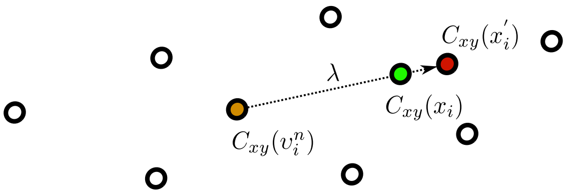


Fig. 3. Given some 2D domain, vertices in $\Upsilon$ can be projected onto that domain (empty circles). One of these is selected (the orange circle) and used by Linear Transition Kernels to form a ray from that point to a projection of a vertex from the current path (the green circle). A distance along this ray is sampled $\lambda$ which generates a new point in this domain (the red circle).

**Linear Transition Kernel.** The simplest form of guided transition kernels are the linear transition kernels, which is suitable for guided sampling on the lens. These operate in $\mathbb{R}^2$ in our implementation. Given coordinates $C_{xy}(x_i) \in \mathbb{R}^2$ of the current path vertex and the coordinates of a path vertex from $\Upsilon$: $C_{xy}(v_i^n) \in \mathbb{R}^2$, this generates a proposal along a ray in $\mathbb{R}^2$: $C_{xy}(x_i') = C_{xy}(v_i^n) + \lambda \cdot (C_{xy}(x_i) - C_{xy}(v_i^n))$. The distance along the ray, $\lambda$ is sampled from a distribution centered on $C_{xy}(x_i)$. This is illustrated in Figure 3.

Goodman and Weare [2010] proposed the stretch move which samples $\lambda$ from a distribution $\lambda \sim g(c) = \frac{1}{\sqrt{c}}$, where $c \in [\frac{1}{1+\alpha}, 1+\alpha]$, where $\alpha \in \mathbb{R}^+$ is a scaling term. This density is symmetric $g(c) = cg(\frac{1}{c})$ (see Goodman and Weare [2010]), and this leads to the ratio of $\frac{K(\overline{x'}^i_j \rightarrow \overline{x}^i_j|\mathrm{Y})}{K(\overline{x}^i_j \rightarrow \overline{x'}^i_j|\mathrm{Y})} = \lambda$, simplifying the acceptance probability. However, other 1D distributions can be sampled to generate $\lambda$. For example, a uniform $\lambda \sim [1-\beta, 1+\beta], \beta \in \mathbb{R}^+$, or truncated Gaussian can be used, and so long as these are symmetric they simplify in the computation of the acceptance probability.

**Linear Hemispherical Transition Kernels.** While linear transition kernels are defined in $\mathbb{R}^2$, many transition kernels are required to be defined over the (hemi)sphere $\mathbb{S}^2$. Therefore, we extend the linear transition kernels to the (hemi)sphere. This starts by sampling $\lambda$ from one of the linear distributions, then mapping this to a perturbation of the original direction on the sphere. Given two directions in the sphere $\omega_1$ and $\omega_2$, these directions may correspond to a direction on the original path, and the other on a path from $\Upsilon$, a new direction $\omega_{new}$ can be sampled along the great arc connecting these two directions: $\omega_{new} = \frac{\omega_1 \sin((\omega_1 \cdot \omega_2)\lambda)}{\sin((\omega_1 \cdot \omega_2))} + \frac{\omega_2 \sin((\omega_1 \cdot \omega_2)(1-\lambda))}{\sin((\omega_1 \cdot \omega_2))}$, i.e., a slerp between $\omega_1$ and $\omega_2$ with parameter $1-\lambda$ (see Figure 4). This leads to

$$\frac{K(\overline{x'}^i_j \rightarrow \overline{x}^i_j|\mathrm{Y})}{K(\overline{x}^i_j \rightarrow \overline{x'}^i_j|\mathrm{Y})} = \frac{\sin\cos^{-1}(\omega_1 \cdot \omega_2)}{\sin\cos^{-1}(\omega_{new} \cdot \omega_2)} = \frac{\sqrt{1-(\omega_1 \cdot \omega_2)^2}}{\sqrt{1-(\omega_{new} \cdot \omega_2)^2}}. \tag{10}$$

**Guided Anisotropic Transition Kernels.** The linear and hemispherical transition kernels rely on a single path from the complementary ensemble, and domains in $\mathbb{R}^2$ or $\mathbb{S}^2$. However, more information can be gained from utilizing all $M$ paths in $\Upsilon$. For example, a distribution in world space can be fit to the vertices at a certain point along the path, recentered at the current path vertex, and this distribution can be used for sampling. This allows lighting information from multiple paths to inform sampling of the current path, similar to Reibold et al. [2018].

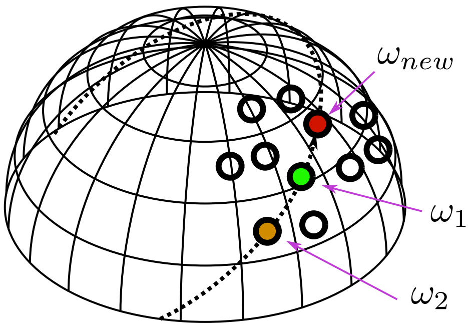


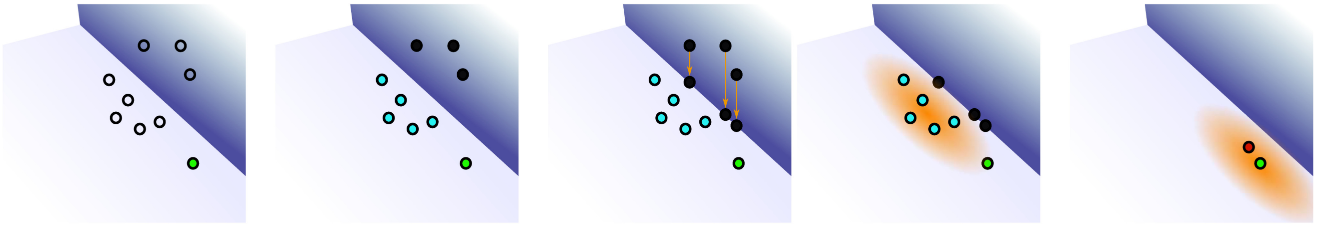
Fig. 4. The Linear Hemispherical Transition Kernel uses the outgoing direction of a path from $\Upsilon$ (the orange circle), and the current path (the green circle) to propose a new direction (the red circle) along the great arc denoted by the dashed line. Other directions from $\Upsilon$ which are not considered are shown as empty circles.

There are multiple methods to achieve this; we describe one such approach. We start with the $j$'th vertex in a path $x_j$ and another vertex in the scene $x'_{j-1}$, and then retrieve the set of path vertices from $\Upsilon$ which match the index: $v_j^n, v_{j-1}^n \in \Upsilon$. For each subpath, we assign a weight:

$$w(n) = \frac{S(x^i, v^n, j, j+1)}{\sum_{k=1}^{M} S(x_j, v^k, j, j+1)}. \tag{11}$$

Then each of these points is projected onto the plane defined by $x_j$ and the normal at $x_j$: $N(x_j)$. Next, an anisotropic Gaussian $\mathcal{N}(\mu, \Sigma; \Upsilon, x_j)$ is fitted to these points via weighted maximum likelihood estimation where the weight of each point is that assigned to each path: $\mu = \frac{1}{M}\sum_{n=1}^{M} w(n) v_j^n$ and $\Sigma = \frac{1}{M-1}\sum_{n=1}^{M} w(n)(v_j^n - \mu)^2$. This is recentered such that $\mu = x_j$, leading to $\mathcal{N}(x_j, \Sigma; \Upsilon, \overline{x}_j)$. This recentering is required such that the sampled point is close to the original, and to ensure that the evaluation of the reverse transition kernel returns a value similar to the proposed transition kernel in the computation of the acceptance probability. The steps of this algorithm are shown in Figure 5. Occasionally, all weights can be zero, or a degenerate covariance matrix can be computed. We detect these cases, and revert to a von Mises–Fisher distribution aligned in the direction $x'_{j-1} \rightarrow \overline{x}_j$ with a high concentration parameter for sampling. Another approach could be to convolve with an isotropic Gaussian similar to Li et al. [2015]; however, the value to use for the variance of the isotropic Gaussian is unclear in our case.

Once the anisotropic Gaussian is defined in world space, it is sampled producing a point $z'_j$. This point may not be aligned to the scene geometry, so a ray is traced from $x_{j-1}$ in the direction $x_{j-1} \rightarrow z'_j$, producing a new point on the scene manifold $x'_j$. The



(a) Initially the set of path vertices from $\Upsilon$ are located (empty circles) which may be spread around the scene. The resulting distribution needs to be centered on $x_j$, shown as the filled green circle

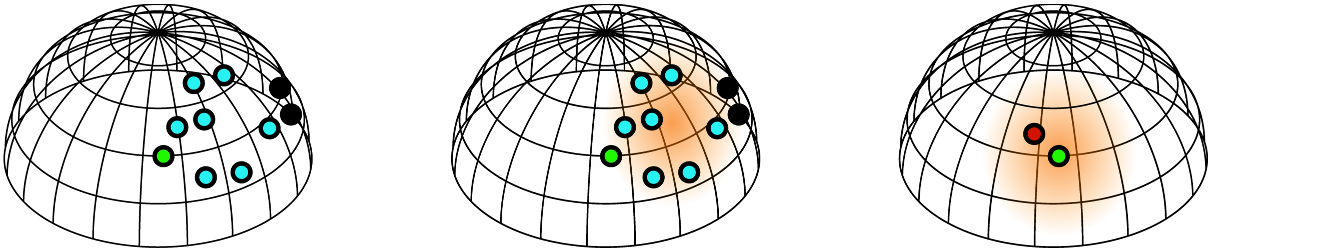
(b) The set of vertices from $\Upsilon$ are weighted by $S(x_j, v^k, j, j+1)$, blue filled circles indicate larger weights.

(c) All vertices are then are projected onto the plane defined by $x_j$ and the normal at $x_j$

(d) An anisotropic gaussian $\mathcal{N}(\mu, \Sigma; \Upsilon, x_j)$ is created on the plane from a weighted maximum likelihood estimation of the vertices from $\Upsilon$ projected onto the plane

(e) The anisotropic gaussian is then recentered at $x_j$ producing $\mathcal{N}(x_j, \Sigma; \Upsilon, \overline{x}_j)$. A nearby sample can be generated by sampling the gaussian (the red filled circle).

Fig. 5. The procedure to build and sample anisotropic Gaussian transition kernels.



(a) The nearby directions $\omega_n$ from $\Upsilon$ are computed and weighted (blue and black circles).

(b) A spherical distribution is fit to this set of directions (orange)

(c) This distribution is then recentered around the original direction (green circle), and a new direction is sampled (red circle).

Fig. 6. Generating and sampling using the guided directional transition kernels.

density w.r.t. area of computing this point is $K(\overline{x}^i_j \to \overline{x'}^i_j|\Upsilon) = \mathcal{N}(z'_j|x_j, \Sigma; \Upsilon, x_j)\frac{G(x'_j \leftrightarrow x_{j-1})}{G(z'_j \leftrightarrow x_{j-1})}$, where the final term stems from the ratio of geometry terms resulting from the Jacobian from sampling a point on the plane over which $\mathcal{N}(z'_j|x_j, \Sigma; \Upsilon, x_j)$ is defined, to the scene manifold.

To compute the acceptance probability, this process has to be computed in reverse; the distribution $\mathcal{N}(x'_j, \Sigma; \Upsilon, x'_j)$ is first computed, then the vertex $x_j$ is projected onto the plane defined by $x'_j$ and $N(\overline{x'}_j)$ leading to a point $z_j$. This leads to a resulting ratio:

$$\frac{K(\overline{x'}^i_j \to \overline{x}^i_j|\Upsilon)}{K(\overline{x}^i_j \to \overline{x'}^i_j|\Upsilon)} = \frac{\mathcal{N}(z_j|x'_j, \Sigma; \Upsilon, x'_j)G(z'_j \leftrightarrow x_{j-1})G(x_j \leftrightarrow x_{j-1})}{\mathcal{N}(z'_j|x_j, \Sigma; \Upsilon, x_j)G(z_j \leftrightarrow x_{j-1})G(x'_j \leftrightarrow x_{j-1})}. \tag{12}$$

A simpler version of this approach can be used on the image plane. In this case, an anisotropic Gaussian can be fit to the image plane coordinates of the paths in $\Upsilon$, each weighted by the similarity measure. Again, this can be centered at the image plane coordinates of the current path, and a new point on the image plane for the proposed path can be sampled from this distribution.

**Guided Directional Transition Kernels.** Guided anisotropic transition kernels form an anisotropic distribution in world space. However, sometimes it is useful to sample perturbations over solid angle. The linear hemispherical transition kernel performs this, but restricted along a great arc. Another approach is to fit a distribution on $\mathbb{S}^2$. Various approaches for this exist, for example tabulated, spherical Gaussian or a von Mises–Fisher distribution. Any distribution on the sphere whose parameters can be estimated from a set of directions can be used. Given a set of normalized directions from some base vertex $x_j$ to each member of $\Upsilon$, $\omega_n = x_j \to v^n_j$, and weights computed in the same manner as Equation (11), the parameters of a distribution can be estimated. Similar to the guided anisotropic perturbations, this distribution is recentered around the original direction from the vertex. This can then be sampled generating directions which are guided by nearby paths. Figure 6 shows this process.

### 3.3 Guided Perturbation Strategies

The previous section defined a range of guided transition kernels which are designed to update individual path vertices guided by global information from the ensemble. When perturbing a path, these guided transition kernels can be combined into a wide range of guided perturbation strategies designed to explore different lighting effects. Note that these can be combined with the original mutation and perturbation strategies; this simply adds to the strategies available. We always include the bidirectional mutation strategy from Veach and Guibas [1997] as this ensures ergodicity, thereby guaranteeing that the whole space will be explored.

The following lists the strategies we have implemented, but many more can be built using combinations of the kernels defined in Section 3.2.

**Linear Lens Perturbation.**

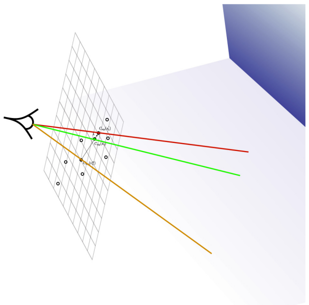


Fig. 7. Linear Lens Perturbation.

The Linear Lens Perturbation uses the linear transition kernel on the image plane, then similar to Veach and Guibas [1997] traces a subpath over any specular vertices, then connects to the original path. As $\Upsilon$ contains paths with similar interaction types and lengths to the current path, this strategy aims to explore the image plane around similar interaction types and also typically moves paths toward higher contribution regions for that path type. This strategy samples a path for the perturbation from $\Upsilon$, where the weight for the $k$'th path is given by $\frac{S(x^i,v^k,j,k)}{\sum_{p=1}^{N} S(x^i,v^p,j,k)}$, where $j$ is the index of a vertex on the camera, and $k$ is the index of the first non-specular vertex in the path. Figure 7 illustrates this strategy.

**Linear Caustic Perturbation.**

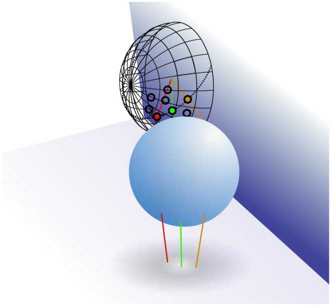


Fig. 8. Linear Caustic Perturbation.

Linear Caustic Perturbation uses the linear hemispherical transition kernel to perturb the sampled direction on the hemisphere to take into account nearby caustic paths. In this case, $\Upsilon$ will only contain caustic paths of the same number of path vertices, so they are likely to be exploring a similar region of the scene. This first finds the best path from $\Upsilon$ which closest matches the starting point and first specular vertex of the original caustic subpath, and sets the directions $\omega_1 = x^i_c \to x^i_{c-1}$, and $\omega_2 = x^i_c \to v^n_{c-1}$. This then perturbs the direction on the hemisphere, and traces the specular subpath to the first diffuse vertex, and connects to the camera. This is visualized in Figure 8.

**Linear Multi-Chain Perturbation.**

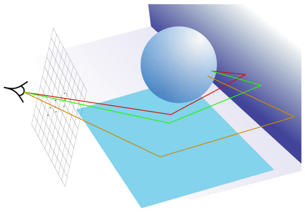


Fig. 9. Linear Multi-chain Perturbation.

Similar to the multi-chain strategy described in Veach and Guibas [1997], Linear Multi-chain Perturbation uses the linear transition kernel on the image plane in the same way as the Linear Lens Perturbation. This then traces a specular chain until a non-specular vertex is generated. A deterministic connection to the next specular subpath is then made and this process repeats until the path can be reconnected to the light subpath of the original path (see Figure 9).

**Anisotropic Path Perturbation.**

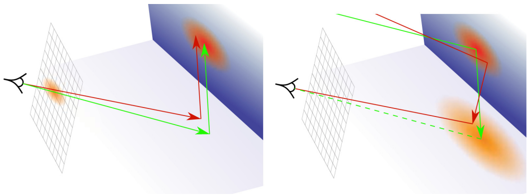


Fig. 10. The Anisotropic Path Perturbation uses the guided anisotropic perturbations for the first and second non-specular interactions from the camera. This can either start from the camera (left image), or toward the camera (right image).

This perturbation strategy comprises using the guided anisotropic transition kernels to perturb the current path. This can be applied to any number of path vertices, either from the light source or the eye. As the process of fitting an anisotropic Gaussian is relatively expensive, we restrict this perturbation to the first two vertices from the camera, and randomly select whether to sample from or toward the camera. If sampling from the camera is selected, an anisotropic Gaussian is created on the image plane and sampled, and for all other non-specular interactions the guided anisotropic transition kernels in world space are used, then reconnected to the original path. Likewise, if sampling toward the camera is selected, the guided anisotropic transition kernel is used to generate path vertices which are deterministically connected to the camera. This perturbation strategy helps to explore the local region around the path, and sampling more than one vertex from the camera helps to minimize the impact of the weak singularity in the geometry term near edges visible from the camera (see Figure 10). Another option is to sample with respect to solid angle similar to Otsu et al. [2018] or using the guided hemispherical perturbation which would cancel geometry terms in the calculation of the acceptance probability.

**Environment Perturbation.**

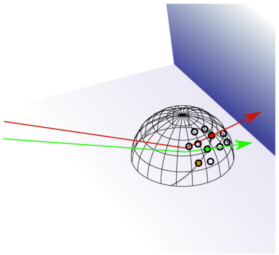


Fig. 11. Environment Perturbation.

This perturbation is designed to explore environment lighting based on nearby paths. This perturbation uses the linear hemispherical transition kernel or the guided directional transition kernel applied toward the environment map to perturb the direction to the environment map, assuming that the first path vertex from the light is non-specular. This is illustrated in Figure 11. Variants of this strategy can also be applied to other area light sources, or for light source selection.

## 3.4 Implementation Details

This method is initialized similar to the resampling approach described in MLT [Veach and Guibas 1997]. A large set of paths is computed with bidirectional path tracing, the contributions of these paths stored and used to scale the result, then these paths are resampled into two subsets: one to generate the ensemble $X$, and the other used to generate the complementary ensemble $Y$.

There is no limit on the size of $\mathbf{X}$ and $\mathbf{Y}$; however, for convenience we choose them to be the same size of $|\mathbf{X}| = |\mathbf{Y}| = 16,384$ (please see the supplementary material for further analysis). There is also much flexibility about when to update and swap ensembles (see Section 3 for details). Although updating the data structures for finding paths sampling is relatively inexpensive, this does come with some computational overhead of clearing out previous values and reinserting new values. Therefore, each path in the ensemble is perturbed or mutated $it$ times (see Algorithm 1) before the ensembles are swapped, which amortizes the overhead of updating the data structure compared to swapping after every mutation or perturbation. We choose $it$ to take a value of $\left\lfloor \frac{W \times H}{|\mathbf{X}|+|\mathbf{Y}|} \right\rfloor$, where $W$ and $H$ are the width and height of the image plane, respectively, which balances the computation cost of rebuilding the data structures and runtime performance.

There is significant freedom to choose the value of $\alpha$ used for the linear transition kernels described in Section 3.2. However, the method does not work well if this is set to a constant, as if the paths in $\Upsilon$ are clustered in a small region of space, e.g., in a small area on the image plane, then $\alpha$ should be large to facilitate exploration of a small space. Conversely, when paths are spread over a large area $\alpha$ should be small such that the proposed path is able to explore a similar region of the space. We solve this issue by adapting $\alpha$ based on $\Upsilon$. For a linear perturbation on the lens, we first compute a ratio of the bounding box of the image plane coordinates of each path in $\Upsilon$ to the image plane resolution: $b_{res}$. $\alpha$ is then computed by linearly interpolating between two bounds $\alpha_l$ and $\alpha_s$ based on a weight $w_\alpha = (1 + e^{-\frac{b_{res}-c}{\sigma^2}})^{-1}$, where $c \in [0..1]$. This uses a generalized sigmoid as a weighting function as it gives control over where and how fast the weights transition from 0 to 1. We use the parameters $\alpha_l = 0.5$, $\alpha_s = 0.05$, $c = 0.1$, and $\sigma^2 = 0.02$, although the algorithm is quite robust to these values. The supplementary material provides further details on the impact of $\alpha$.

For the linear lens, caustic, and multi-chain perturbations, if the number of paths in $\Upsilon$ is less than two, then the original perturbation strategies are used. This is to handle two situations: one is if no nearby paths are found, then the path can still be perturbed, and secondly, if only one path is found then there is too little information about nearby paths to create a useful sampling distribution. We also set the probabilities of sampling each proposed mutation type to be equal.

## 4 RESULTS

EMLT, MLT [Veach and Guibas 1997], and Geometry Aware MLT (GAMLT) [Otsu et al. 2018] were implemented into the same rendering framework for comparison. We implemented the Bidirectional, Lens, Caustic, and Multichain perturbations in MLT, and compare to GAMLT as it is the closest method to ours in terms of using adaptive sized perturbations in path space. We tested the methods in a variety of scenes, from those which exhibit challenging light transport where MCMC methods are expected to perform well, to simpler scenes which represent more common use cases for rendering. All results were computed on a laptop with an i7-8750H and 16GB RAM. Computation was spread over 12 threads using a thread pool to process paths in parallel and the ensemble was the same size per scene (see Section 3.4 for more information). We set a constant probability for the bidirectional mutation of $\frac{1}{3}$. All results were rendered at an average of 64 mutations per pixel to allow equal comparison between methods.

### 4.1 Indirect Lighting

Our method is primarily focused on efficiently computing global illumination. Therefore, we first investigate the performance of EMLT in scenes with indirect lighting only, as direct lighting can be efficiently handled by other techniques in these scenes when compared to MLT. We show results for six scenes which exhibit different types of lighting effects. The DOOR AJAR scene in Figure 1 is a challenging scenario where light propagates through the ajar door. Similarly, the BEDROOM scene in Figure 15 has thick, diffuse curtains with a light source on the other side leading to a very challenging lighting scenario. The CLASSROOM scene in Figure 15 is lit by an environment map with light entering through the windows. The KITCHEN scene in Figure 15 shows strong indirect lighting on the back wall and glossy reflections. The CORNELL BOX scene in Figure 16 is representative of many real-world scenes with simple lighting configurations. Finally, the STAIRCASE scene exhibits simple indirect lighting above the stairs, and more complicated indirect lighting under the stairs. Insets in the images show details, and the values printed on the top left of full resolution images correspond to **Mean Squared Error** (**MSE**) for the whole image.

In Figure 17, we show loglog convergence plots for MSE versus average mutations per pixel for the scenes used in this article to show how error decreases. This shows that there is an improvement in convergence using EMLT (blue line) compared to MLT (green line) and GAMLT (red line; see below for further discussion). This can be seen in the rendered images as a reduction in noise compared to MLT. Diffuse and low glossy surfaces, such as the walls in the DOOR AJAR, CLASSROOM, and CORNELL BOX scenes, or behind the cooker in the KITCHEN scene, exhibit significantly reduced variance with EMLT. This is due to the transition kernels adapting to both the illumination and scene as encoded in the paths in the ensemble. However, EMLT also captures higher frequency lighting effects, and can adapt to higher glossy materials, as can be seen in the BEDROOM scene above the curtains, the metal on the chairs in the CLASSROOM scene, and the strong indirect lighting on the back wall in the KITCHEN scene. EMLT leads to improvements in MSE for all tested scenes compared to MLT: 2.23× for DOOR AJAR, 1.38× for the BEDROOM scene, 2.11× for the CLASSROOM scene, 1.44× for KITCHEN, 1.71× for the STAIRCASE scene, and 2.93× for the CORNELL BOX scene.

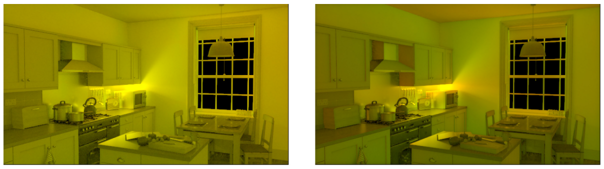


Fig. 12. Visualization of perturbations on the image plane for the KITCHEN scene. Green colors mean perturbations were predominantly vertical, red means predominantly horizontal, and yellow means perturbations were predominantly isotropic. The left image shows perturbations from MLT and the right shows our method.

Figure 12 visualizes the anisotropic nature of the perturbations proposed in this article for the KITCHEN scene; other scenes are shown in the supplementary material. The colors visualize the predominant direction of the perturbations on the image plane, green represents vertical, red shows horizontal, while yellow are isotropic. The perturbations for MLT are predominantly isotropic, as the lens and multi-chain perturbations use an isotropic transition kernel. In MLT, caustic perturbations are isotropic over the hemisphere, but do lead to an anisotropic distribution on the image plane. Our method in contrast adapts to both geometry and lighting information as can be seen in the right image in Figure 12. This shows that EMLT proposes perturbations which are predominantly horizontal in the strong horizontal indirect lighting on the rear wall, whereas on the rest of the wall the perturbations are predominantly vertical.

*4.1.1 Comparison to GAMLT.* Our implementation of GAMLT for indirect lighting used the geometry aware multi-chain perturbations with the same parameters as used for the results in Otsu et al. [2018], and we also extended this to use geometry aware lens and caustic perturbations following the same approach as described in Otsu et al. [2018]. We found this significantly improved the results in GAMLT and these extra perturbation strategies were implemented to facilitate a fair comparison between EMLT, MLT, and GAMLT. The convergence plots in Figure 17 show that EMLT outperforms GAMLT in several scenes, although both EMLT and GAMLT have similar variance reduction properties in the KITCHEN and CLASSROOM scenes. This is due to more small scale details in these scenes which GAMLT can adapt to, leading to approximately equal performance to EMLT. However, EMLT also adapts to these details, while also adapting to incident illumination. GAMLT is also significantly more computationally expensive than EMLT. In our implementation, we observed a 4 to 10 times increase in time to generate the same number of samples. Figure 18 shows results comparing EMLT to GAMLT for the STAIRCASE scene. Both EMLT and GAMLT are able to achieve variance reduction by considering scene geometry in the transition kernels, but EMLT is also able to better adapt to illumination, as can be seen in the inset images.

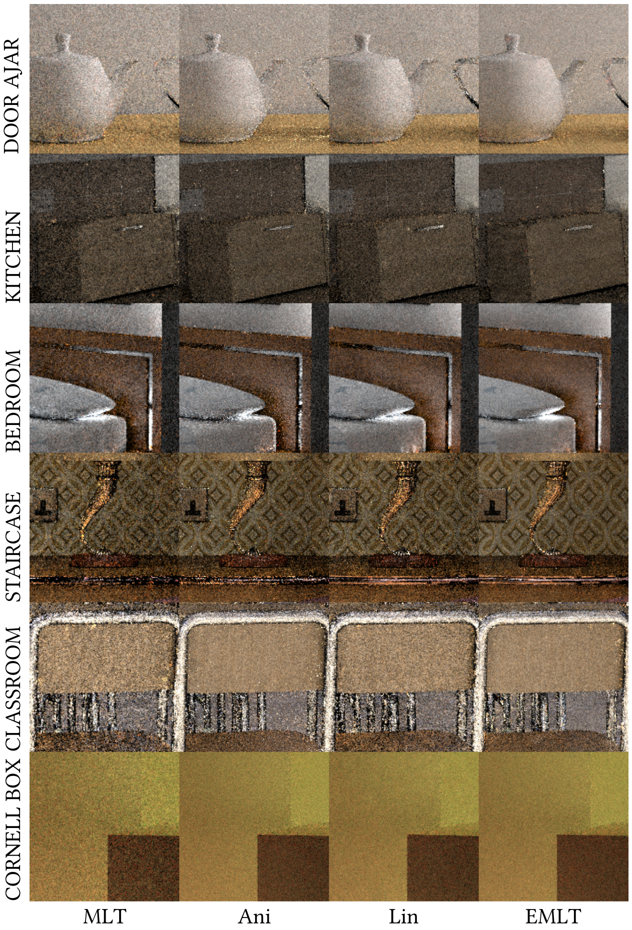


Fig. 13. Ablation study for the perturbation types presented in this article, where Ani refers to the Anisotropic Path Perturbation, Lin refers to the Linear Guided Perturbations, and EMLT refers to the combination. Combination refers to using all strategies presented in this article. The images show insets of the scenes used in this article, and MSE values are shown in Table 1.

Table 1. MSE Values for the Different Strategies Used in the Ablation Study (for the Scenes Used in this Article, see Figure 13 for Accompanying Images)

| Scene | MLT | Ani | Lin | EMLT |
|---|---|---|---|---|
| DOOR AJAR | 7.32e-3 | 4.14e-3 | 3.86e-3 | 3.28e-3 |
| KITCHEN | 2.19e-2 | 1.75e-2 | 1.56e-2 | 1.52e-2 |
| BEDROOM | 1.15e-4 | 9.28e-5 | 9.24e-5 | 8.34e-5 |
| STAIRCASE | 7.42e-5 | 5.07e-5 | 4.92e-5 | 4.34e-5 |
| CLASSROOM | 2.45e-2 | 1.26e-2 | 1.32e-2 | 1.16e-2 |
| CORNELL BOX | 9.62e-3 | 4.72e-3 | 4.34e-3 | 3.28e-3 |

## 4.2 Ablation Study

We performed an ablation study to assess the impact of the proposed strategies. We rendered the scenes using either the Anisotropic Path Perturbation, the Linear Guided Perturbations, or the combination of these two strategies. Figure 13 shows zoomed in regions of the scenes used in this article highlighting the differences between strategies visually, and Table 1 provides MSE values for these strategies across all scenes. This shows that the Anisotropic Path Perturbation is effective at reducing noise in regions with low frequency variations in lighting and is responsible for an average of 63% improvement over MLT in the scenes

Table 2. MSE Values for Direct Lighting

| Scene | MIS | MLT | EMLT |
|---|---|---|---|
| BREAKFAST ROOM | 0.1082 | 0.0213 | 0.0177 |
| SPONZA | 0.0193 | 0.0038 | 0.0031 |

in this article. However, this exhibits noise when sampling higher frequency lighting as the subset of paths contained in $\Upsilon$ are less likely to be able to capture this type of lighting effect. Conversely, the Linear Lens, Caustic, and Multichain Perturbations are efficient at capturing these higher frequency effects and are responsible for an average of 68% improvement over MLT, but exhibit more noise in more uniform regions of the scene. The combination of these strategies is able to reduce noise in both low and high frequency variations in lighting.

## 4.3 Direct Lighting

To more clearly demonstrate the performance of the environment perturbation, we apply our method to direct lighting. This uses a combination of the Guided Lens Perturbation to perturb path position on the image plane, and the Environment Perturbation to guide sampling on the environment map. We compare with a traditional approach of BRDF and environment sampling combined with MIS, MLT applied to direct lighting, and EMLT. Please note, this is not meant to compete with specialized direct lighting approaches, but to illustrate the proposed perturbation strategies. In Figure 19, we show results for the BREAKFAST ROOM and SPONZA scene, only showing direct illumination from the environment map. Table 2 shows MSE for the scenes and sampling techniques. This shows that our perturbations outperform MLT, and significantly improve on using MIS for direct lighting.

## 4.4 Performance

Our method has some computational overhead compared to MLT. On average, we observed an 18% overhead with our method due to (a) rebuilding the pools of paths (see Section 3.1.1), although this is amortized by infrequently updating the pools, and (b) slightly more complicated procedures for sampling perturbations. The majority of the overhead in our implementation comes from computing the similarity measure (Section 3.1.2). The impact of this overhead is illustrated in Figure 14, where we show the results for equal time versus equal quality for the scenes where our method performs best (DOOR AJAR) and worst (KITCHEN). The images show the same insets corresponding to Figures 1 and 15, respectively. These results show that the overhead of our method is outweighed by the variance reduction of guided transition kernels in EMLT.

EMLT also proposes perturbations which are more likely to be accepted than with MLT. For the scenes in this article, we observed a 14% increase in the acceptance probability averaged over all the scenes.

# 5 DISCUSSION AND FUTURE WORK

In this section, we briefly discuss our method, and propose directions to extend this work.

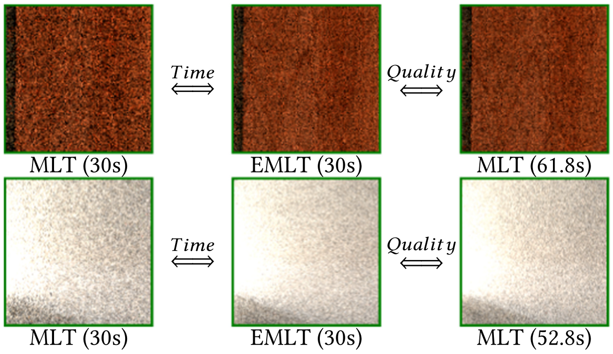


Fig. 14. Equal time versus equal quality for the DOOR AJAR and KITCHEN scenes. The images correspond to the same insets as the main results. The left column shows MLT rendered for 30 s, the middle is EMLT rendered for 30 s, and the right images show MLT rendered to the same MSE for the whole image.

**Derivative-Based Approaches.** Our method guides sampling based on creating distributions from an ensemble of paths which capture lighting information in the region near the current path. The size of this region depends on the number of paths selected from the pool ($\Upsilon$), and the size of the pool. Approaches such as Li et al. [2015] and Luan et al. [2020] use gradient information associated with the path to create proposal distributions which allow perturbations to be proportional to the local gradient. Our work is complementary to these approaches as we target perturbations guided over a wider region, and as such can take into account larger scale geometric and lighting details, whereas these approaches allow for more optimal local perturbations but do not consider lighting from nearby paths, or geometrical detail. Combining both approaches would be an interesting avenue for future work.

**Combination with Primary Sample Space.** Our approach works in world space as discussed in Section 3. However, using the approaches which fuse world space Metropolis Light Transport and PSSMLT [Bitterli et al. 2018; Otsu et al. 2017; Pantaleoni 2017] would allow our approach to be combined with PSS approaches. Another approach would be to adapt our method to work in PSS; however, this is not trivial as discussed in Section 3.

**Combination with Other Mutation Strategies.** Perturbation strategies such as Manifold Perturbations [Jakob and Marschner 2012], Multiple Try Metropolis [Nimier-David et al. 2019; Segovia et al. 2007], selectively choosing paths to perturb [Bitterli and Jarosz 2019], and Delayed Rejection MLT [Rioux-Lavoie et al. 2020] could all be combined with our approach into a larger set of possible strategies. Manifold perturbations are especially effective at locally perturbing specular paths, thus complementary to our approach which perturbs paths in a wider region. Delayed Rejection MLT would help balance between when to use the different strategies, leading to a more efficient method.

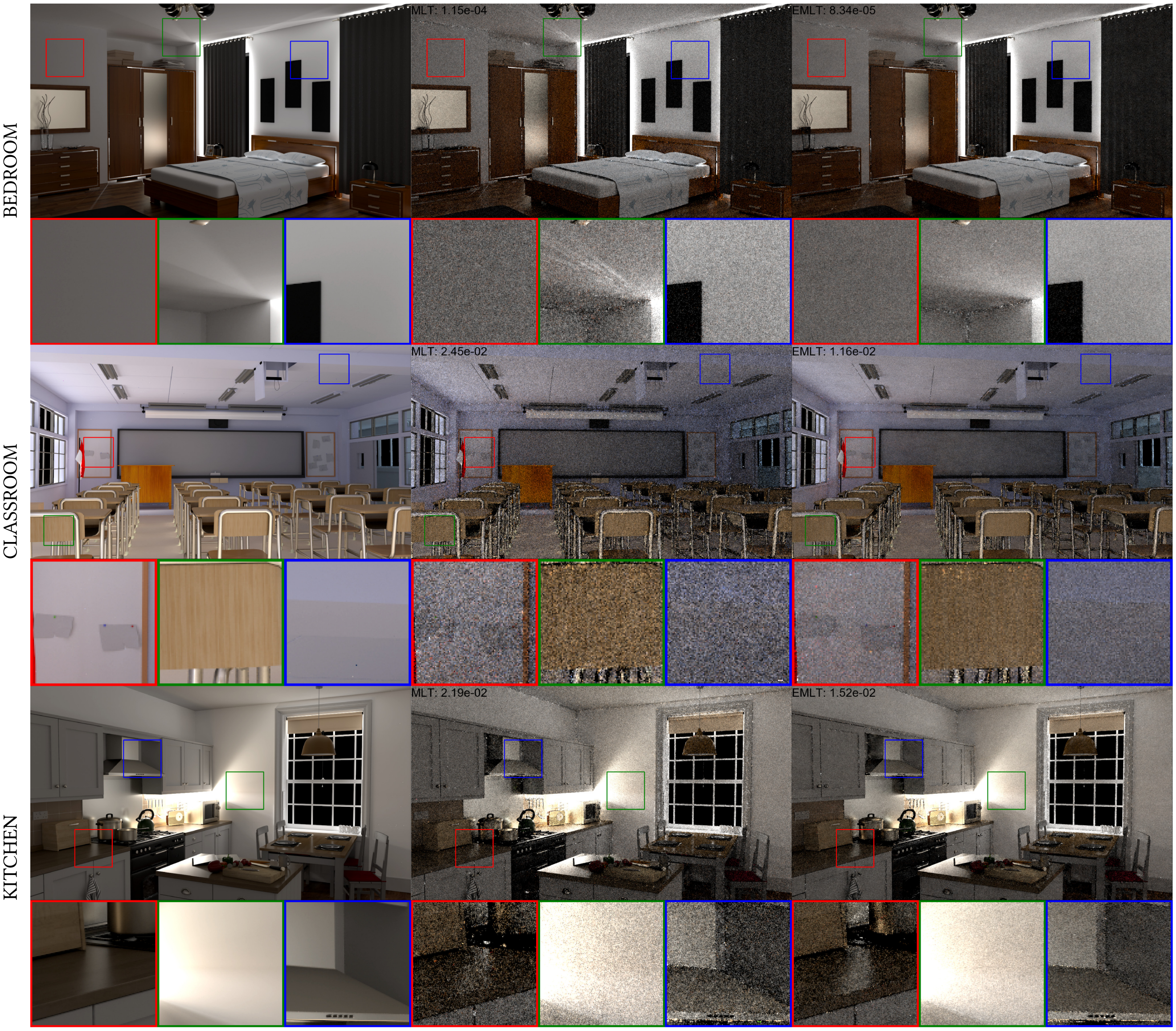


Fig. 15. Results for the BEDROOM, KITCHEN, and CLASSROOM scenes. The left images are the reference, middle is MLT, and the right images are EMLT. Insets highlight reduced variance with our method, for example showing where the sampling has adapted to scene geometry or illumination.

**Parameters.** Our method requires several parameters such as pool size, update frequency, the number of paths in $\Upsilon$, and the parameters used to compute $\alpha$. We discuss these parameters in Section 3.4; however, we do not claim these parameters are optimal. Theoretically finding optimal values of these parameters would be useful as it would further increase the efficiency of our method. One possibility is to use the scene acceleration structure to estimate maximum values for the parameters used in the same manner as the approach taken by Otsu et al. [2018].

**Limitations.** While our method achieves variance reduction for scenes with both complicated and simple lighting, there are some situations or sets of parameters where our method is outperformed by MLT. An example of this is when the size of the ensemble becomes very small. In this case, there is not enough information in the complementary ensemble to guide sampling, and our method falls back to MLT, albeit with the computational overhead of maintaining a pool. We found this was not a problem using the range of parameters outlined in Section 3.4, but there may be scenes which require the ensemble to represent more paths.

Finally, the approach of deterministically selecting a subset of paths from an ensemble of paths could be used for path guiding in non-MCMC methods such as path tracing. This would have the advantage of no longer requiring an additional spatial data structure as is needed by the approaches in Section 2.2, and would likely require different probability distributions than those used

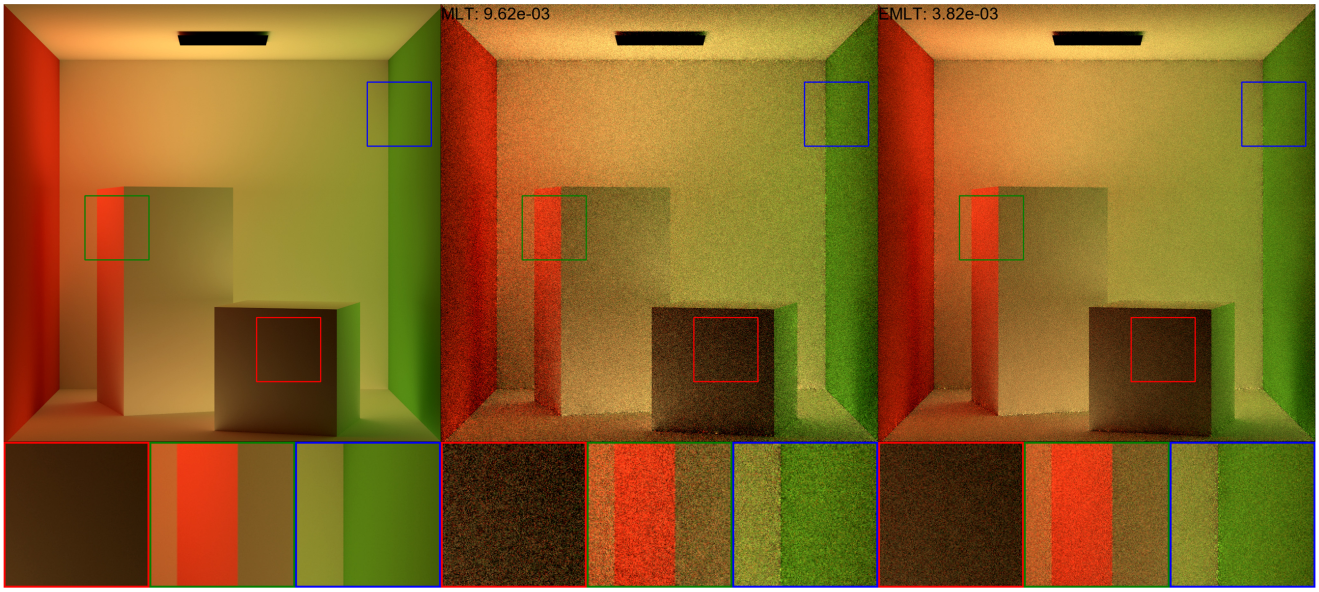


Fig. 16. Results for the CORNELL BOX scene. This illustrates that EMLT provides an advantage over MLT in simple scenes.

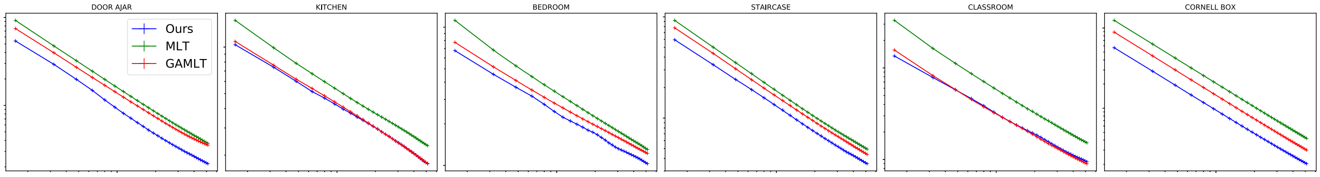


Fig. 17. Convergence graphs showing MSE versus average mutations per pixel on a logarithmic scale for the scenes used in this article. The blue line is EMLT, green is MLT, and red is GAMLT.

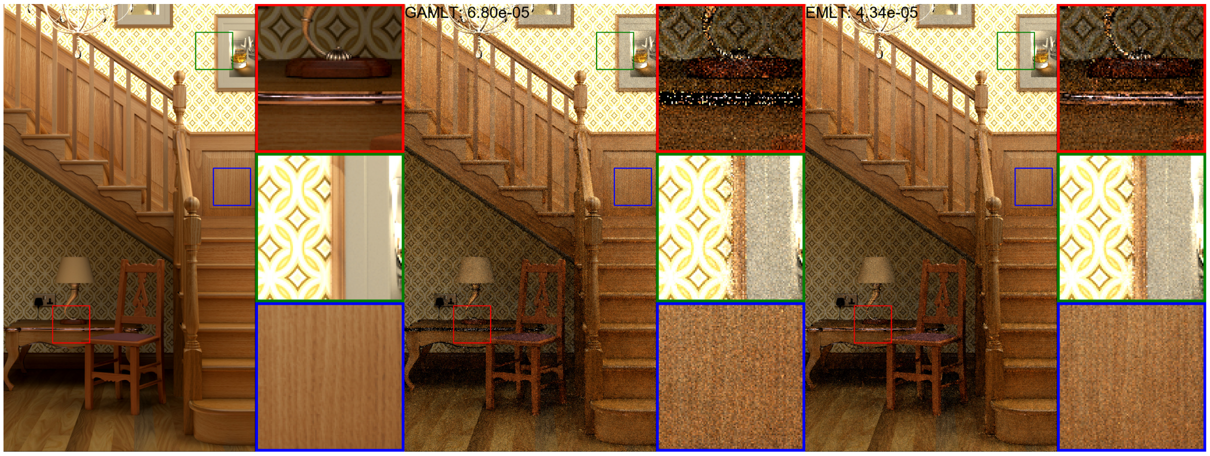


Fig. 18. Results for the STAIRCASE scene showing the reference on the left, GAMLT in the middle, and EMLT on the right. This shows that both EMLT and GAMLT can adapt transition kernels to the scene geometry, but EMLT can also sample illumination, leading to variance reduction.

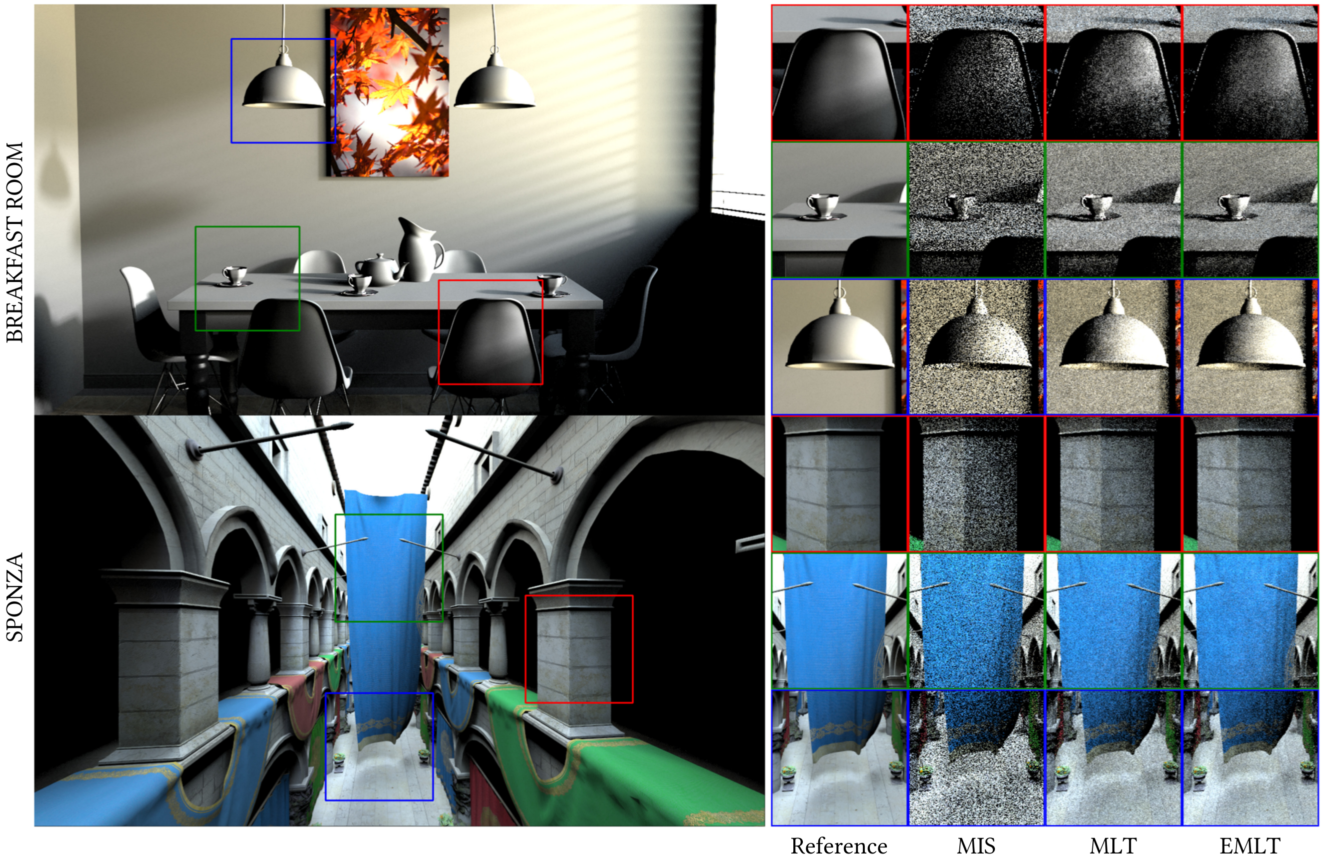


Fig. 19. Environment Lighting in the BREAKFAST ROOM scene (top row) and SPONZA (bottom row). The inset images show (left to right): Reference, MIS, MLT, and EMLT. Our guided perturbations reduce variance compared to MLT, and significantly compared to MIS.

in this article. However, we believe this could lead to conceptually simpler and easier to implement path guiding with lower memory overheads.

## 6 CONCLUSION

This article has presented a new family of transition kernels for MCMC rendering algorithms. These are based on efficiently sampling ensembles of transport paths, and utilizing these ensembles to guide path mutations. This approach does not require spatial caching of radiance or importance distributions, nor the associated spatial data structures, yet is efficient and reduces variance in scenes of different complexity and light transport effects. We believe that many more transport kernels of the type presented in this article are possible, and we hope this work opens up new possibilities for further variance reduction strategies for MCMC methods in the future.

## ACKNOWLEDGMENTS

We would like to thank the reviewers for their helpful comments and feedback. We also thank *Benedikt Bitterli* for the DOOR AJAR and CORNELL BOX scenes, *SlykDrako* for the BEDROOM, *Jay-Artist* for KITCHEN, *NovaZeeke* for CLASSROOM, *Wig42* for STAIRCASE and BREAKFAST ROOM, *Frank Meinl* for the SPONZA scene, and *Benedikt Bitterli* for making these scenes available [Bitterli 2016].

## REFERENCES

Steve Bako, Mark Meyer, Tony DeRose, and Pradeep Sen. 2019. Offline deep importance sampling for Monte Carlo path tracing. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 527–542.

Thomas Bashford-Rogers, Kurt Debattista, and Alan Chalmers. 2012. A significance cache for accelerating global illumination. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 1837–1851.

Benedikt Bitterli. 2016. Rendering resources. Retrieved on 14 March, 2020 from https://benedikt-bitterli.me/resources/.

Benedikt Bitterli, Wenzel Jakob, Jan Novák, and Wojciech Jarosz. 2018. Reversible jump metropolis light transport using inverse mappings. *ACM Transactions on Graphics (TOG)* 37, 1 (2018), 1.

Benedikt Bitterli and Wojciech Jarosz. 2019. Selectively metropolised Monte Carlo light transport simulation. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–10.

Olivier Cappé, Arnaud Guillin, Jean-Michel Marin, and Christian P. Robert. 2004. Population Monte Carlo. *Journal of Computational and Graphical Statistics* 13, 4 (2004), 907–929.

Chakravarty R. Alla Chaitanya, Laurent Belcour, Toshiya Hachisuka, Simon Premoze, Jacopo Pantaleoni, and Derek Nowrouzezahrai. 2018. Matrix bidirectional path tracing. In *Proceedings of the Eurographics Symposium on Rendering: Experimental Ideas & Implementations*. Eurographics Association, 23–32.

Per H. Christensen and Wojciech Jarosz. 2016. The path to path-traced movies. *Foundations and Trends® in Computer Graphics and Vision* 10, 2 (2016), 103–175.

David Cline, Justin Talbot, and Parris Egbert. 2005. Energy redistribution path tracing. In *ACM Transactions on Graphics (TOG)*, Vol. 24. ACM, 1186–1195.

Ken Dahm and Alexander Keller. 2017. Learning light transport the reinforced way. arXiv:1701.07403. https://arxiv.org/abs/1701.07403.

Stavros Diolatzis, Adrien Gruson, Wenzel Jakob, Derek Nowrouzezahrai, and George Drettakis. 2020. Practical product path guiding using linearly transformed cosines. In *Computer Graphics Forum*, Vol. 39.

ShaoHua Fan, Yu-Chi Lai, Stephen Chenney, and Charles Dyer. 2007. *Population Monte Carlo Samplers for Rendering*. Technical Report. University of Wisconsin-Madison, Department of Computer Sciences.

Daniel Foreman-Mackey, David W. Hogg, Dustin Lang, and Jonathan Goodman. 2013. emcee: The MCMC hammer. *Publications of the Astronomical Society of the Pacific* 125, 925 (2013), 306.

Jonathan Goodman and Jonathan Weare. 2010. Ensemble samplers with affine invariance. *Communications in Applied Mathematics and Computational Science* 5, 1 (2010), 65–80.

Adrien Gruson, Mickaël Ribardière, Martin Šik, Jiří Vorba, Rémi Cozot, Kadi Bouatouch, and Jaroslav Křivánek. 2016. A spatial target function for metropolis photon tracing. *ACM Transactions on Graphics (TOG)* 36, 4 (2016), 1.

Adrian Gruson, Rex West, and Toshiya Hachisuka. 2020. Stratified Markov chain Monte Carlo light transport. In *Eurographics*. The Eurographics Association.

Jerry Guo, Pablo Bauszat, Jacco Bikker, and Elmar Eisemann. 2018. Primary sample space path guiding. In *Eurographics Symposium on Rendering*, Vol. 2018. The Eurographics Association, 73–82.

Toshiya Hachisuka and Henrik Wann Jensen. 2011. Robust adaptive photon tracing using photon path visibility. *ACM Transactions on Graphics (TOG)* 30, 5 (2011), 114.

Toshiya Hachisuka, Anton S. Kaplanyan, and Carsten Dachsbacher. 2014. Multiplexed metropolis light transport. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 100.

W. Keith Hastings. 1970. Monte Carlo sampling methods using Markov chains and their applications.

Paul S. Heckbert. 1990. Adaptive radiosity textures for bidirectional ray tracing. *ACM SIGGRAPH Computer Graphics* 24, 4 (1990), 145–154.

Sebastian Herholz, Oskar Elek, Jiří Vorba, Hendrik Lensch, and Jaroslav Křivánek. 2016. Product importance sampling for light transport path guiding. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 67–77.

Heinrich Hey and Werner Purgathofer. 2002. Importance sampling with hemispherical particle footprints. In *Proceedings of the 18th Spring Conference on Computer Graphics*. ACM, 107–114.

Jared Hoberock and John C. Hart. 2010. Arbitrary importance functions for metropolis light transport. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 1993–2003.

Wenzel Jakob and Steve Marschner. 2012. Manifold exploration: A Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 58.

Henrik Wann Jensen. 1995. Importance driven path tracing using the photon map. In *Rendering Techniques '95*. Springer, 326–335.

James T. Kajiya. 1986. The rendering equation. In *ACM SIGGRAPH Computer Graphics*, Vol. 20. ACM, 143–150.

Anton S. Kaplanyan, Johannes Hanika, and Carsten Dachsbacher. 2014. The natural-constraint representation of the path space for efficient light transport simulation. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 102.

Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. 2002. A simple and robust mutation strategy for the metropolis light transport algorithm. In *Computer Graphics Forum*, Vol. 21. Wiley Online Library, 531–540.

Alexander Keller. 1997. Instant radiosity. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'97)*. ACM Press/Addison-Wesley Publishing Co., 49–56. https://doi.org/10.1145/258734.258769

Shinya Kitaoka, Yoshifumi Kitamura, and Fumio Kishino. 2009. Replica exchange light transport. In *Computer Graphics Forum*, Vol. 28. Wiley Online Library, 2330–2342.

Eric P. Lafortune and Yves D. Willems. 1995. A 5D tree to reduce the variance of Monte Carlo ray tracing. In *Rendering Techniques '95*. Springer, 11–20.

Yu-Chi Lai, Shao Hua Fan, Stephen Chenney, and Charcle Dyer. 2007. Photorealistic image rendering with population monte carlo energy redistribution. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*. Eurographics Association, 287–295.

Tzu-Mao Li, Jaakko Lehtinen, Ravi Ramamoorthi, Wenzel Jakob, and Frédo Durand. 2015. Anisotropic Gaussian mutations for metropolis light transport through Hessian-Hamiltonian dynamics. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 209.

Jun S. Liu. 2008. *Monte Carlo Strategies in Scientific Computing*. Springer Science & Business Media.

Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. 2020. Langevin Monte Carlo rendering with gradient-based adaptation. *ACM Transactions on Graphics* 39, 4 (2020).

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* 21, 6 (1953), 1087–1092.

Thomas Müller, Markus Gross, and Jan Novák. 2017. Practical path guiding for efficient light-transport simulation. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 91–100.

Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. 2018. Neural importance sampling. arXiv:1808.03856. https://arxiv.org/abs/1808.03856.

Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. 2019. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 203.

Hisanari Otsu, Johannes Hanika, Toshiya Hachisuka, and Carsten Dachsbacher. 2018. Geometry-aware metropolis light transport. In *SIGGRAPH Asia 2018 Technical Papers*. ACM, 278.

Hisanari Otsu, Anton S. Kaplanyan, Johannes Hanika, Carsten Dachsbacher, and Toshiya Hachisuka. 2017. Fusing state spaces for Markov chain Monte Carlo rendering. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 74.

Hisanari Otsu, Yonghao Yue, Qiming Hou, Kei Iwasaki, Yoshinori Dobashi, and Tomoyuki Nishita. 2013. Replica exchange light transport on relaxed distributions. In *ACM SIGGRAPH 2013 Posters*. ACM, 106.

Jacopo Pantaleoni. 2017. Charted Metropolis light transport. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 75.

Jacopo Pantaleoni. 2020. Online path sampling control with progressive spatio-temporal filtering. *SN Computer Science* 1, 5 (Aug. 2020), 279. https://doi.org/10.1007/s42979-020-00291-z

Mark Pauly, Thomas Kollig, and Alexander Keller. 2000. Metropolis light transport for participating media. In *Rendering Techniques 2000*. Springer, 11–22.

Florian Reibold, Johannes Hanika, Alisa Jung, and Carsten Dachsbacher. 2018. Selective guided sampling with complete light transport paths. In *SIGGRAPH Asia 2018 Technical Papers*. ACM, 223.

Damien Rioux-Lavoie, Joey Litalien, Adrien Gruson, Toshiya Hachisuka, and Derek Nowrouzezahrai. 2020. Delayed rejection metropolis light transport. *ACM Transactions on Graphics* 39, 3, Article 26 (April 2020). https://doi.org/10.1145/3388538

Lukas Ruppert, Sebastian Herholz, and Hendrik P. A. Lensch. 2020. Robust fitting of parallax-aware mixtures for path guiding. *ACM Transactions on Graphics* 39, 4 (July 2020), Article 147, 15 pages. https://doi.org/10.1145/3386569.3392421

Benjamin Segovia, Jean Claude Iehl, and Bernard Péroche. 2007. Metropolis instant radiosity. In *Computer Graphics Forum*, Vol. 26. Wiley Online Library, 425–434.

Martin Šik and Jaroslav Křivánek. 2016. Improving global exploration of MCMC light transport simulation. In *ACM SIGGRAPH 2016 Posters*. ACM, 50.

Martin Šik and Jaroslav Křivánek. 2018. Survey of Markov chain Monte Carlo methods in light transport simulation. *IEEE Transactions on Visualization and Computer Graphics* 26, 4 (2018), 1821–1840.

Martin Šik, Hisanari Otsu, Toshiya Hachisuka, and Jaroslav Křivánek. 2016. Robust light transport simulation via metropolised bidirectional estimators. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 245.

Robert H. Swendsen and Jian-Sheng Wang. 1986. Replica Monte Carlo simulation of spin-glasses. *Physical Review Letters* 57, 21 (1986), 2607.

Eric Veach and Leonidas J. Guibas. 1995. Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 419–428.

Eric Veach and Leonidas J. Guibas. 1997. Metropolis light transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press/Addison-Wesley Publishing Co., 65–76.

Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. 2014. Online learning of parametric mixture models for light transport simulation. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 101.

Quan Zheng and Matthias Zwicker. 2019. Learning to importance sample in primary sample space. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 169–179.

Received September 2020; revised April 2021; accepted June 2021