

Technical Report: Path Space Partitioning and Guided Image Sampling for MCMC

Thomas Bashford-Rogers and Luis Paulo Santos

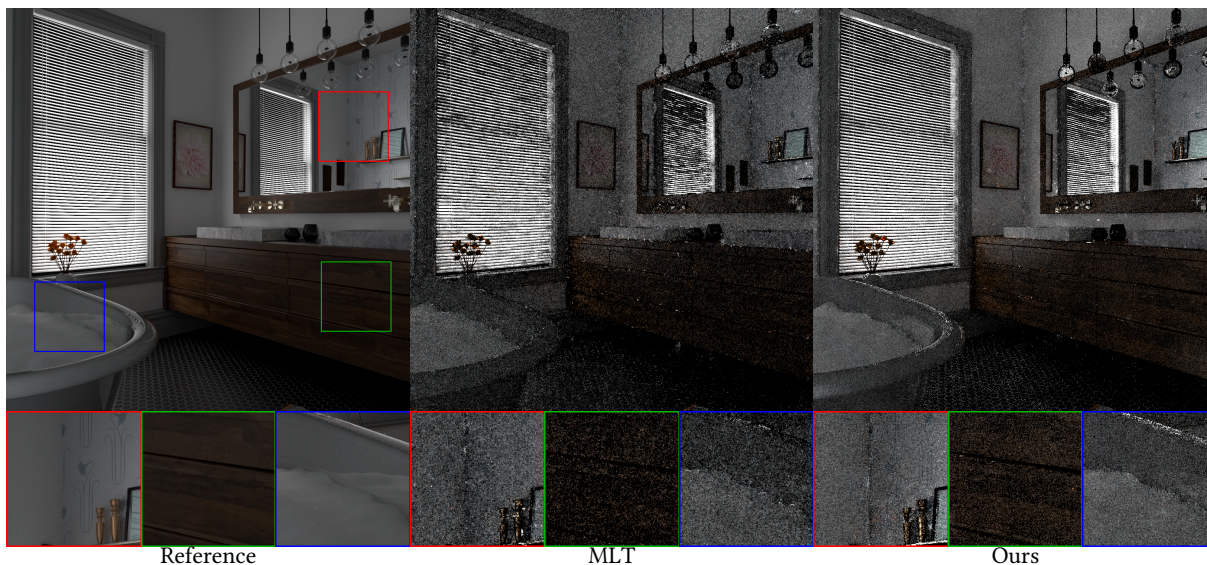


Figure 1: Our approach splits path space into a discrete set of partitions, each of which can be integrated by a separate estimator. As these are now integrating over sparser spaces, we propose a guided image plane sampling approach based on an analysis of the acceptance probability for image plane perturbations and accelerated by denoising the information used to create the partitions. This image shows the *bathroom* scene showing variance is reduced using our approach (on the right) and Metropolis Light Transport (in the middle) computed at the same number of samples.

ABSTRACT

Rendering algorithms typically integrate light paths over path space. However, integrating over this one unified space is not necessarily the most efficient approach, and we show that partitioning path space and integrating each of these partitioned spaces with a separate estimator can have advantages. We propose an approach for partitioning path space based on analyzing paths from a standard Monte Carlo estimator and integrating these partitioned path spaces using a Markov Chain Monte Carlo (MCMC) estimator. This also means that integration happens within a sparser subset of path space, so we propose the use of guided proposal distributions in image space to improve efficiency. We show that our method improves image quality over other MCMC integration approaches at the same number of samples.

CCS CONCEPTS

• Computing methodologies → Ray tracing.

KEYWORDS

MCMC, Path Space, Denoising

1 INTRODUCTION

Modern rendering algorithms rely on well established approaches for integrating over light paths. These typically find efficient ways of generating paths which connect light sources to the camera via a series of interactions with a scene. To render scenes containing complicated light transport efficiently, various methods have been proposed, such as Bidirectional Path Tracing [Lafortune and Willems 1993; Veach and Guibas 1995], ReSTIR approaches [Kettunen et al. 2023; Lin et al. 2022] and Markov Chain Monte Carlo [Hachisuka et al. 2014; Kelemen et al. 2002; Veach and Guibas 1997]. Many of the concepts in these works overlap, for instance exploiting the correlation between nearby paths in path or image space.

This work develops an approach for partitioning path space to further exploit these correlations, where each of these partitions can be integrated by a separate estimator. This can be more efficient than purely integrating over path space as a whole, and in this work we focus on integrating using MCMC methods as these form a useful starting point for developing traditional Monte Carlo estimators.

However, these partitions of path space are sparser than path space, and this poses a challenge to generate valid paths within each subspace. This means effective perturbation strategies need to be aware of the partitioned space, and constrain proposal distributions to this space. Motivated by the success of other image space

and low dimensional integration approaches, we propose utilizing information in image space to guide perturbations.

We achieve the partitioning by using a Monte Carlo path tracing pre-pass to estimate partitions of path space, then also denoising the contributions of the paths within each partition to build the image plane guidance distribution. To summarise, our contributions are as follows:

- A principled approach to partition path space into sub-spaces, each of which can be integrated by a separate estimator
- An image plane path guidance distribution to generate proposals which explore this partitioned path space, and propose a method to use sparse image plane information to build this proposal distribution.
- Results for our approach applied to path space MLT algorithms showing improvements in image quality.

2 RELATED WORK

Our work focuses on partitioning path space and guiding perturbations so we briefly review related work on MCMC algorithms and path guiding.

MCMC was first applied to rendering in Metropolis Light Transport by [Veach and Guibas 1997] who formulated the now ubiquitously used path space formalism of light transport. This applied Metropolis sampling [Hastings 1970; Metropolis et al. 1953] over a space of all possible paths, and achieved a substantial improvement in scenes containing hard to sample paths over conventional techniques such as path tracing [Kajiya 1986] or bidirectional path tracing [Lafortune and Willems 1993; Veach and Guibas 1995]. The main advantage of this approach was the local exploration of space around an existing path, allowing for substantially more non-zero contribution paths to be created at the same computational cost.

Path space MCMC has been further developed in several directions, for example [Pauly et al. 2000] extended MLT to participating media [Pauly et al. 2000]. [Jakob and Marschner 2012] proposed a perturbation based on manifold walks which efficiently connects path vertices through specular or near-specular interactions. [Otsu et al. 2018] proposed a perturbation strategy which adapted the perturbation size based on the local geometry, meaning that perturbations adapted well to high frequency geometry. Half vector space was used in [Kaplanyan et al. 2014] to constrain perturbations to a path and had the elegant property of canceling out most geometry terms in the acceptance probability. [Bashford-Rogers et al. 2021] proposed several perturbations based on using an ensemble of paths to guide sampling of an individual path. [Manzi et al. 2014] proposed a series of improvements which led to a decrease in variance in Metropolis sampling for gradient domain rendering, and used image space exploration to find proposed paths, although in the context of finding pixel shifts rather than a general perturbation strategy. Other approaches such as Energy Redistribution Path Tracing [Cline et al. 2005], Multiple-Try MCMC [Nimier-David et al. 2019; Segovia et al. 2007], and Delayed Rejection [Rioux-Lavoie et al. 2020] all improve MCMC methods by proposing the use of multiple short chains or multiple attempts at creating a proposal distribution respectively.

[Kelemen et al. 2002] proposed Primary Sample Space MLT (PSSMLT), an alternative approach to path space MCMC which operates on the random numbers used to generate paths. This formulation had the advantage that it is substantially simpler than path space methods, yet could still explore local regions of path space. [Hachisuka et al. 2014] operates in a space of multiple paths, each of a different length, and selects the number of vertices to perturb from the light and camera. Multiple Importance Sampling is used to weight the path contributions of the selected combination of light and eye path. This splitting path space into paths of the same length is close to our work, and we discuss this more in Section 4.2.

Several other works have extended PSSMLT. Work such as [Li et al. 2015] and [Luan et al. 2020] used local path gradient information to generate anisotropic proposal distributions which are very effective at adapting proposals in path space. These are complementary to our approach which uses *global* information in image space to construct a proposal distribution. [Sawhney et al. 2022] recently proposed the use of PSS perturbations to decorrelate contributions in ReSTIR algorithms [Lin et al. 2022]. Path space MLT and PSSMLT were combined into a single space by several works [Bitterli et al. 2018; Otsu et al. 2017; Pantaleoni 2017], and again our approach is complementary to this work.

Path guiding uses information gathered during rendering or via a pre-pass to build distributions which match the integrand in the Rendering Equation [Kajiya 1986] better than purely sampling the BSDF and cosine term. This encompasses a large amount of work, from guided sampling of whole paths [Reibold et al. 2018], use of incident radiance stored on basis functions in world space [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], 5D Trees [Lafortune and Willems 1995; Müller et al. 2017] to selective sampling for specific transport phenomena [Fan et al. 2023; Yu et al. 2023]. Our work generates samples in image space, and the closest guiding approach is [Cline et al. 2008] who store information about recent sampling decisions in image space and use this to guide future sampling. This however is not directly applicable to our work as this uses a small cache of sampling decisions in image space, and is based on traditional Monte Carlo sampling rather than MCMC.

3 BACKGROUND

To begin, we start by introducing the path integral form of light transport upon which most rendering algorithms are based. This was described by [Hachisuka et al. 2014; Veach and Guibas 1997] as:

$$I_j = \int_{\mathcal{P}} h_j(\bar{x}) f(\bar{x}) d\mu(\bar{x}), \quad (1)$$

where the value of the j 'th pixel I_j in an image is given by integrating the contribution of a light path $f(\bar{x})$ weighted by a filter at the pixel $h_j(\bar{x})$. A path is defined as a series of path vertices $x_0..x_M$, and integration is typically performed with respect to the product area measure μ . The contribution of a path is then defined as the product of terms associated with interactions with a scene (note for notational convenience we index path vertices starting at the camera):

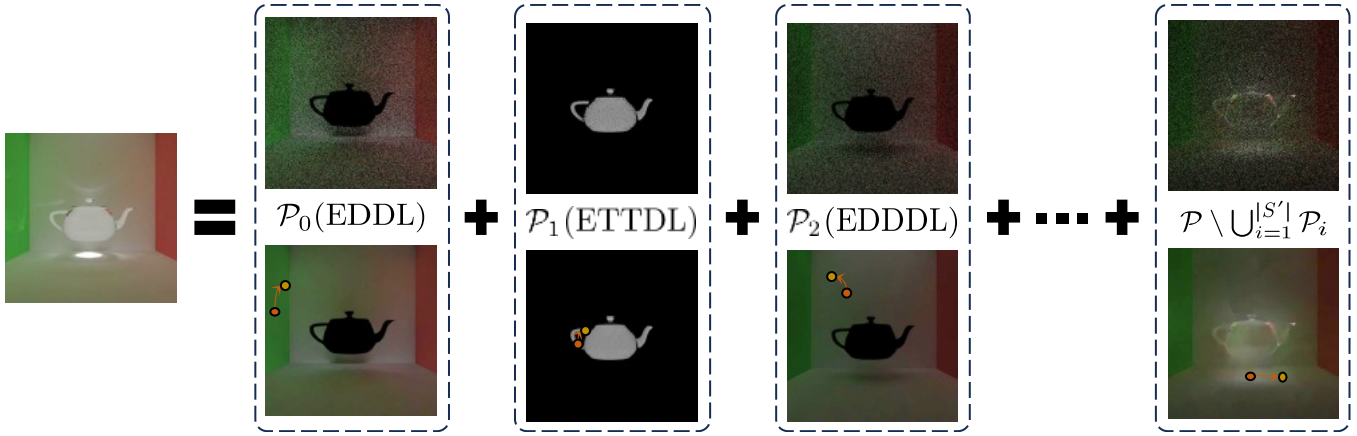


Figure 2: Our approach partitions the whole of path space (the left image) into a series of partitions, discussed in Section 4, each corresponding to a different subset of path space. We propose splitting on the interaction types, as illustrated in the boxes on the right, with the rightmost box illustrating the contribution from the complementary partition (Equation 6). Each of these partitions are constructed by performing a Monte Carlo sampling pre-pass to find the contribution of each found interaction type (top images, and discussed in Section 4.2). These contributions are denoised and are used to guide MCMC sampling (proposals illustrated in the denoised bottom images), see Section 5.

$$f(\bar{x}) = G(x_0 \leftrightarrow x_1) \left[\prod_{k=1}^{M-1} fr(x_{k-1} \rightarrow x_k \rightarrow x_{k+1}) G(x_k \leftrightarrow x_{k+1}) \right] Le(x_M), \quad (2)$$

where $fr(x_{k-1} \rightarrow x_k \rightarrow x_{k+1})$ is the BSDF and $G(x_k \leftrightarrow x_{k+1}) = \frac{\cos(\theta) \cos(\theta')}{\|x_k - x_{k+1}\|_2} V(x_k \leftrightarrow x_{k+1})$ is the Geometry Term where θ and θ' are the angles between the surface normals and the outgoing direction from the surface. $V(x_k \leftrightarrow x_{k+1})$ denotes visibility between two points.

This is typically integrated over the set of all paths \mathcal{P} which is defined as the union of all path lengths $\mathcal{P} = \bigcup_{i=2}^{\infty} \mathcal{P}(i)$ where $\mathcal{P}(i)$ are all paths of length i which connect the camera to the light source. This also indicates that paths can be used to integral all pixels in an image, but only paths for which $h_j(\bar{x}) > 0$ will contribute to the j 'th pixel.

Equation 1 can be solved with several numerical methods, but in this work we focus on MCMC approaches. Metropolis Sampling [Hastings 1970; Metropolis et al. 1953] starts from an initial state \bar{x} , and proposes a new tentative state \bar{x}' by sampling from a proposal distribution $T(\bar{x} \rightarrow \bar{x}')$ and the state is updated according to an acceptance probability:

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

where $f^*(\bar{x})$ is the scalar contribution function which maps RGB or spectral radiance to a scalar. This has been shown to generate states which are distributed proportional to f^* while allowing more flexibility than a traditional Monte Carlo estimator to explore state space. [Veach and Guibas 1997] used this to solve Equation 1 via the following estimator:

$$I_j \approx \frac{b}{N} \sum_{k=1}^N \frac{h_j(\bar{x}_k) f(\bar{x}_k)}{f^*(\bar{x}_k)}, \quad (4)$$

where b is a normalizing constant used to appropriately scale the histogram estimated by MCMC algorithms. b can be estimated via a separate Monte Carlo estimator: $b = \int_{\mathcal{P}} f^*(\bar{x}) d_{\mu}(\bar{x})$.

4 PATH SPACE PARTITIONING

Path space was defined in the previous sections as the union of paths of different lengths. While this is a general form of writing this, an alternative, and sometimes advantageous formulation is to partition path space into a set S' of K discrete subsets:

$$S' = \{\mathcal{P}_0, \mathcal{P}_1, \dots, \mathcal{P}_K\}. \quad (5)$$

To guarantee full coverage of path space, we need to augment this set with the remaining paths in path space not covered by this set (we refer to this as the complementary partition):

$$S = \{S', \mathcal{P} \setminus \bigcup_{i=1}^{|S'|} \mathcal{P}_i\}. \quad (6)$$

This means that for the i 'th partition of path space, Equation 1 can be written as:

$$I_{i,j} = \int_{\mathcal{P}_i} h_j(\bar{x}) f(\bar{x}) d_{\mu}(\bar{x}), \quad (7)$$

where each path $\bar{x} \in \mathcal{P}_i$. A MCMC estimator for Equation 9 can be written similar to Equation 4:

$$I_{i,j} \approx \frac{b_i}{N} \sum_{k=1}^N \frac{h_j(\bar{x}_k) f(\bar{x}_k)}{f^*(\bar{x}_k)}, \quad (8)$$

where b_i is defined as before but with respect to the partitioned path space $b_i = \int_{\mathcal{P}_i} f^*(\bar{x}) d\mu(\bar{x})$. The final value of the pixel is clearly then the sum over integrals over each partition:

$$I_j = \sum_{i=1}^{|S|} I_{i,j}. \quad (9)$$

Equation 10 can also be written as a Monte Carlo estimator where each partition is selected with probability $P(i)$:

$$I_j \approx \frac{1}{N} \sum_{k=1}^N \frac{b_i h_j(\bar{x}_k) f(\bar{x}_k)}{P(i) f^*(\bar{x}_k)}. \quad (10)$$

While this might initially seem to add unnecessary complication, there is an advantage to this formulation. MCMC algorithms transition between states proportional to their contribution. This means that the sampling algorithm will spend more time in states with a higher scalar contribution function than those with a lower contribution. While this is to be expected, in applications such as rendering this is suboptimal as scenes in which MCMC algorithms are effective often have widely varying values on the image plane. An example of this is path spaces which include specular interactions such as caustics which often lead to small regions of the scene having values which are orders of magnitude larger than the majority of the scene. As a result, the chain spends orders of magnitude more time in these regions, which comes at the cost of more variance in darker regions in an image.

Therefore, if we can partition path space into partitions which have similar contributions, then rather than chains spending a significant amount of time in brighter regions, each chain will explore its own reduced space. Chains in bright regions, with associated higher normalizing constant b_i , will only explore bright regions, and chains associated with darker regions will be able to expend more computation in these regions, thereby reducing variance in these regions.

4.1 1D Example

As an example of this, Figure 3 shows a 1D example of using partitioning. 3 a) and 3 b) show a function (red dashed line) being integrated using standard MCMC sampling without (a)) and with (b)) partitioning. Both use the same gaussian proposal distribution, but b) uses two chains, one for the left side of the partition (shown by the vertical line) and one for the right. Using two chains in this example allows one chain to explore the high contribution region on the right, and one to explore the left, low contribution region. This means that both regions are sampled adequately given the low sample count used in these figures. The horizontal lines indicate the value of the normalizing constant in each region. In contrast not using partitions leads to higher variance in both regions as the chain has explored the higher contribution region excessively at this low sample count. Note that both a) and b) converge to the correct value, but b) has 89 times less variance in this example.

4.2 Practical Considerations

Now we have a formalism to partition path space, this leaves three questions: how many partitions (K) should be used, how should

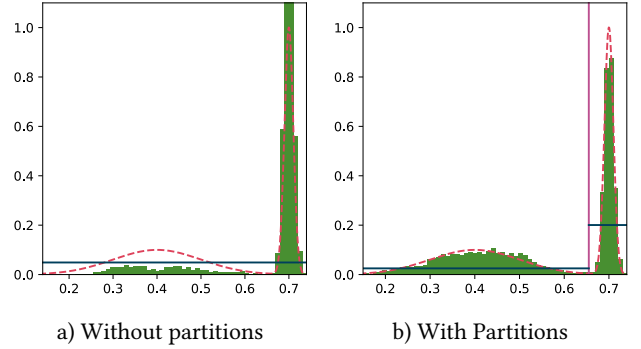


Figure 3: Example in 1D of using partitions. a) and b) show MCMC integration with and without partitioning (the partition is shown by the vertical line in b)). Using partitioning decreases variance significantly by allowing two chains to explore the lower and higher contributions separately, each of which uses a different normalizing constant (the horizontal line).

path space be partitioned, and with what probability should each partition be sampled ($P(i)$)?

There has been some previous work which has considered this problem, i.e. [Hachisuka et al. 2014], [Bashford-Rogers et al. 2021] and [Lin et al. 2022]. [Hachisuka et al. 2014] is the most similar as they explicitly integrate in a set of partitions of different path lengths, i.e.

$$S_h = \{\mathcal{P}_0, \mathcal{P}_1, \dots, \mathcal{P}_K\}, \quad (11)$$

where \mathcal{P}_i denotes a path space with $i + 4$ vertices. Each of these partitions was sampled proportional to its contribution, and path lengths up to $K + 4$ could be computed. However, while this solution is unbiased and quite effective for these paths, this neglects the rest of path space from Equation 6, and faces the same issue that paths of the same length may explore regions of path space with contributions of significantly different orders of magnitude. For example, using Heckbert path notation [Heckbert 1990], a path of length 5 may include both low contribution *LDDDE* paths, but very high contribution *LSSDE* paths.

We propose a solution to the three questions. Like [Hachisuka et al. 2014], we use a Monte Carlo estimator as a pre-pass and trace a number of paths. Each subpath can be viewed as estimating a separate integral, i.e. estimating a value of a partition of path space, as the path is progressively constructed. Therefore this pre-pass gives us much of the information needed to partition path space if we can efficiently extract this information. At this point any information about each sub-path could be used, for example interaction types which classify vertices into specular or diffuse akin to Heckert notation, regions of the scene or image space explored by the sub path etc. We take the approach of storing paths corresponding to a unique set of interactions as specified by their interaction types.

Practically, we store a linked list of buffers B corresponding to each interaction type (i.e. *LDDE*, *LSDE*, *LDDDE*), and accumulate the contribution of this sub path ($C(\bar{x}_k) = \frac{f(\bar{x}_k)}{p(\bar{x}_k)}$, where $p(\bar{x})$ is the

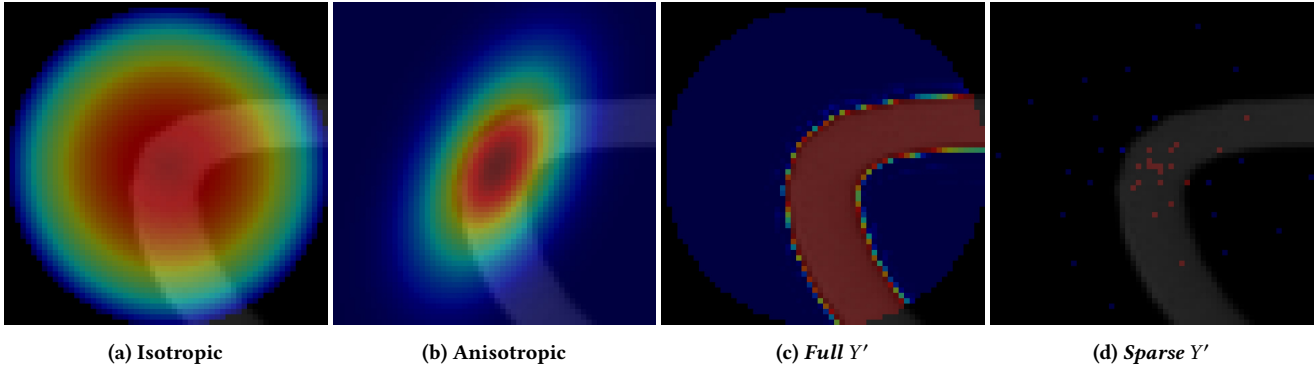


Figure 4: Illustration of different image space proposal distributions for a region in a partition from Figure 2, red regions indicate higher and blue lower values of the proposal distribution. Isotropic approaches (4a) do not use any information and can propose paths that are likely to be rejected. Anisotropic proposals (4b) adapt better to the distribution of radiance but are still limited to a parametric distribution. Figure 4c shows our approach which adapts *per-pixel* to the estimated lighting based on the denoised estimate of lighting per partition D_i , and Figure 4d shows our sparse approximation.

pdf of generating this path), the seed used to generate this sub-path. We store this as a linked list to minimize the memory requirements associated with the combinatorial explosion of interaction types in a complicated scene.

As each buffer defines a partition of path space, we can map some attributes from each buffer to a scalar and can calculate the K most important partitions for use in Equation 5. The mapping can use any heuristic function, however, we want to both consider the contribution of the paths in the partition as well as the area of the image plane over which the paths explore. The reason for this is that we want to explore large, low contribution regions as well as small high contribution regions. Therefore, we choose to consider the total contribution of half the paths $|B|/2$ (the other half is used for initialization of the MCMC algorithm) in the i 'th partition ($\gamma(i) = \sum_{k=1}^{|B|/2} C(\bar{x}_k)$) as this accounts for both the contributions of the paths and the area of the image covered by these paths.

We can then sort the buffers by γ and choose the K largest which fit within the memory requirements as discussed in Section 5. The remaining paths are assigned to the complementary partition (see Equation 6). The partition is then defined by the path interaction types in the partition and the probability of choosing a partition is then $P(i) = \frac{\gamma(i)}{\sum_{k=1}^K \gamma(k)}$.

We now need to compute the normalization constant and initialize each MCMC chain in each partition. To estimate both, we split the initial samples used into two sets of size $|B|/2$. One set is used to estimate the partitions as discussed above, and the contributions from the other are used to compute the normalizing constant for the partition (b_i). Furthermore, an initial path is resampled from this other set, and we also run burn-in [Brooks et al. 2011] for 1024 iterations to further reduce startup bias.

5 GUIDED IMAGE SAMPLING

Partitioning path space as described in the previous section leads to a discrete set of regions of path space that will be explored by a chain. However, this partitioning of path space leads to sparser regions of path space which contain valid contributions on the

image plane. In image space, this often means that paths from one partition can only contribute within a small region, but paths from other partitions may contribute widely over the image plane. If the original proposal distributions defined in MLT or PSSMLT are used, these are not likely to be able to explore these spaces well. Motivated by this, we investigate guiding perturbations on the image plane such that both small, sparse regions can be explored effectively, while also having the ability to widely and efficiently explore the image plane.

We propose to solve this via guiding perturbations on the image plane. There are two main reasons for choosing the image plane for guidance rather than the whole of path space. Firstly, in Path Space MLT [Veach and Guibas 1997], for most scenes the lens perturbation is typically responsible for the majority of the variance reduction. Secondly, several other algorithms in rendering exploit combining sparser estimates of suffix or light paths with a denser sampling of prefix or camera paths, such as Instance Radiosity [Keller 1997] and final gathering for photon mapping [Jensen 2001] or conditional ReSTIR [Kettunen et al. 2023], which again shows that well converged results can be obtained by focusing sampling on the image space.

We next discuss guided sampling in path space and formulate practical guidance distributions for both spaces.

5.1 Path Space Image Plane Guiding

To generate guided samples on the image plane that can still explore the local region around the current path requires a proposal distribution which is aware of the contributions of a path to the image plane in a local region. States will still be visited proportional to their contribution, but we aim to derive a proposal distribution that can increase the probability of moving between states based on their contribution rather than a fixed proposal distribution.

We start by considering the acceptance probability computation given in Equation 3. Perturbing a path in image space means that we are perturbing vertices successively until they can be connected

to a fixed remainder of a path which starts at the s 'th vertex from the camera, which has the contribution:

$$\alpha(\bar{x}) = G(x_s \leftrightarrow x_{s+1}) \left[\prod_{k=s+1}^{M-1} fr(x_{k-1} \rightarrow x_k \rightarrow x_{k+1}) G(x_k \leftrightarrow x_{k+1}) \right] Le(x_M). \quad (12)$$

Therefore, the prefix path we want to perturb has the following contribution:

$$S(\bar{x}) = G(x_0 \leftrightarrow x_1) \left[\prod_{k=1}^s fr(x_{k-1} \rightarrow x_k \rightarrow x_{k+1}) G(x_k \leftrightarrow x_{k+1}) \right] fr(x_{s-1} \rightarrow x_s \rightarrow x_{s+1}). \quad (13)$$

The acceptance probability can now be written as:

$$a(\bar{x} \rightarrow \bar{x}') = \min \left(1, \frac{S^*(\bar{x}') \alpha^*(\bar{x}') T(\bar{x}' \rightarrow \bar{x})}{S^*(\bar{x}) \alpha^*(\bar{x}) T(\bar{x} \rightarrow \bar{x}')} \right), \quad (14)$$

Therefore, $T(\bar{x} \rightarrow \bar{x}') \propto S^*(\bar{x}')$ then states would be generated exactly proportional to their contribution up to a normalizing constant. This is challenging to evaluate analytically in a potentially wide region around the current path in image space. However, if a set of candidate prefix paths $Y' = \{\bar{x}'_0, \bar{x}'_1, \dots, \bar{x}'_{|Y'|}\}$, each with contribution $Q' = \{S^*(\bar{x}'_0), S^*(\bar{x}'_1), \dots, S^*(\bar{x}'_{|Q'|})\}$ can be generated, then a candidate path \bar{x}' can be sampled from Y' proportional to Q' , thereby closely approximating the ideal distribution in a local region.

As we are perturbing in image space, the logical domain on which to create the set of paths is over pixels in the local image space neighborhood of the current path \bar{x} . Therefore, all we need is to evaluate S^* for the subset of pixels. However, even though $S(\bar{x})$ does not involve many terms, it still requires, at a minimum, evaluating two BSDFs and two geometry terms (also considering that each geometry term includes a visibility test). However, we can leverage the Monte Carlo pre-pass from Section 4.2 to substantially reduce the computation requirements by making a series of approximations.

Firstly, we assume that the last BSDF evaluation will be similar for all paths in Y' , and set $fr(x_{s-1} \rightarrow x_s \rightarrow x_{s+1}) = 1$. Secondly, as the BSDFs in a local region often do not vary much we approximate all other BSDF evaluations as a diffuse BRDF, i.e. $\frac{\rho_j}{\pi}$, where ρ_j is the albedo of the first non-specular vertex visible through pixel j . Thirdly, we can precompute $G(x_0 \leftrightarrow x_1)$ for all pixels as G_j . Finally, we use an approximate value for the visibility term (V'_j in the final Geometry Term, which allows us to avoid tracing any visibility rays. We discuss this in Section 5.1.1.

We can then use this information to approximate $S(\bar{x})$ for the j 'th pixel in the set Y' :

$$S'(\bar{x}_j) \approx G_j \frac{\rho_j \cos(\theta_j) \cos(\theta')}{\pi \|x_{s-1} - x_s\|_2} V'_j, \quad (15)$$

where θ_j is the angle between a stored normal at the pixel j and the outgoing direction. This leads to the final acceptance probability for the proposed guided image perturbation:

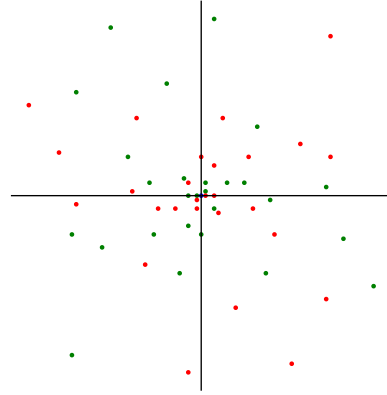


Figure 5: Sparse offsets used to compute Y' . Green points are the arbitrarily chosen initial points and red are the inverse to guarantee reversibility.

$$a(\bar{x} \rightarrow \bar{x}') = \min \left(1, \frac{S^*(\bar{x}') \frac{S'^*(\bar{x})}{S^*(\bar{x})} \sum_{k=1}^{|Y'|} S'^*(\bar{x}'_k)}{S^*(\bar{x}) \frac{S'^*(\bar{x}')}{S^*(\bar{x}')} \sum_{k=1}^{|Y'|} S'^*(\bar{x}_k)} \right), \quad (16)$$

The ratio of sums at the end of this expression comes from normalization constants that depend on both the set Y' evaluated at pixels around \bar{x} , and also another set Y which is evaluated at pixels around the proposed path \bar{x}' . This adds extra computation, so the size of Y' must be chosen carefully to minimize computation cost.

We now discuss how to build V'_j , and which pixels to use for the set Y' .

5.1.1 Building Image Space Approximations. To build the image space approximations for G_j , ρ_j , and V'_j we use the fact that we have stored the set of paths used to generate each partition, and that we can easily store other attributes such as image space position X_j , normal n_j , and albedo ρ_j of the first non-specular vertex in a path in a GBuffer. G_j and ρ_j are then easy to precompute and store in an image space buffer, however, the visibility approximation is harder.

If we assume that the visibility will be similar in a local region of the image plane, then we can estimate an image space map of visibility for each of the pixels where paths from the partition can contribute. To achieve this, we leverage image space denoising (we use [Áfra 2023]) of the paths from the Monte Carlo prepass which were used to estimate the partition. Note that any image space denoiser can be used for this, but instead of denoising all pixels in an image, the denoiser needs to focus on the smaller subset of pixels that have a non-zero contribution. This results in a denoised image for each partition D_i . We then directly leverage this to compute the visibility approximation:

$$V'_j = \begin{cases} 1 & \text{if } D_{i,j} > \epsilon \\ \epsilon & \text{otherwise} \end{cases} \quad (17)$$

5.1.2 *Choosing Y'* . We also have substantial freedom to choose the pixels which will form the elements of Y' given one condition: the choice of elements of Y' must preserve detailed balance, specifically the reversibility condition. Another way of saying this is that $T(\bar{x}' \rightarrow \bar{x})$ must be greater than zero for any proposed state \bar{x}' . An obvious approach could be to use all pixels within a fixed radius R around and including the current state \bar{x} , referred to as *Full Y'* . This guarantees that there is a non-zero density for reaching any pixel from any other pixel within R . We illustrate this in Figure 4 which shows the difference between using isotropic proposal distributions, anisotropic proposals, and our approach.

This approach leads to a proposal distribution which adapts to the estimate of the illumination within a local region. However, computing the set of contributions Q' is computationally expensive if all pixels within the region R are considered. This cost can be substantially reduced by only considering a sparse approximation to *Full Y'* which obeys the above condition.

We describe the set of shifts in image space, measured in pixels, required to construct Y' as a set of 2D offsets from the image plane coordinates of the current path:

$$\Delta = \{(\delta.x, \delta.y)_0, (\delta.x, \delta.y)_1, \dots, (\delta.x, \delta.y)_{|Y'|}\}. \quad (18)$$

The first $\frac{|Y'|-1}{2}$ offsets in this set can be chosen arbitrarily, for example from a uniform sampling of a disk of radius R . The middle element needs to be $(0, 0)$, and the remaining elements of Δ must be set as the inverse of the first $\frac{|Y'|-1}{2}$ elements, e.g. $(\delta.x, \delta.y)_{\frac{|Y'|-1}{2}+k+1} = (-\delta.x, -\delta.y)_k, \forall k \leq \frac{|Y'|-1}{2}$. This inversion guarantees that the initial state can be reached from any proposed state. We find it advantageous to propose more states closer to the current path, so we create the initial $\frac{|Y'|-1}{2}$ offsets using a low discrepancy sequence which generates points $(\zeta.x, \zeta.y)$ which we mapped to the disk using a non-uniformity preserving mapping to polar coordinates $(r, \Theta) = (\zeta.x^2, 2\pi\zeta.y)$ which are mapped to pixel shifts in Δ . This is illustrated in Figure 5 which shows the initial $\frac{|Y'|-1}{2}$ points in green, the center in blue, and the inverse set as red.

6 RESULTS

We show results for our approach when applied to path space MLT [Veach and Guibas 1997]. We chose this algorithm for our baseline comparisons as it is the main path space MCMC integration algorithm, and as noted in Section 2, our method is complementary to most approaches which extend these works. We used a CPU prototype implementation of all algorithms, and generated results on a laptop with a i9-11980 CPU and 32GB RAM. All methods use the same number of samples for initialization (computation of the normalization constants and resampling an initial path), but ours has an extra computational overhead of creating the denoised images, although this took a maximum of two seconds for all our results. We used the Intel Open Image Denoiser [Áfra 2023] to denoise our images; and although this is not trained on denoising partitioned path space contributions on the image space, we found that it performed well. Our method has a linear memory cost in terms of the number of partitions, consisting of a tiny amount of extra memory to store per-partition chain information, and a larger resolution dependent buffer for use when evaluating V'_j in Equation

15 This added around 80MB memory usage for all scenes in our results. We use 10 partitions plus the complementary partition in all our results as we found this allows the important majority of lighting to be found for all scenes, and use $|Y'| = 128$, as we found this balanced variance reduction with computational cost (our unoptimized implementation has a 19% overhead compared to MLT)

All scenes were rendered on average at 32 mutations per pixel. Figure 1 shows the *bathroom* scene, where our method has an RMSE of 0.4462 and MLT 0.5788, an 29% improvement. Figure 7 compares our approach (RMSE 0.00816) with MLT (RMSE 0.01156), Ensemble Metropolis Light Transport (EMLT 0.00742) [Bashford-Rogers et al. 2021] and Geometry Aware MLT (GAMLT 0.00927) [Otsu et al. 2018] with the extensions proposed in [Bashford-Rogers et al. 2021]. While EMLT slightly outperforms our method, we remain competitive both in terms of objective metrics and visually as can be seen in the figure.

Figure 8 shows the *door-ajar* and *kitchen* scenes which have an RMSE of 0.0811 (Ours) vs 0.0924 (MLT) and 0.0222 (Ours) vs 0.0225 (MLT) for the scenes respectively. Our approach leads to better convergence by distributing samples between partitions, as is highlighted by the unbalanced sampling of glass teapot in MLT in the *door-ajar* scene at this low sample number compared to our method which leads to more equal coverage of the partitions.

Finally, we show the impact of different sizes of $|Y'|$ and different with different radii in Figure 6, which shows that too few points in $|Y'|$ leads to structured noise, and a too large radius leads to more high frequency noise.

Overall the results show that our method leads to an overall reduction in variance, however there can be increased noise compared to MLT in local regions of the scene. This is due to the sparse nature of Y' , and future work will investigate a more optimal set of values for Y' to reduce this variance.

7 CONCLUSION

This paper proposed a principled approach to partition path space into a discrete set of subspaces. We propose an automated algorithm to choose these partitions, and an image space MCMC proposal distribution which can explore these spaces efficiently. This distribution is based on an analysis of terms in the acceptance probability and can be efficiently approximated via the use of denoising the initial samples used to create the partitions. Our results show an improvement in variance for the same number of samples across several scenes.

REFERENCES

- Attila T. Áfra. 2023. Intel® Open Image Denoise. <https://www.openimagedenoise.org>.
- 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.
- Thomas Bashford-Rogers, Luís Paulo Santos, Demetris Marnerides, and Kurt Debattista. 2021. Ensemble metropolis light transport. *ACM Transactions on Graphics (TOG)* 41, 1 (2021), 1–15.
- 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.
- Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. 2011. *Handbook of markov chain monte carlo*. CRC press.
- David Cline, Daniel Adams, and Parris Egbert. 2008. Table-driven adaptive importance sampling. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 1115–1123.

- David Cline, Justin Talbot, and Parris Egbert. 2005. Energy redistribution path tracing. In *ACM Transactions on Graphics (TOG)*, Vol. 24. ACM, 1186–1195.
- 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.
- Zhimin Fan, Pengpei Hong, Jie Guo, Changqing Zou, Yanwen Guo, and Ling-Qi Yan. 2023. Manifold path guiding for importance sampling specular chains. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 1–14.
- 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. (1970).
- 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 Krivánek. 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.
- 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.
- Henrik Wann Jensen. 2001. *Realistic image synthesis using photon mapping*. Vol. 364. Ak Peters Natick.
- 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., USA, 49–56. <https://doi.org/10.1145/258734.258769>
- Markus Kettunen, Daqi Lin, Ravi Ramamoorthi, Thomas Bashford-Rogers, and Chris Wyman. 2023. Conditional Resampled Importance Sampling and ReSTIR. In *SIGGRAPH Asia 2023 Conference Papers*. 1–11.
- Eric P LaFortune and Yves D Willems. 1993. Bi-directional path tracing. (1993).
- 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.
- 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.
- Daqi Lin, Markus Kettunen, Benedikt Bitterli, Jacopo Pantaleoni, Cem Yuksel, and Chris Wyman. 2022. Generalized resampled importance sampling: foundations of ReSTIR. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–23.
- Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. 2020. Langevin Monte Carlo rendering with gradient-based adaptation. *ACM Trans. Graph.* 39, 4 (2020).
- Marco Manzi, Fabrice Rousselle, Markus Kettunen, Jaakko Lehtinen, and Matthias Zwicker. 2014. Improved sampling for gradient-domain metropolis light transport. *ACM Transactions on Graphics (TOG)* 33, 6 (2014), 1–12.
- 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.
- 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.
- Jacopo Pantaleoni. 2017. Charted metropolis light transport. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 75.
- 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 Trans. Graph.* 39, 4, Article 147 (July 2020), 15 pages. <https://doi.org/10.1145/3386569.3392421>
- Rohan Sawhney, Daqi Lin, Markus Kettunen, Benedikt Bitterli, Ravi Ramamoorthi, Chris Wyman, and Matt Pharr. 2022. Decorrelating restir samplers via mcmc mutations. *ACM Transactions on Graphics* (2022).
- Benjamin Segovia, Jean Claude Lehl, and Bernard Péroche. 2007. Metropolis instant radiosity. In *Computer Graphics Forum*, Vol. 26. Wiley Online Library, 425–434.
- Eric Veach and Leonidas Guibas. 1995. Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques*. Springer, 145–167.
- 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 Krivánek. 2014. On-line learning of parametric mixture models for light transport simulation. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 101.
- Rui Yu, Yue Dong, Youkang Kong, and Xin Tong. 2023. Neural Path Sampling for Rendering Pure Specular Light Transport. In *Computer Graphics Forum*. Wiley Online Library, e14997.

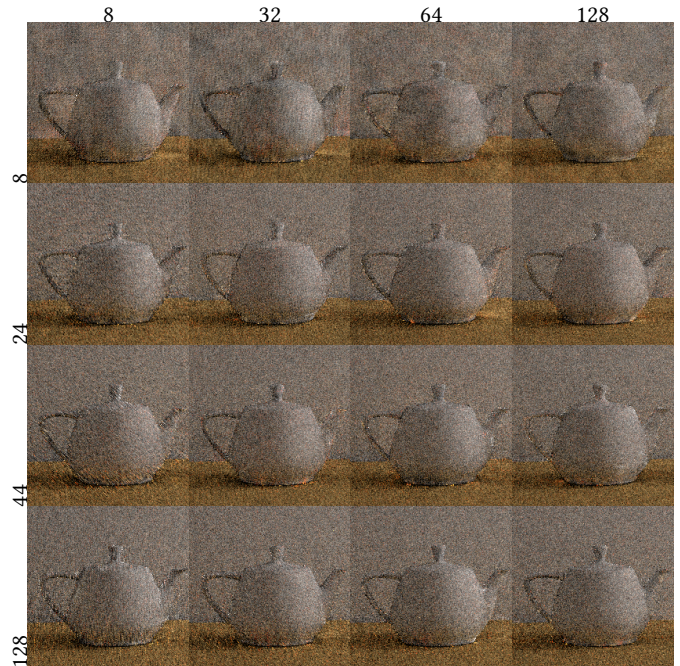


Figure 6: Impact of the size and radius of the points in Y' . The rows show different radii, and the columns show the numbers of points.

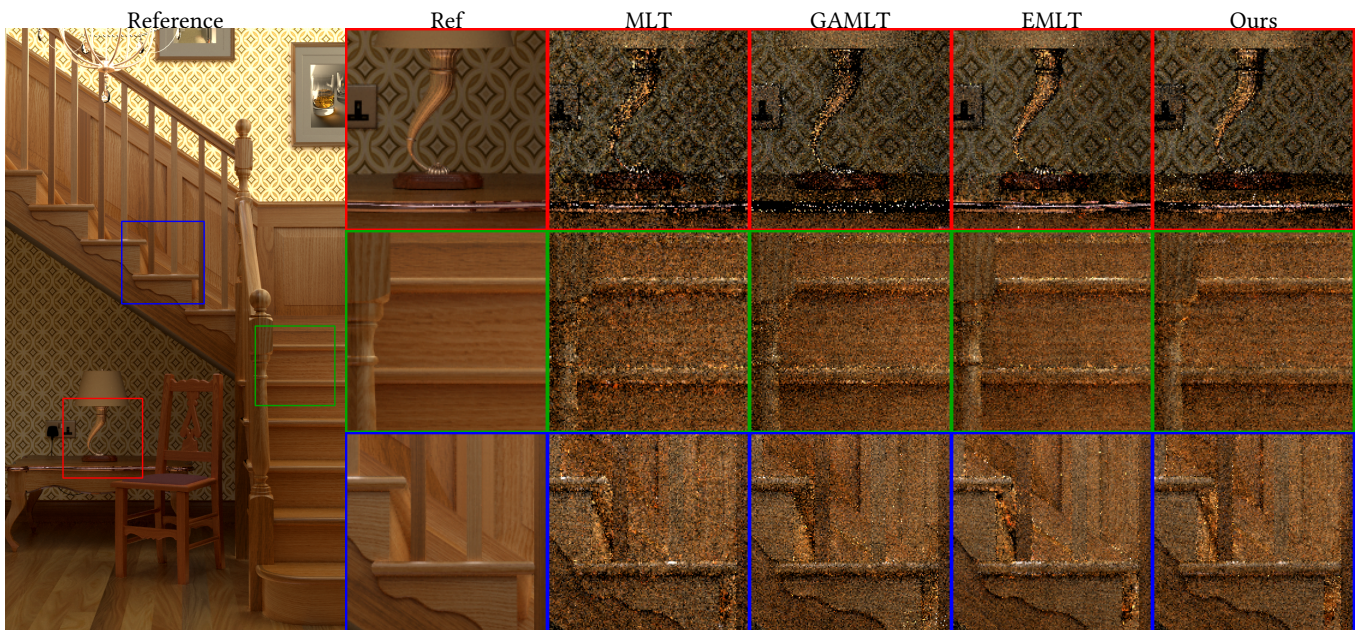


Figure 7: Results for the *staircase* scene comparing MLT, Geometry Aware MLT (GAML) with extensions proposed in [Bashford-Rogers et al. 2021], Ensemble Metropolis Light Transport (EMLT) and Ours.

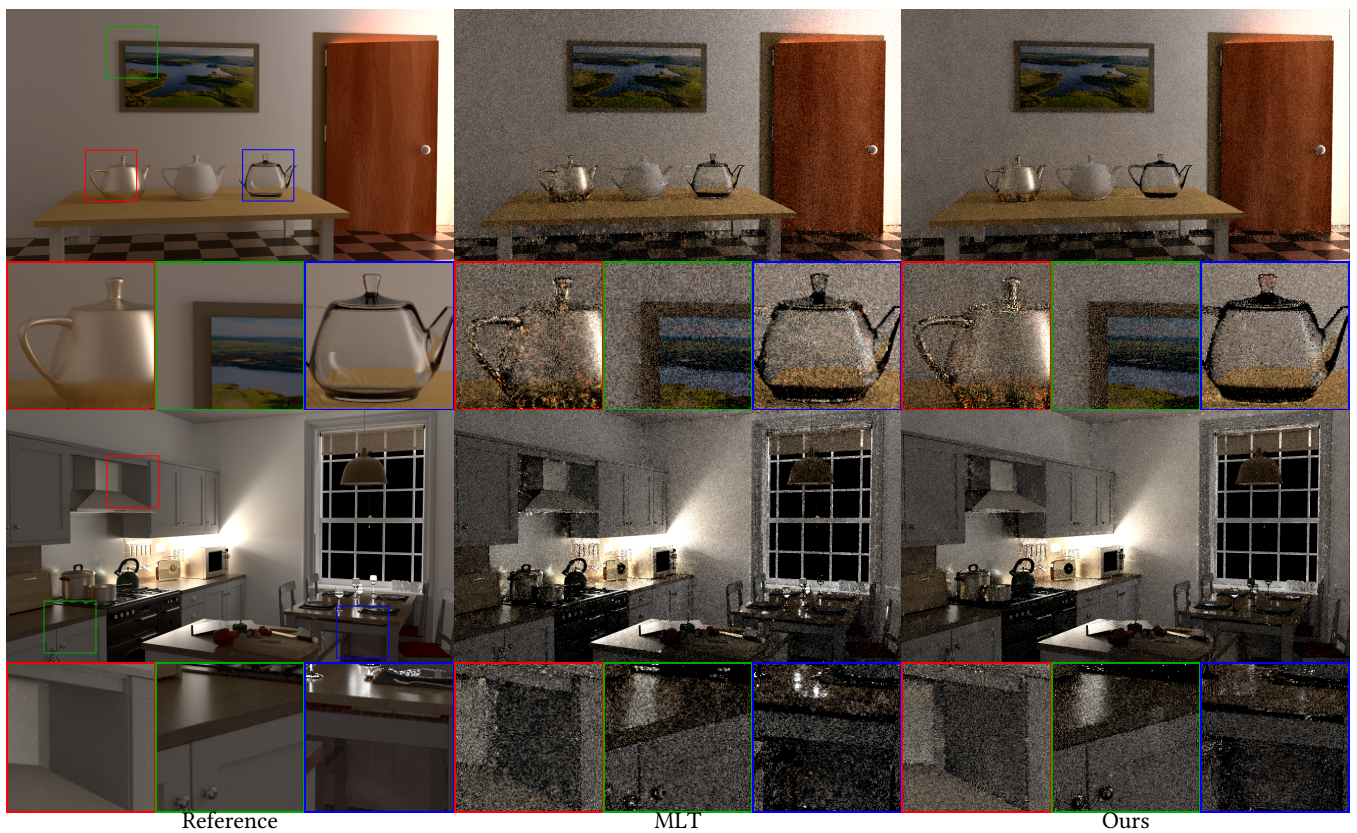


Figure 8: Results for the *door-ajar* and *kitchen* scenes showing the reference (left) MLT (middle), and our method (right). Our approach reduces variance by distributing samples between partitions and the use of guided image perturbations.