DOI: 10.1111/cgf.13935
EUROGRAPHICS 2020 / U. Assarsson and D. Panozzo
(Guest Editors)

Volume 39 (2020), Number 2

Stratified Markov Chain Monte Carlo Light Transport

Adrien Gruson[1,2] Rex West[1] Toshiya Hachisuka[1]

[1]The University of Tokyo, Japan [2]McGill University, Canada

Figure 1: *Our stratified MCMC shows less noise than primary sample space MLT (PSSMLT) [KSAC02] in the same rendering time (30 minutes). PSSMLT gets stuck exploring only a portion of the image while our technique produces more predictable results. This translates to a lower mean absolute percentage error (MAPE) across the whole image.*

Abstract

Markov chain Monte Carlo (MCMC) sampling is a powerful approach to generate samples from an arbitrary distribution. The application to light transport simulation allows us to efficiently handle complex light transport such as highly occluded scenes. Since light transport paths in MCMC methods are sampled according to the path contributions over the sampling domain covering the whole image, bright pixels receive more samples than dark pixels to represent differences in the brightness. This variation in the number of samples per pixel is a fundamental property of MCMC methods. This property often leads to uneven convergence over the image, which is a notorious and fundamental issue of any MCMC method to date. We present a novel stratification method of MCMC light transport methods. Our stratification method, for the first time, breaks the fundamental limitation that the number of samples per pixel is uncontrollable. Our method guarantees that every pixel receives a specified number of samples by running a single Markov chain per pixel. We rely on the fact that different MCMC processes should converge to the same result when the sampling domain and the integrand are the same. We thus subdivide an image into multiple overlapping tiles associated with each pixel, run an independent MCMC process in each of them, and then align all of the tiles such that overlapping regions match. This can be formulated as an optimization problem similar to the reconstruction step for gradient-domain rendering. Further, our method can exploit the coherency of integrands among neighboring pixels via coherent Markov chains and replica exchange. Images rendered with our method exhibit much more predictable convergence compared to existing MCMC methods.

1. Introduction

Research on how to handle complex light transport in Monte Carlo (MC) rendering has received significant attention in recent years. Markov chain Monte Carlo [VG97, KSAC02, HKD14] is particularly a powerful concept which allows us to take all of the factors (such as visibility and BSDFs) into account when sampling light transport paths. It works by storing a current path and exploring the path-space locally by mutating the stored path. MCMC is currently the only approach that can generate a path according to its path throughput under an arbitrary scene configuration. This property enables MCMC to handle difficult configurations such as strong indirect illumination coming from a small gap.

Rendering an image using MCMC is fundamentally different from MC. In MC, each pixel has an integral defined by the path integral formulation, and the integral is estimated using random samples per pixel. Since the integral of each pixel is estimated independently with the same number of samples for all of the pixels, the image converges in a predictable manner. In contrast, each

pixel in MCMC is not estimated independently. Instead, an image is rendered by a single sequence of samples which covers the entire image according to the *random walk* defined by the Markov chain.

In other words, the number of samples in the image-space is not *stratified* in MCMC. Each pixel thus does not receive the same number of samples, especially when the total number of samples is *finite*. This lack of image-space stratification results in unpredictable convergence that can be commonly seen in MCMC rendering techniques. This behavior prevents a wider adoption of MCMC techniques in the industry [CJ16] despite its powerful ability to handle complex light transport paths.

We propose a MCMC method which achieves, for the first time, *perfect* image-space stratification with a finite number of samples. Just like path tracing, all the pixels receive exactly the same number of samples in our method. Our method is motivated by a naive approach of stratifying MCMC: an independent Markov chain per pixel. While this naive approach achieves stratification, it is not useful since each pixel needs to be scaled by a constant which is equal to the integral we want to solve in the first place. Our key observation is that we can entirely remove this need for the scaling factors by introducing overlapping regions around each pixel. In those overlapping regions, different Markov chains should converge to the same value.

This observation leads to a simple two-step algorithm: we run many Markov chains in parallel at each pixel considering the overlaps, and simply scale each pixel so that the overlapping pixel values from different Markov chains match. We show that this process can be formulated as an optimization problem similarly to the reconstruction step of gradient-domain rendering [LKL*13], at the cost of negligible bias. Similar to shift mapping in gradient-domain rendering, we use coherent Markov chains to estimate a set of neighboring pixel values. We also take advantage of the coherency of neighboring pixels to perform replica exchange between their chains with a high acceptance rate. This greatly helps facilitate the exploration of the short chains of our method. To summarize, our contributions are:

- First MCMC with guaranteed stratification in the image-space even when the total number of samples is *finite*.
- Reconstruction of scaling factors as an optimization problem using overlapping pixels.
- Improved mixing of chains via image-local replica exchange.

Figure 1 illustrates the performance of our method compared to a classical MCMC approach [KSAC02]. The classical MCMC approach exhibits a high level of noise in under-sampled regions due to uneven exploration in image-space. Moreover, as a natural consequence of its exploration process, the classical MCMC approach can get stuck in high contribution regions of image-space, resulting in unpredictable convergence. Our method does not suffer from such fundamental issues of MCMC and achieves more predictable convergence with perfect stratification in the image plane.

2. Stratification in MC and MCMC

We recapitulate light transport simulation using MC and MCMC to clarify why stratification is missing in MCMC. Although stratification in a high-dimensional space has been explored in the context of MC integration [SNJ*14, CKK18], none of these methods are applicable to MCMC to date. We thus focus on introducing image-space (2D) stratification as the first step which is also important as we explain later. Readers familiar with MC and MCMC in rendering can skip to Section 2.4 for the problem statement.

2.1. Monte Carlo Integration in Light Transport

The path integral formulation [Vea97] states that the intensity I_j of the pixel j is defined as

$$I_j = \int_{\mathcal{P}} h_j(\mathbf{x}) f(\mathbf{x}) \mathrm{d}\mu(\mathbf{x}) \tag{1}$$

where $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1, \ldots)$ is a path, f is the measurement contribution function, h_j is a filter function which determines the contribution of $\mathbf{x}$ to the pixel j, and $\mathcal{P}$ is the path-space. We used the notation I_j to clarify that I is a function of the pixel index j. To simplify the explanation below, we assume a box filter for $h_j(\mathbf{x})$ where $h_j(\mathbf{x}) = 1$ when $\mathbf{x}$ goes through the pixel j and $h_j(\mathbf{x}) = 0$ otherwise.

MC integration estimates each integral I_j by randomly sampling N paths $\mathbf{X}_i$ according to a user-defined probability density function $p(\mathbf{x})$. The resulting MC estimator $\hat{I}_j$ is defined as

$$\hat{I}_j = \frac{1}{N} \sum_{i=1}^{N} \frac{h_j(\mathbf{X}_i) f(\mathbf{X}_i)}{p(\mathbf{X}_i)}. \tag{2}$$

In many MC rendering algorithms, we estimate each I_j independently from other pixels by generating samples $\mathbf{X}_i$ that are distributed only to the pixel j. Note that the number of samples N for each pixel is controllable independently of f, p, and h. It is common to use a constant N for all of the pixels to achieve a perfectly stratified and uniform distribution of samples in the image-space.

2.2. Metropolis Light Transport

The expected squared error of the MC estimator $\hat{I}_j$ is proportional to the variance of $h_j(\mathbf{x}) f(\mathbf{x}) / p(\mathbf{x})$. We thus generally want $p(\mathbf{x})$ to be close to proportional to $h_j(\mathbf{x}) f(\mathbf{x})$, but defining and generating samples from such a probability density function is non-trivial.

Metropolis light transport [VG97] (MLT) allows us to generate samples directly according to a probability density function proportional to $h_j(\mathbf{x}) f(\mathbf{x})$ based on the Metropolis-Hastings algorithm [Has70]. Given an arbitrary positive scalar function $T(\mathbf{x})$, the algorithm generates a series of samples $\mathbf{X}_1, \mathbf{X}_2, \ldots$ that follows the distribution (often called a target distribution) $p(\mathbf{x}) \propto T(\mathbf{x})$ by iterating the following steps:

1. Generates a proposal $\mathbf{Y}$ from a proposal distribution $g(\mathbf{Y}|\mathbf{X}_i)$ given the current sample $\mathbf{X}_i$. For simplicity, we assume that the proposal distribution is symmetric ($g(\mathbf{Y}|\mathbf{X}_i) = g(\mathbf{X}_i|\mathbf{Y})$).
2. Accepts the proposal with probability $\min\left(1, \frac{T(\mathbf{Y})}{T(\mathbf{X}_i)}\right)$.
   - If $\mathbf{Y}$ was accepted, it becomes the next sample $\mathbf{X}_{i+1}$.
   - If $\mathbf{Y}$ was rejected, the current sample $\mathbf{X}_i$ becomes the next sample $\mathbf{X}_{i+1}$

The probability density function $p(\mathbf{x})$ of generated samples in this algorithm converges to

$$p(\mathbf{x}) = \frac{T(\mathbf{x})}{\int_P T(\mathbf{x}')d\mu(\mathbf{x}')} = \frac{T(\mathbf{x})}{b} \tag{3}$$

where b is called the normalization factor. In a typical application of the Metropolis-Hastings algorithm, samples are assumed to be well distributed according to $p(\mathbf{x})$ after a certain number of iterations (called *burn-in*). The probability density function $p(\mathbf{x})$ is the distribution of samples at such an equilibrium state.

Let us consider such a sequence of samples $\mathbf{X}_1, \mathbf{X}_2, \ldots$ with $T(\mathbf{x}) = h_j(\mathbf{x})f(\mathbf{x})$. Since samples $\mathbf{X}_1, \mathbf{X}_2, \ldots$ can be considered distributed according to $p(\mathbf{x}) \propto T(\mathbf{x}) = h_j(\mathbf{x})f(\mathbf{x})$, it is tempting to use them in Equation 2. This estimator is not useful since

$$\begin{aligned}\hat{\mathcal{I}}_j &= \frac{1}{N}\sum_{i=1}^{N}\frac{h_j(\mathbf{X}_i)f(\mathbf{X}_i)}{p(\mathbf{X}_i)} \\ &= \frac{1}{N}\sum_{i=1}^{N}\frac{h_j(\mathbf{X}_i)f(\mathbf{X}_i)}{b^{-1}h_j(\mathbf{X}_i)f(\mathbf{X}_i)} = \frac{1}{N}\sum_{i=1}^{N} b = b\end{aligned} \tag{4}$$

and b is exactly the path integral formulation (Equation 1) that we are trying to solve in the first place. In the following, we use $\hat{\mathcal{I}}_j$ for MCMC to distinguish it from the MC estimator $\hat{I}_j$.

MLT in fact is not using such a target distribution, but generates samples according to the *sum* of all $h_j(\mathbf{X})f(\mathbf{X})$ as $T(\mathbf{X}) = \sum_{j'} h_{j'}(\mathbf{X})f(\mathbf{X})$ which is *shared among all the pixels*. Samples $\mathbf{X}_i$ can now visit any pixel in the image-space, not only the jth pixel. Considering a pixelwise box-filter for $h_j(\mathbf{X})$, the estimator becomes

$$\begin{aligned}\hat{\mathcal{I}}_j &= \frac{1}{N}\sum_{i=1}^{N}\frac{h_j(\mathbf{X}_i)f(\mathbf{X}_i)}{b^{-1}\sum_{j'} h_{j'}(\mathbf{X}_i)f(\mathbf{X}_i)} \\ &= \frac{1}{N}\sum_{i=1}^{N}\left(\begin{cases}\frac{h_j(\mathbf{X}_i)f(\mathbf{X}_i)}{b^{-1}h_j(\mathbf{X}_i)f(\mathbf{X}_i)} = b & (\mathbf{X}_i \text{ in pixel j}) \\ \frac{0}{b^{-1}h_{j'}(\mathbf{X}_i)f(\mathbf{X}_i)} = 0 & (\text{otherwise}).\end{cases}\right)\end{aligned} \tag{5}$$

As a given state X_i will contribute to one pixel at a time, we can simplify the denominator. This estimator essentially counts the number of samples that visit the pixel j among all of the N samples, and scales it by the common normalization factor b. Unlike MC where each pixel is estimated by its own sequence of samples, MLT estimates *all of the pixels* using a single sequence of MCMC samples.

Due to this difference, each pixel in MLT receives a varying number of samples which is decided by the target distribution $p(\mathbf{x}) \propto T(\mathbf{x})$ and the MCMC process. In other words, MCMC samples are not stratified at all in the image-space unlike the MC counterparts. Since we rely on a single Markov chain to explore all of the pixels, the convergence of an image in MLT depends on the convergence of this Markov chain process, which unfortunately is known to be quite unpredictable [CJ16].

2.3. Improved Stratification in MLT

Several previous work have attempted to improve the image-space stratification in MLT. Veach and Guibas [Vea97] explained a customized proposal distribution which sequentially proposes samples in all the pixels in a fixed order. While this technique can make sure that *proposals* to cover all of the pixels, the number of samples is *not* stratified due to the accept/reject step and a Markov chain still tends to get stuck at bright pixels.

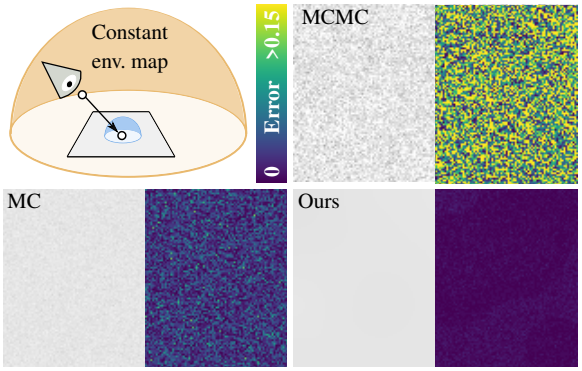

Figure 2: *Comparison of MC, MCMC, and our method for an equal sample count. The scene is a diffuse plane lit by a constant environment map. For this simple scene, it is easy to construct a low-variance estimator, and the result of MC exhibits very little MC noise. MCMC with the same estimator, however, produces a significant amount of noise due to its random-walk nature in the image-space. Our MCMC method does not suffer from such random-walk noise thanks to its perfect stratification in the image-space.*

Energy redistribution path tracing [CTE05] aims to improve stratification by starting a chain from each pixel and running many of them simultaneously. While such a chain tends to explore a neighboring region around each pixel, there is no guarantee of stratification since each chain still freely visits bright pixels more often than dark pixels due to the fundamental properties of MCMC.

Some previous work [Vea97, HH10, GRŠ*16] proposed to modify the target distribution T such that the number of samples in the image-space is more uniform. While such a modified target distribution achieves better stratification in the image-space, the fundamental correlation of samples in MCMC prevents any theoretical guarantee of stratification with a *finite* number of samples. Quasi-Monte Carlo sequences are known to provide guaranteed stratification [CKK18]. While it is tempting to combine quasi-Monte Carlo with MCMC, there is still limited success with no guarantee of stratification in such a combination [OT05].

2.4. Problem Statement

To summarize, there is currently no method that achieves theoretically guaranteed stratification in MCMC with a *finite* number of samples, even if we just focus on stratification in the image-space. The main difficulty of stratification in MCMC is that the distribution of the number of samples is not easily controllable. The distribution of samples is a consequence of running a Markov chain process for many iterations, which is difficult to control compared to independent sampling in MC integration. Since this behavior is fundamental to any MCMC method, stratification in MCMC has been an open problem.

Figure 2 shows a simple motivational example that highlights an issue due to the lack of stratification in MCMC. This scene consists of a diffuse plane lit by a constant environment map. We rendered the scene with MC, MCMC, and our method, all based on the same

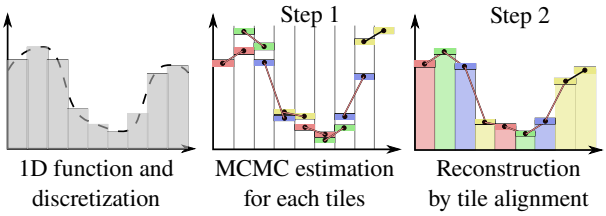

Figure 3: *To estimate a 1D function over a domain, we run several independent MCMC estimation processes on overlapping tiles (Step 1). During the reconstruction, the tiles are aligned such that overlapping regions match (Step 2). The result of this reconstruction is our pixel estimates.*

MIS estimator which combines BSDF sampling and emitter sampling (i.e., sampling the environment map). The comparison here is an equal sample-count. MC integration based on this estimator produces very little noise coming only from the cosine term due to emitter sampling. Based on the primary sample space [KSAC02], we can build an MCMC sampler with the same estimator. Despite using the same low-noise estimator, MCMC suffers from a significant amount of noise compared to MC. It means that the source of noise in this case is not the estimator itself, but the random-walk nature of MCMC in the image-space. Achieving the stratification of MCMC in the image-space should remove such random-walk noise and recover low-noise estimates similar to MC.

Note that stratification in our work follows its definition in statistics; the sampling domain is partitioned into strata, and a sample is generated within each stratum. In our case, what we partition is the image space, and it is partitioned into tiles, which overlap each other. Our method achieves *perfect* stratification in the image space, meaning that each pixel receives exactly the same number of samples for a *finite* number of total samples.

3. Stratified Markov Chain Monte Carlo

We propose a novel stratification framework which enables perfectly-stratified MCMC rendering in the image-space. Our stratified MCMC method completely removes the random-walk noise of MCMC in the image-space and produces more accurate results (see Figure 2). Unlike previous work [Vea97, HH10, GRŠ*16] which modifies the target distribution to improve stratification in the image-space, our work provides perfect stratification in the image-space for any *finite* number of samples per pixel. In other words, we can make sure that all pixels receive an exactly equal number of samples at any point, just like MC integration.

We build upon the fact that MC methods achieve stratification by solving a separate integral for each pixel. We take this approach to achieve the stratification in MCMC by similarly reducing the domain of integration to the size of a pixel. As only the domain of integration is modified, stratification in this manner will work for any arbitrary target function. However, as the arbitrary target function is only defined on the smaller domain, each chain with have its own normalization factor (Equation 3). As we discussed already, this normalization factor is exactly equivalent to the integral we are trying to estimate (Equation. 4). This naive stratification results in MCMC methods providing no benefit over MC methods.

We propose to overcome this limitation by overlapping the domains of each chain and then reconstructing a final result. We rely on the fact that, for a given pixel, the estimates of two independent chains will converge to the same value on the limit. Figure 3 illustrates this idea for a 1D image. Similar to 2D images, the domain of this 1D integration example is discretized into several pixels. For each pixel, we are interested in the expected value of the arbitrary function, which is analogous to light transport simulation. Instead of making the domain of a chain contained within each pixel, we make them overlap by assigning two consecutive pixels for each domain, which call a *tile*. Each tile has its own chain, and the two pixel values in each tile are estimated using its own MCMC process. Following Equation 4, the pixel values of each tile are scaled by its unknown normalization factor (see Figure 3 (1) for illustration). Although those values seem not useful, using the fact that the two overlapping estimates of the same pixel should converge to the same value, we can *align* overlapping tiles. By scaling each tile to minimize the difference between their overlapping regions, we can recover their local normalization factor. The aligned tiles form a global domain which allows us to efficiently estimate only a single normalization factor (Equation 3) for all tiles.

Expanding this idea to light transport, we need to decide how to split the 2D image plane into overlapping tiles and how we compute the MCMC estimates efficiently. As the estimates might be noisy, we will need a robust alignment process for the overlapping tiles.

Overlapping tiles. We choose a cross-shape centered on a pixel with an overlap of one pixel in both the horizontal and vertical directions. This tile shape is the simplest and smallest domain possible that still has uniform overlap with its neighbors in all directions. Uniform overlap between tiles is essential for the reconstruction step as it will spread the error uniformly. In theory, tiles can be of any shape and size, and can share any overlap with their neighbors. We have left such alternate tile configurations for future work and have used cross-shaped tiles for simplicity. Tiles near the image border do not extend beyond the image plane.

3.1. Sampling within a Tile

We index each tile by s, and the Markov chain of this tile will estimate all of the pixels in this tile, independently from other tiles. Following MLT, we define the target function for this tile $T_s(\mathbf{x})$ as

$$T_s(\mathbf{x}) = \sum_{j' \in \mathbb{P}(s)} h_{j'}(\mathbf{x}) f(\mathbf{x}) \tag{6}$$

where $\mathbb{P}(s)$ is a set of pixel indices for the tile s. The measurement contribution function $f(\mathbf{x})$ is usually a vector of chromatic values (e.g., RGB), thus we take the maximum of all the elements in this vector. We, however, keep the notation $f(\mathbf{x})$ for simplicity.

The corresponding normalization factor b_s and the pdf $p_s(\mathbf{x})$ are defined similarly to Equation 3:

$$b_s = \int_P T_s(\mathbf{x}') d\mu(\mathbf{x}') \qquad p_s(\mathbf{x}) = \frac{T_s(\mathbf{x})}{b_s}. \tag{7}$$

Note that $T_s(\mathbf{x}')$ is zero outside the tile, thus the domain of integration P is effectively shrunk to the region of the tile.

Let us consider a sequence of samples $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N$ from the Markov chain which are distributed according to $p_s(\mathbf{x}) \propto T_s(\mathbf{x})$. Equation 5 gives us the estimator $\hat{\mathcal{I}}_s(j)$ for the pixel j in the tile s:

$$\hat{\mathcal{I}}_s(j) = \frac{1}{N}\sum_{i=1}^{N}\left(\begin{cases} \frac{h_j(\mathbf{X}_i)f(\mathbf{X}_i)}{b_s^{-1}h_j(\mathbf{X}_i)f(\mathbf{X}_i)} = b_s & (\mathbf{X}_i \text{ in pixel } j) \\ \frac{0}{b_s^{-1}h_{j'}(\mathbf{X}_i)f(\mathbf{X}_i)} = 0 & (\text{otherwise}). \end{cases}\right) \tag{8}$$

Unlike MLT, this estimator $\hat{\mathcal{I}}_s(j)$ is not useful as we still need to estimate b_s and the number of b_s terms is *equal to the number of pixels*, due to the use of per-pixel Markov chains. We, however, do not need to know b_s at this point, but we can reconstruct them later using the overlapping regions. The Markov chain in each tile thus estimates the *unscaled* intensity $G_s(j) = \hat{\mathcal{I}}_s(j)/b_s$ of the pixel j of the tile s

$$G_s(j) = \frac{1}{N}\sum_{i=1}^{N}\left(\begin{cases} \frac{h_j(\mathbf{X}_i)f(\mathbf{X}_i)}{h_j(\mathbf{X}_i)f(\mathbf{X}_i)} = 1 & (\mathbf{X}_i \text{ in pixel } j) \\ \frac{0}{h_{j'}(\mathbf{X}_i)f(\mathbf{X}_i)} = 0 & (\text{otherwise}). \end{cases}\right) \tag{9}$$

which does not require us to evaluate b_s and can be estimated locally within each tile independently.

Coherent sampling. Inspired by the use of shift mapping in gradient-domain rendering [LKL*13], we found that estimating all of the pixels in the same tile s using a *coherent* set of MCMC chains is effective. In other words, given a sample $\mathbf{X}_i$, we generate four other samples $\mathbf{X}_i^1, \mathbf{X}_i^2, \mathbf{X}_i^3, \mathbf{X}_i^4$ so that a set of samples $\mathbf{X}_i, \mathbf{X}_i^1, \mathbf{X}_i^2, \mathbf{X}_i^3, \mathbf{X}_i^4$ always covers all the five pixels in each tile (see Figure 4). To incorporate this change inside MCMC sampling, we propose to use the maximal value of $T_s(\mathbf{X}_i), T_s(\mathbf{X}_i^1), T_s(\mathbf{X}_i^2), T_s(\mathbf{X}_i^3), T_s(\mathbf{X}_i^4)$. We denote this target function as $T_s(\mathbf{X}_i)$ for simplicity in the next equation. These two changes modify the estimator $G_s(j)$ to

$$G_s(j) = \frac{1}{N}\sum_{i=1}^{N}\left(\begin{cases} \frac{h_j(\mathbf{X}_i)f(\mathbf{X}_i)}{T_s(\mathbf{X}_i)} & (\mathbf{X}_i \text{ in pixel } j) \\ \frac{h_j(\mathbf{X}_i^k)f(\mathbf{X}_i^k)}{T_s(\mathbf{X}_i)} & (\mathbf{X}_i^k \text{ in pixel } j). \end{cases}\right) \tag{10}$$

Note that the pixel j is always sampled even when $\mathbf{X}_i$ is not in this pixel since there is always another coherent sample $\mathbf{X}_i^k$ that is in this pixel. One can consider the sample $\mathbf{X}_i$ as the base path and other four samples $\mathbf{X}_i^1, \mathbf{X}_i^2, \mathbf{X}_i^3, \mathbf{X}_i^4$ as offset paths in the terminology of gradient-domain rendering. While it is conceivable to adopt existing shift mapping techniques [LKL*13] we chose to replay the random numbers used for generating $\mathbf{X}_i$ by offsetting the pixel index for simplicity. The use of more advanced shift mapping algorithms is possible and left for future work.

3.2. Reconstruction of b_s

The above sampling process gives us only $G_s(j)$ for all pixels in all tiles. To obtain an estimator for each pixel, we need to reconstruct b_s. Inspired by gradient-domain rendering [LKL*13], we formulate the process of reconstructing b as an *optimization problem*. We introduce two different approaches: one based on an iterative solver, and the other based on the ratios of pixel values. They are both related to the reconstruction step of gradient-domain rendering from different perspectives.

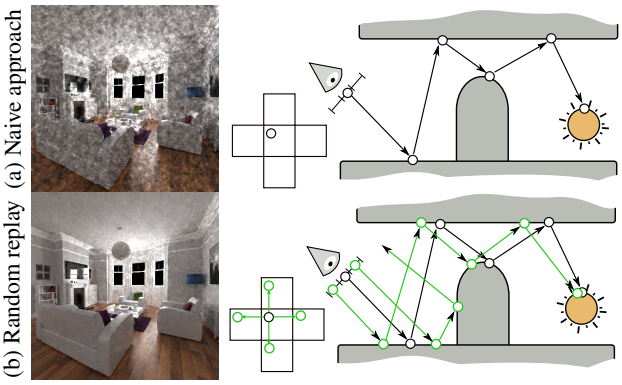

Figure 4: *Performance comparison with different approaches to estimate the unscaled pixel intensities inside tiles: (a) trace only one path (Equation 9) (b) Coherent sampling by replaying the random number generator (Equation 10).*

Since each pixel j has multiple estimates $G_s(j)$ from a set of tiles that contain it, we denote this set of *tile* indices for the pixel j as $\mathbb{N}(j)$. We further define all of the pairs of tiles s and t that contain this pixel j as $\mathbb{M}(j) = \{(s,t), s \in \mathbb{N}(j), t \in \mathbb{N}(j), s \neq t\}$. The reconstruction can be formulated as a minimization of the loss

$$\mathcal{L} = \alpha \left\| \sum_j \hat{\mathcal{I}}(j) - \hat{I}(j) \right\|_2^2 + \sum_j \sum_{(s,t)\in\mathbb{M}(j)} \left\| \hat{\mathcal{I}}_s(j) - \hat{\mathcal{I}}_t(j) \right\|_2^2 \tag{11}$$

where $\hat{\mathcal{I}}_s(j) = b_s G_s(j)$ (Equation 9) and the final estimate $\hat{\mathcal{I}}(j)$ is

$$\hat{\mathcal{I}}(j) = \frac{1}{|\mathbb{N}(j)|}\sum_{s\in\mathbb{N}(j)} G_s(j) b_s. \tag{12}$$

The first term is a regularization term based on the MCMC estimate $\hat{\mathcal{I}}(j)$ and MC estimate $\hat{I}(j)$. Similar to primary sample space MLT [KSAC02], the MC estimate $\hat{I}(j)$ can be obtained by taking samples only from large step mutations from the MCMC process. The constant α is a user parameter that controls the influence of the regularization term. The second term in Equation 11 accounts for the mismatch of overlapping pixel estimates.

Weighted iterative solver. One approach to minimize Equation 11 is to use a Jacobi solver similarly to Rousselle et al. [RJN16] which iteratively updates b_s as

$$b_s^{n+1} = b_s^n + \frac{\alpha w^n(s)(\hat{I}_s - \hat{\mathcal{I}}_s) + \sum_t \sum_{k\in\mathbb{O}(s,t)} w^n(s,t) F_{s,t}^n(k)}{\alpha w^n(s)\bar{G}_s + \sum_t \sum_{k\in\mathbb{O}(s,t)} w^n(s,t) G_t(k)} \tag{13}$$

where $\mathbb{O}(s,t)$ is a set of pairs of overlapping pixels between tiles s and t. The terms $\bar{G}_s$, $\hat{I}_s$ and $\hat{\mathcal{I}}_s$ are the averages of $G_s(j)$, $\hat{I}(j)$, and $\hat{\mathcal{I}}(j)$ within the tile s, respectively. We used b_s^n to denote the estimate of b_s at the nth iteration. The function $F_{s,t}^n(k)$ is defined as

$$F_{s,t}^n(k) = 0.5(G_t(k) b_t^n - G_s(k) b_s^n) \tag{14}$$

where its role is to penalize the difference between the estimates of the pixel k for the overlapping tiles s and t for each iteration n.

Figure 5: *To reconstruct b_s, we can use the solver defined by Equation 20 (left) or Equation 13 (right). While the two approaches generate similar results, the log-domain solver results in some energy loss around bright areas.*

Inspired by the iteratively reweighted least squares solver (IRLS) [CY08], we define the weight for each tile $w^n(s)$ and the weight for two overlapping tiles $w^n(s,t)$ as

$$w^n(s) = \frac{1}{e^n(s) + \beta_1 \beta_2^n} \qquad w^n(s,t) = \min(w^n(s), w^n(t)) \tag{15}$$

where β_1 and β_2 are user-defined parameters to regularize the estimated weights [CY08]. The function $e^n(s)$ models the error introduced by each tile s at the nth iteration:

$$e^n(s) = \alpha \left(\hat{\mathcal{I}}_s^n - \hat{I}_s \right) + \sum_t \sum_{k \in \mathbb{O}(s,t)} \left\| F_{s,t}^n(k) \right\| \tag{16}$$

where $\hat{\mathcal{I}}^n(s)$ is the estimate the nth iteration. Note that we have $\mathcal{L} = \sum_s e^n(s)$. Similarly to IRLS, we chose to update the weight w^n once per 50 iterations to improve the robustness of error estimation. This weighted iterative solver is similar to L1 reconstruction [LKL*13] and weighted reconstruction [RJN16] in gradient-domain rendering.

Ratio-based solver. Another solver we propose is based on the ratios of pixel values. In this solver, the estimator of each pixel is solely decided by the tile c that contains the pixel j in the *center*:

$$\hat{\mathcal{I}}(j) = b_c G_c(j). \tag{17}$$

Let us consider the ratio of two neighboring pixels $I(i)/I(j)$. If all of the b_s are reconstructed correctly, the ratio of $\hat{\mathcal{I}}(i)/\hat{\mathcal{I}}(j)$ should be equal to the correct ratio $I(i)/I(j)$. It is thus reasonable to replace the first term of Equation 11 by the difference of ratios

$$\alpha \left\| \sum_j \hat{\mathcal{I}}(j) - \hat{I}(j) \right\|_2^2 + \sum_j \sum_{i \in \mathbb{N}(j)} \left\| \frac{\hat{\mathcal{I}}(i)}{\hat{\mathcal{I}}(j)} - \frac{I(i)}{I(j)} \right\|_2^2 \tag{18}$$

where we used $\mathbb{N}(j)$ to denote a set of neighboring pixels to the pixel j. In the following discussion, we ignore the regularization term for brevity.

Taking advantage of the fact that all the pixels in the same tile are scaled by a common normalization factor b_c, we can estimate $I(i)/I(j)$ without knowing b_c

$$\frac{I(i)}{I(j)} \approx \frac{\hat{\mathcal{I}}_c(i)}{\hat{\mathcal{I}}_c(j)} = \frac{b_c G_c(i)}{b_c G_c(j)} = \frac{G_c(i)}{G_c(j)} \tag{19}$$

thus we can minimize Equation 18 with respect to b_c for a given estimate of $I(i)/I(j) \approx G_c(i)/G_c(j)$. Unlike Equation 11, however, the loss above is nonlinear to b_s and it is difficult to minimize numerically. We thus consider taking logarithms of each term in order to utilize the relationship of $\log A/B = \log A - \log B$:

$$\begin{aligned} &\sum_j \sum_{i \in \mathbb{N}(j)} \left\| \log \frac{\hat{\mathcal{I}}(i)}{\hat{\mathcal{I}}(j)} - \log \frac{I(i)}{I(j)} \right\|_2^2 \\ &= \sum_j \sum_{i \in \mathbb{N}(j)} \left\| \left(\log \hat{\mathcal{I}}(i) - \log \hat{\mathcal{I}}(j) \right) - \left(\log I(i) - \log I(j) \right) \right\|_2^2 \end{aligned} \tag{20}$$

This loss is easy to minimize by a linear solver for each $\log \hat{\mathcal{I}}(i)$. We then exponentiate the solution to obtain $\hat{\mathcal{I}}(i)$.

Let us compare Equation 20 with the loss for gradient-domain rendering [LKL*13]:

$$\sum_j \sum_{i \in \mathbb{N}(j)} \left\| \left(\hat{\mathcal{I}}(i) - \hat{\mathcal{I}}(j) \right) - \left(I(i) - I(j) \right) \right\|_2^2.$$

We can see that Equation 20 is just *replacing each pixel value by its logarithm*. In gradient-domain rendering, the difference $I(i) - I(j)$ is estimated using shift mapping between the pixel i and the pixel j in order to have a low-variance estimate of $I(i) - I(j)$. In our case, the log difference $\log I(i) - \log I(j)$ is estimated as $\log {}^{I(i)}/_{I(j)} = \log {}^{G_c(i)}/_{G_c(j)}$ using coherent sampling of the pixel i and the pixel j within each tile. In both cases, each pixel $\hat{\mathcal{I}}(i)$ is reconstructed based on the difference by minimizing the loss as defined above.

Discussion. Figure 5 shows a comparison between the reconstructions based on Equation 13 and Equation 20. Overall, the two approaches work reasonably well, but we have found that Equation 20 tends to be numerically unstable due to logarithms and division. After further experiments, we identified two issues that prevent us from using Equation 20 in practice. The first issue is that adding the regularization term based on the *original* pixel values makes Equation 20 nonlinear and challenging to minimize. We tested another regularization term with the logarithms to keep it linear with respect to the logarithms, but we found that such a log-domain regularization term does not perform well in practice. The second issue is that error in the log-domain is nonlinear in the primal domain. Using more sophisticated reconstruction algorithms based on error estimation is thus difficult. For those two issues, we do not use Equation 20 for the rest of the results in this paper. The connection between Equation 20 and gradient-domain rendering, however, remains informative and may lead to future work.

Similar to other reconstruction-based techniques, like L1 Poisson reconstruction [LKL*13, HGP*19], our method is robust against outliers generated from low probability path densities (e.g., fireflies) in the initial normalization factor estimates. This robustness comes at the cost of some bias inside the reconstructed image that will vanish as the number of samples increases.

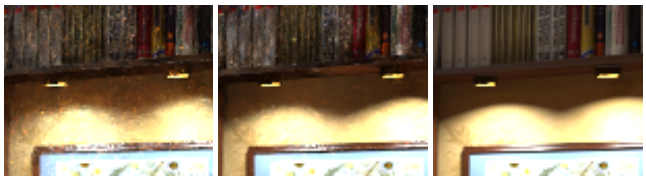

Figure 6: *Comparison of different chain initialization procedures with the same number of initial samples. Left: Using only a limited number of uniform samples to select the initial state cannot ensure that the initial state is selected according to the target function, which results in visible bias. Center: Our method uses on a properly initialized global MCMC chain to perform initialization of many chains, reducing the start-up bias.*

3.3. Chain Initialization

A proper initialization of chains is essential to reduce the start-up bias in MCMC. The start-up bias comes from initial states that start in a low probability area. This start-up bias can result in color shift or low frequency artifact in the final reconstructed image.

To reduce the start-up bias, initial states are often selected using re-sampling from a population of random samples according to the target function [Vea97]. Since our approach uses one chain per pixel, it is difficult to use a large enough population of initial samples per chain, which can lead to artifacts (see Figure 6, left). Discarding samples (burn-in) is also impractical as the number of samples per chain is small compared to a chain exploring the entire image-space as in MLT.

Instead, we propose to use a long chain in MLT to initialize our short chains. This long chain, which we call the "global chain", gets initialized using the re-sampling approach described above. The global chain will use the same target function associated to each tile. The only difference is that the global chain can explore the whole set of tiles. When the global chain visits a tile, its state is distributed according to the target distribution of the visited tile. A chain in each tile is then initialized when the global chain visits the tile for the first time. This initialization performs well and reduces the start-up bias (Figure 6, center).

Note that this procedure does not guarantee that all chains will be initialized. In this case, we will continue to draw a uniform state until we find a state with a non-zero target function value. To reduce the time that the chain is not initialized, we also use a state borrowing procedure during the replica exchange step between neighboring tiles (Section 3.4).

3.4. Local Replica Exchange

MCMC chains are efficient at exploring the state space locally. However, a chain might get trapped inside of a local mode of the target distribution and explore only a small portion of the domain. In our method, the chain of each tile is significantly shorter compared to previous techniques [VG97, KSAC02] and such short chains are prone to this issue. To avoid this problem, we use the replica exchange algorithm [SW86] to exchange the states of two

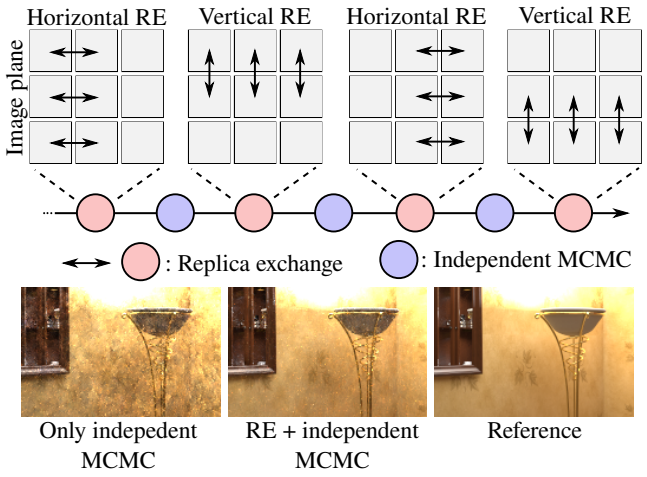

Figure 7: *We perform replica exchange between neighboring tiles by alternating horizontal and vertical pairing. Doing replica exchange improves exploration and leads to better reconstruction compared to only relying on independent MCMC mutations per tile.*

neighboring chains $\mathbf{X}_s$ and $\mathbf{X}_t$ with the probability

$$r(\mathbf{X}_s, \mathbf{X}_t) = \min\left(1, \frac{T_s(\mathbf{X}_t)T_t(\mathbf{X}_s)}{T_s(\mathbf{X}_s)T_t(\mathbf{X}_t)}\right) \tag{21}$$

where T_s and T_t are the target function for the chain $\mathbf{X}_s$ and $\mathbf{X}_t$ respectively. This exchange is done by alternating the exchange between vertical and horizontal neighboring tiles (see Figure 7).

We chose to perform replica exchange between neighboring tiles since they are likely to have similar target functions. This similarity leads to a high exchange probability and thus efficiently improves exploration. This replica exchange operator is done in addition to independent MCMC mutations per tile. Our application of replica exchange is different from the existing work [KKK09, HJ11, SK16] in that we propose to perform replica exchange in the image-space while keeping the target distribution unmodified.

A large step in the primary sample space MLT [KSAC02] can be seen as a particular form of replica exchange [HJ11] where one of the target functions is constant and its state is drawn from the uniform distribution. We thus rely on replica exchange with the uniform distribution to compute the MC estimate for each tile. This MC estimate is used as a regularization term in our reconstruction.

4. Results

We have implemented our stratified MCMC method on top of Mitsuba [Jak13]. All of the results were generated on a dual-socket Intel Core E5-2683 v4 Broadwell CPU at 2.1 GHz with 32 cores and 32 GB of memory. We compared our method against path tracing (MC), primary sample space MLT (PSSMLT) [KSAC02], and energy redistribution path tracing (ERPT) [CTE05] using the same rendering time. The path sampler for all the techniques is path tracing with next-event estimation. The reference images were rendered with bidirectional path tracing. The error metric is the mean

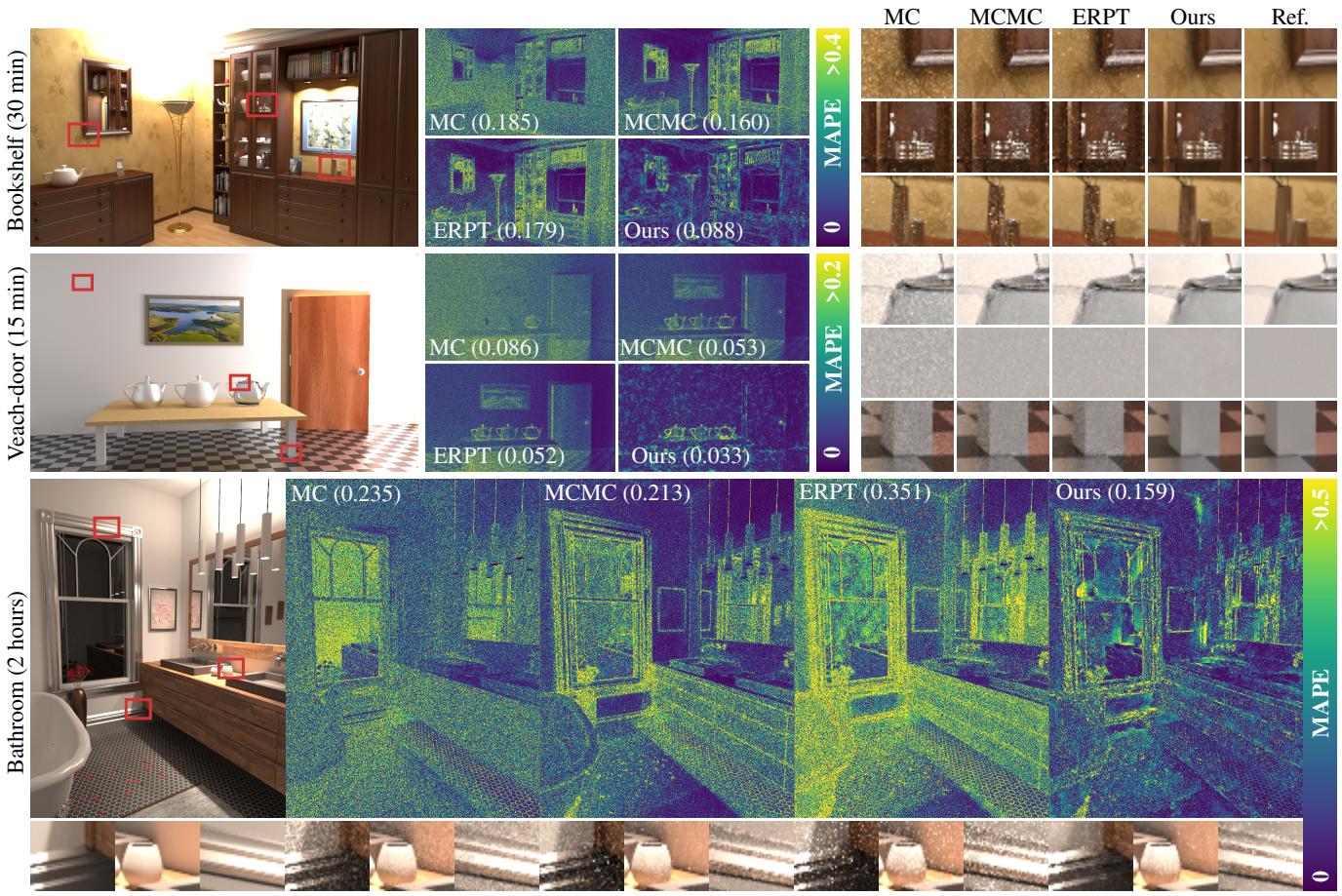

Figure 8: *Equal-time comparisons among path tracing (MC), primary sample space MLT (PSSMLT) [KSAC02], energy redistribution path tracing (ERPT) [CTE05], and our stratified MCMC. All of the scenes we tested feature complex light transport, making those challenging to render efficiently with MC. Among the MCMC methods, both PSSMLT and ERPT tend to produce many erroneously bright pixels. Such pixels are caused by Markov chains being stuck at the same pixels, which is due to the random-walk nature of those two MCMC methods. While being designed to improve stratification in the image-space, ERPT does not work well in our experiments since such bright pixels erroneously consume many iterations of Markov chains, sacrificing the global exploration in the image-space. Our stratified MCMC significantly reduces such erroneously bright pixels thanks to its perfect stratification in the image-space and efficient image-space exploration via replica exchange.*

absolute percentage error $MAPE = (1/n)\sum_j |R_j - I_j|/(R_j + \varepsilon)$, where R_j is the reference and I_j is the rendered pixel luminance value with $\varepsilon = 0.01$.

The small step mutation for all the MCMC methods is the Kelemen's mutation [KSAC02] with the minimum size of $1/1024$ and the maximum size of $1/32$, and the large step probability is 0.3 for both PSSMLT and our method. To conduct a fair comparison with PSSMLT and our method, we have implemented a version of ERPT that uses primary sample space mutations. The chain length for ERPT is set to 100. We use cross-shape tile containing five pixels. For the initialization step of our method, we used global Markov chains to initialize the state of the Markov chain of each tile. This initialization step is performed until either 30% of the pixels have been initialized or the maximum computation budget is reached. We allocated the maximum computational budget of an effective 32 samples per pixel.

Our reconstruction method uses $\beta_1 = 0.05$, $\beta_2 = 0.5$, and $n = 1000$ iterations to reconstruct the normalization factor of each tile (Equation 13). This reconstruction only occurs once at the end of the rendering process and takes few seconds with a CPU implementation. For the regularization term, we use $\alpha = 0.05$ as the unscaled pixel intensity estimates are less noisy than the MC estimates.

Equal-time comparisons. Figure 1 and Figure 8 show the results of equal-time comparisons among different methods. In *Bookshelf* (Figure 1 and Figure 8, top row), the error of our stratified MCMC is the lowest among all of the techniques. Path tracing does not perform well due to the lack of a proper sample strategy for complex paths. All the MCMC methods perform well since MCMC

		MC	MCMC	ERPT	Ours
Bookshelf	L1	0.817	**0.317**	0.444	0.456
	L2	0.563	**0.055**	0.215	0.266
	MAPE	1.859	1.604	1.793	**0.877**
	relMSE	0.403	0.357	0.635	**0.092**
Veach-door	L1	0.511	0.244	0.226	**0.211**
	L2	0.066	0.013	**0.011**	0.014
	MAPE	0.863	0.531	0.528	**0.338**
	relMSE	0.114	0.046	0.047	**0.022**
Bathroom	L1	0.525	**0.353**	0.711	0.429
	L2	0.145	**0.136**	0.226	0.978
	MAPE	2.350	2.135	3.516	**1.598**
	relMSE	0.447	0.531	1.705	**0.377**

Table 1: *The different metric values* $(\times 10^{-1})$ *for all results shown in Figure 8. Our method outperforms previous works on relative metrics consistently. However, MCMC usually achieves the best score in absolute metrics due to its non-uniform sample distribution over the image plane.*

can automatically perform importance sampling according to the path throughput. This fundamental behavior of MCMC, however, leads to different issues that are unique to MCMC. For PSSMLT, the global chain visits bright pixels more often, making dark pixels generally under-sampled and noisy. While ERPT is designed to balance the image-space exploration of bright and dark pixels, it performs worse in this scene since bright pixels spawned too many Markov chains, sacrificing the overall exploration under the same computation time. Our stratified MCMC does not suffer from such issues of MCMC since it guarantees stratification in the image-space while maintaining the benefits of using MCMC over MC. *Veach-door* scene (Figure 8, middle row) showcases difficult visibility due to all of the scene's light coming from a narrow door opening. The path tracing result shows significant noise since the sampling strategy is unable to consider visibility. PSSMLT and ERPT both perform equally well since MCMC can focus computation only on visible paths. Our method performs even better than PSSMLT and ERPT. Note that the number of samples is the same for all pixels for both path tracing and our method, yet our method produces a significantly more accurate result since MCMC in our method still performs importance sampling including visibility.

Bathroom scene (Figure 8, last row) is the most challenging scene since the light is coming from areas where multiple light bounces occur. The images rendered by PSSMLT and path tracing both exhibit significant amount of noise with bright spikes. Similar to *Bookshelf* scene, while it is designed to avoid such issues, switching to ERPT had a negative effect in this scene since bright pixels again consumed too many MCMC iterations. Our method produces significantly fewer spikes as it achieves perfect image-space stratification without significantly sacrificing exploration.

Gradient-domain comparison. In addition to the comparisons shown in the paper, the supplementary document shows results with gradient-domain path tracing [KMA*15] and gradient-domain Metropolis light transport [LKL*13]. To avoid introducing a difference in performance due to a difference in path sampling techniques, we employed primary sample space random-replay [MKA*15, HGP*19] for shift mapping in gradient-domain techniques, and use the primary sample space for mutations in MCMC techniques. The reconstruction for gradient-domain techniques uses the L1 metric. In these comparisons, our MCMC method outperforms all of the gradient-domain techniques.

Error metrics. Table 1 shows the different absolute and relative metrics values for the Figure 8's results. MCMC usually achieves the best score with absolute metrics since brighter pixels received more samples. However, MCMC has more noise in darker image regions, which are not well captured by absolute metrics. Our technique, by distributing the samples evenly on the image space, achieves the best score in relative metric, at the cost of introducing more error in the brightest image regions. Visually inspecting the rendered images, one can observe that our method tends to produce lower-frequency errors than the other methods do, especially in *Bathroom*. It is similar to the fact that gradient-domain methods tend to produce lower-frequency errors than the primal counterparts [HGP*19]. An error metric that can fully capture such a local correlation of pixels is still an open problem even with recent work by Celarek et al. [CJWL19], especially since it involves human perception.

Remaining artifacts Residual errors on the wall and the teapots in the *Veach-door* or glossy surfaces in the *Bathroom* are still visible in our images. These low-frequency artifacts are mainly due to the suboptimal performance of our coherent sampling (Section 3.1). When a path hits a complex shape or goes through many scattering events, the random number replay technique tends to generate incoherent paths. A more sophisticated coherent sampling [KMA*15] can be implemented to ensure better path correlation when sampling the tiles. Further studies on how to incorporate MIS and how to handle failed shifts are left for future work.

5. Limitations and Future work

Path-space MCMC. In our current implementation, we have chosen to use the primary sample space based formulation for simplicity. However, it can be beneficial to use a path-based formulation [VG97] to improve sampling efficiency. In particular, a path-based formulation would allow for several potential improvements including: more coherent mutation of samples, highly coherent sampling within tiles using geometry-aware shift-mapping [LKL*13], and a higher replica exchange acceptance rate via shift-mapping.

MCMC with many chains. Dealing with many chains is challenging due to the length of each chain and how we initialize them. For the same total number of samples, chains in our method have significantly fewer samples than the global chain used in existing MCMC rendering methods. Although replica exchange alleviates this issue, in general, it is difficult to ensure proper exploration with a low sample count. Our stratification of MCMC methods also results in the need to initialize a large number of chains. By using a single, long chain over the entire image-space, we attempt to minimize the start-up bias of as many chains as possible. However, we cannot guarantee that all chains are properly initialized.

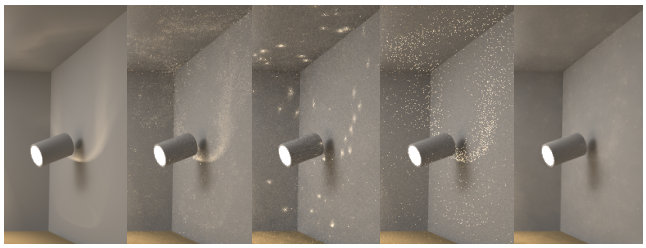

Figure 9: *Some paths are still difficult to sample by path tracing or PSSMLT. Our stratified MCMC shares the same limitation as our chains are unlikely to be properly initialized in this case.*

Artifacts due to difficult paths. Figure 9 shows renderings of a faint caustic. Light paths for such a faint caustic are rarely sampled by the unidirectional path sampler, which results in visible noise in the results of both path tracing and PSSMLT. The chains in our method are also unlikely to be initialized properly and do not explore such paths very well. The reconstruction thus fails to reproduce this caustic entirely. Using a bidirectional path sampler might alleviate the issue, but robust exploration of such rare path events is still an open problem in general. Recent advances in path guiding, where difficult path are selected more often [RHJD18], can potentially be combined with our stratified MCMC.

Future work. We use a cross-shaped tile in this paper, but our formulation is not limited to this particular shape. For example, one can consider having larger tiles to potentially exploit coherency among more pixels, and adaptive tile shapes to numerically maximize the benefit of coherent sampling within a tile. Due to the similarity of our formulation to gradient-domain reconstruction, a more robust reconstruction algorithm such as the one based on control variates [RJN16] might exist for our case as well. We should also emphasize that image-space stratification is only a first step. Higher dimensional stratification, particularly that in the path-space, can provide further improvements.

6. Comparison to Previous Work

Our current implementation uses a path tracer to unidirectionally construct light paths starting from the sensor. Our stratification framework, however, is not limited to this unidirectional path construction and can be applied to other construction techniques such bi-directional path tracing (BDPT) [VG95, LW93]. In BDPT, a family of paths is constructed by connecting subpaths starting from light sources and sensors. Similar to the use of BDPT in PSSMLT [KSAC02], our framework is compatible with BDPT, with the exception of the light tracing technique. The light tracing technique, by construction, cannot be stratified in the image space, and the stratification of light tracing is left for future work. Multiplexed MLT bidirectionally constructs a single path by embedding the choice of which path to generate inside of the chain state [HKD14] and can be combined with our framework.

Besides path construction techniques, researchers have explored improved mutation strategies in the path space such as manifold exploration and variants [JM12, KHD14], geometry-aware perturbation [OHHD18], and derivative-driven mutation [LLR*15]. Recent works [OKH*17, Pan17, BJNJ18] show the connection between the different state spaces used in light transport simulation. All of these improvements do not fundamentally change the behavior of MCMC in terms of image-space exploration. They are thus orthogonal to our contribution and can be incorporated into our framework to improve exploration of the state space.

Metropolis instant radiosity [SIP07b] uses MCMC to generate virtual point lights (VPLs) proportional to their average image-space contribution. This work can be interpreted as achieving perfect stratification over the image space since each VPL contributes to all the pixels, making the contributions of the MCMC chain stratified over the image space. VPLs are, however, prone to singularity artifacts and are inefficient for glossy to glossy light transport. As the results show, our work can leverage the strength of MCMC for such transport while achieving perfect stratification.

For the topic of stratification in MCMC, Šik et al. explained how to improve image-space stratification in their MCMC approach [ŠOHK16] based on UPS/VCM [HPJ12, GKDS12]. Their approach is to use MC sampling for sensor subpaths, and the results demonstrate significant improvement by improving stratification over the image-space. Several previous works [SIP07a, THD17, MDTW19] proposed to generate multiple proposals to improve stratification. Szirmay-Kalos and Szécsi [SKS17] proposed an adaptive mutation strategy which improves stratification in the path space. While the ultimate goal of perfect stratification in both the image space and the path space simultaneously has not been achieved so far, nevertheless, all of these prior works point out the importance of achieving stratification, even if it is only in the image space. Common to all is that none of them ensures perfect stratification in its strict sense, which is what our method achieves in the image space.

Some concurrent work proposes to combine MC and MCMC adaptively to achieve better stratification. Bitterli et al. [BJ19] proposes to selectively run a MCMC process on outlier paths initially generated by an MC technique. Their work is essentially equivalent to the selective path guiding method by Reibold et al. [RHJD18], but path guiding is done by MCMC instead. Grittmann et al. [GGSK19] demonstrate a weighted combination of MC and MCMC images based on variance, and their work also emphasizes the importance of having stratification in the image space. Unlike these concurrent works, we do not combine MC and MCMC, and our framework achieves stratification by using only MCMC.

7. Conclusion

We achieved perfect image-space stratification in MCMC rendering for the first time. The main idea is to use an independent chain per pixel to guarantee stratification by definition. While such an independent MCMC process per pixel leads to an issue of unknown normalization factors, we have identified novel algorithms that can reconstruct them using overlapping image-space tiles per pixel. Our reconstruction algorithms are based on the fact that the same pixel converges to the same value even across different tiles. We showed

how theoretical formulation of this reconstruction process is closely related to gradient-domain reconstruction. This similarity allows us to formulate the reconstruction as a simple optimization problem and motivates the use of coherent sampling for neighboring pixels within each tile. To improve the exploration of many chains, we proposed to perform replica exchange locally between neighboring tiles in the image-space. The numerical results show promising performance compared to MCMC rendering without any stratification. We believe that the theoretical similarity between our technique and gradient-domain rendering opens up many interesting venues for unified treatments between the two.

Acknowledgements We acknowledge Compute Canada for computing resources, and Benedikt Bitterli for veach-door scene [Bit16]. This project is partially funded by Japanese Government (Monbukagakusho - MEXT) Scholarship and JSPS KAKENHI Grant Numbers JP15H05308 and JP18KK0309.

References

[Bit16] BITTERLI B.: Rendering resources, 2016. https://benedikt-bitterli.me/resources/. 11

[BJ19] BITTERLI B., JAROSZ W.: Selectively Metropolised Monte Carlo light transport simulation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) 38*, 6 (2019). doi:10.1145/3355089.3356578. 10

[BJNJ18] BITTERLI B., JAKOB W., NOVÁK J., JAROSZ W.: Reversible jump Metropolis light transport using inverse mappings. *ACM Transactions on Graphics 37*, 1 (Jan. 2018), 1:1–1:12. doi:10/gd52ph. 10

[CJ16] CHRISTENSEN P. H., JAROSZ W.: The path to path-traced movies. *Foundations and Trends® in Computer Graphics and Vision 10*, 2 (Oct. 2016), 103–175. doi:10/gfjwjc. 2, 3

[CJWL19] CELAREK A., JAKOB W., WIMMER M., LEHTINEN J.: Quantifying the error of light transport algorithms. *Computer Graphics Forum 38*, 4 (2019). doi:10.1111/cgf.13775. 9

[CKK18] CHRISTENSEN P., KENSLER A., KILPATRICK C.: Progressive multi-jittered sample sequences. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering) 37*, 4 (July 2018), 21–33. doi:10/gdvj4n. 2, 3

[CTE05] CLINE D., TALBOT J., EGBERT P.: Energy redistribution path tracing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 24*, 3 (July 2005), 1186–1195. doi:10/b3xtrn. 3, 7, 8

[CY08] CHARTRAND R., YIN W.: Iteratively reweighted algorithms for compressive sensing. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (2008), pp. 3869–3872. doi:10.1109/ICASSP.2008.4518498. 6

[GGSK19] GRITTMANN P., GEORGIEV I., SLUSALLEK P., KŘIVÁNEK J.: Variance-aware multiple importance sampling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) 38*, 6 (Dec. 2019). doi:10.1145/3355089.3356515. 10

[GKDS12] GEORGIEV I., KŘIVÁNEK J., DAVIDOVIČ T., SLUSALLEK P.: Light transport simulation with vertex connection and merging. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) 31*, 6 (Nov. 2012), 192:1–192:10. doi:10/gbb6q7. 10

[GRŠ*16] GRUSON A., RIBARDIÈRE M., ŠIK M., VORBA J., COZOT R., BOUATOUCH K., KŘIVÁNEK J.: A spatial target function for Metropolis photon tracing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 36*, 1 (Nov. 2016), 4:1–4:13. doi:10/gfz4kh. 3, 4

[Has70] HASTINGS W. K.: Monte Carlo sampling methods using Markov chains and their applications. *Biometrika 57*, 1 (Apr. 1970), 97–109. doi:10/dkbmcf. 2

[HGP*19] HUA B.-S., GRUSON A., PETITJEAN V., ZWICKER M., NOWROUZEZAHRAI D., EISEMANN E., HACHISUKA T.: A survey on gradient-domain rendering. *Computer Graphics Forum 38*, 2 (2019), 455–472. doi:10.1111/cgf.13652. 6, 9

[HH10] HOBEROCK J., HART J. C.: Arbitrary importance functions for Metropolis light transport. *Computer Graphics Forum 29*, 6 (Sept. 2010), 1993–2003. doi:10/dttgfv. 3, 4

[HJ11] HACHISUKA T., JENSEN H. W.: Robust adaptive photon tracing using photon path visibility. *ACM Transactions on Graphics 30*, 5 (Oct. 2011), 114:1–114:11. doi:10/fpwzq9. 7

[HKD14] HACHISUKA T., KAPLANYAN A. S., DACHSBACHER C.: Multiplexed Metropolis light transport. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 33*, 4 (July 2014), 100:1–100:10. doi:10/f6cswv. 1, 10

[HPJ12] HACHISUKA T., PANTALEONI J., JENSEN H. W.: A path space extension for robust light transport simulation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) 31*, 6 (Jan. 2012), 191:1–191:10. doi:10/gbb6n3. 10

[Jak13] JAKOB W.: Mitsuba renderer, 2013. URL: http://www.mitsuba-renderer.org. 7

[JM12] JAKOB W., MARSCHNER S.: Manifold exploration: A Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 31*, 4 (July 2012), 58:1–58:13. doi:10/gfzq4p. 10

[KHD14] KAPLANYAN A. S., HANIKA J., DACHSBACHER C.: The natural-constraint representation of the path space for efficient light transport simulation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 33*, 4 (July 2014), 102:1–102:13. doi:10/f6cz85. 10

[KKK09] KITAOKA S., KITAMURA Y., KISHINO F.: Replica exchange light transport. *Computer Graphics Forum 28*, 8 (Dec. 2009), 2330–2342. doi:10/brnwj2. 7

[KMA*15] KETTUNEN M., MANZI M., AITTALA M., LEHTINEN J., DURAND F., ZWICKER M.: Gradient-domain path tracing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 34*, 4 (July 2015), 123. doi:10/gfzrhn. 9

[KSAC02] KELEMEN C., SZIRMAY-KALOS L., ANTAL G., CSONKA F.: A simple and robust mutation strategy for the Metropolis light transport algorithm. *Computer Graphics Forum 21*, 3 (Sept. 2002), 531–540. doi:10/bfrsqn. 1, 2, 4, 5, 7, 8, 10

[LKL*13] LEHTINEN J., KARRAS T., LAINE S., AITTALA M., DURAND F., AILA T.: Gradient-domain Metropolis light transport. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 32*, 4 (July 2013), 95:1–95:12. doi:10/gbdghd. 2, 5, 6, 9

[LLR*15] LI T.-M., LEHTINEN J., RAMAMOORTHI R., JAKOB W., DURAND F.: Anisotropic Gaussian mutations for Metropolis light transport through Hessian-Hamiltonian dynamics. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) 34*, 6 (Oct. 2015), 209:1–209:13. doi:10/f7wrcs. 10

[LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bi-directional path tracing. In *Proceedings of the International Conference on Computational Graphics and Visualization Techniques (Compugraphics)* (Alvor, Portugal, Dec. 1993), vol. 93, pp. 145–153. 10

[MDTW19] MERLIN N.-D., DELIO V., TIZIAN Z., WENZEL J.: Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) 38*, 6 (Dec. 2019). doi:10.1145/3355089.3356498. 10

[MKA*15] MANZI M., KETTUNEN M., AITTALA M., LEHTINEN J., DURAND F., ZWICKER M.: Gradient-domain bidirectional path tracing. In *Proceedings of EGSR (Experimental Ideas & Implementations)* (2015). doi:10/gf2hcw. 9

[OHHD18] OTSU H., HANIKA J., HACHISUKA T., DACHSBACHER C.: Geometry-aware Metropolis light transport. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) 37*, 6 (2018), 278:1–278:11. doi:10/gf2r3t. 10

[OKH*17] OTSU H., KAPLANYAN A. S., HANIKA J., DACHSBACHER C., HACHISUKA T.: Fusing state spaces for Markov chain Monte Carlo rendering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 36*, 4 (July 2017), 74:1–74:10. doi:10/gbxjs9. 10

[OT05] OWEN A. B., TRIBBLE S. D.: A quasi-monte carlo metropolis algorithm. *Proceedings of the National Academy of Sciences 102*, 25 (2005), 8844–8849. doi:10.1073/pnas.0409596102. 3

[Pan17] PANTALEONI J.: Charted metropolis light transport. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 36*, 4 (July 2017), 75:1–75:14. doi:10/gfzq78. 10

[RHJD18] REIBOLD F., HANIKA J., JUNG A., DACHSBACHER C.: Selective guided sampling with complete light transport paths. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) 37*, 6 (Dec. 2018), 223:1–223:14. doi:10/gf2g93. 10

[RJN16] ROUSSELLE F., JAROSZ W., NOVÁK J.: Image-space control variates for rendering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) 35*, 6 (Nov. 2016), 169:1–169:12. doi:10/f9cphw. 5, 6, 10

[SIP07a] SEGOVIA B., IEHL J.-C., PEROCHE B.: *Coherent Metropolis Light Transport with Multiple-Try Mutations*. Tech. Rep. RR-LIRIS-2007-015, Universite Lyon, Lyon, France, Apr. 2007. 10

[SIP07b] SEGOVIA B., IEHL J. C., PÉROCHE B.: Metropolis instant radiosity. *Computer Graphics Forum (Proceedings of Eurographics) 26*, 3 (2007), 425–434. doi:10.1111/j.1467-8659.2007.01065.x. 10

[SK16] SIK M., KŘIVÁNEK J.: Improving global exploration of mcmc light transport simulation. In *ACM SIGGRAPH Posters* (2016), ACM. doi:10.1145/2945078.2945128. 7

[SKS17] SZIRMAY-KALOS L., SZÉCSI L.: Improved stratification for metropolis light transport. *Computers & Graphics 68* (2017), 11–20. doi:10.1016/j.cag.2017.07.032. 10

[SNJ*14] SUBR K., NOWROUZEZAHRAI D., JAROSZ W., KAUTZ J., MITCHELL K.: Error analysis of estimators that use combinations of stochastic sampling strategies for direct illumination. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering) 33*, 4 (June 2014), 93–102. doi:10/f6fgw4. 2

[ŠOHK16] ŠIK M., OTSU H., HACHISUKA T., KŘIVÁNEK J.: Robust light transport simulation via Metropolised bidirectional estimators. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) 35*, 6 (Nov. 2016), 245:1–245:12. doi:10/gfz4kj. 10

[SW86] SWENDSEN R. H., WANG J.-S.: Replica monte carlo simulation of spin-glasses. *Physical review letters 57*, 21 (1986), 2607. 7

[THD17] TESSARI L., HANIKA J., DACHSBACHER C.: Local quasi-Monte Carlo exploration. In *Proceedings of EGSR (Experimental Ideas & Implementations)* (June 2017). doi:10.2312/sre.20171196. 10

[Vea97] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. Ph.D. Thesis, Stanford University, Dec. 1997. 2, 3, 4, 7

[VG95] VEACH E., GUIBAS L. J.: Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques (Proceedings of the Eurographics Workshop on Rendering)* (1995), Springer-Verlag, pp. 145–167. doi:10/gfznbh. 10

[VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. In *Annual Conference Series (Proceedings of SIGGRAPH)* (Aug. 1997), vol. 31, ACM Press, pp. 65–76. doi:10/bkjqj4. 1, 2, 7, 9