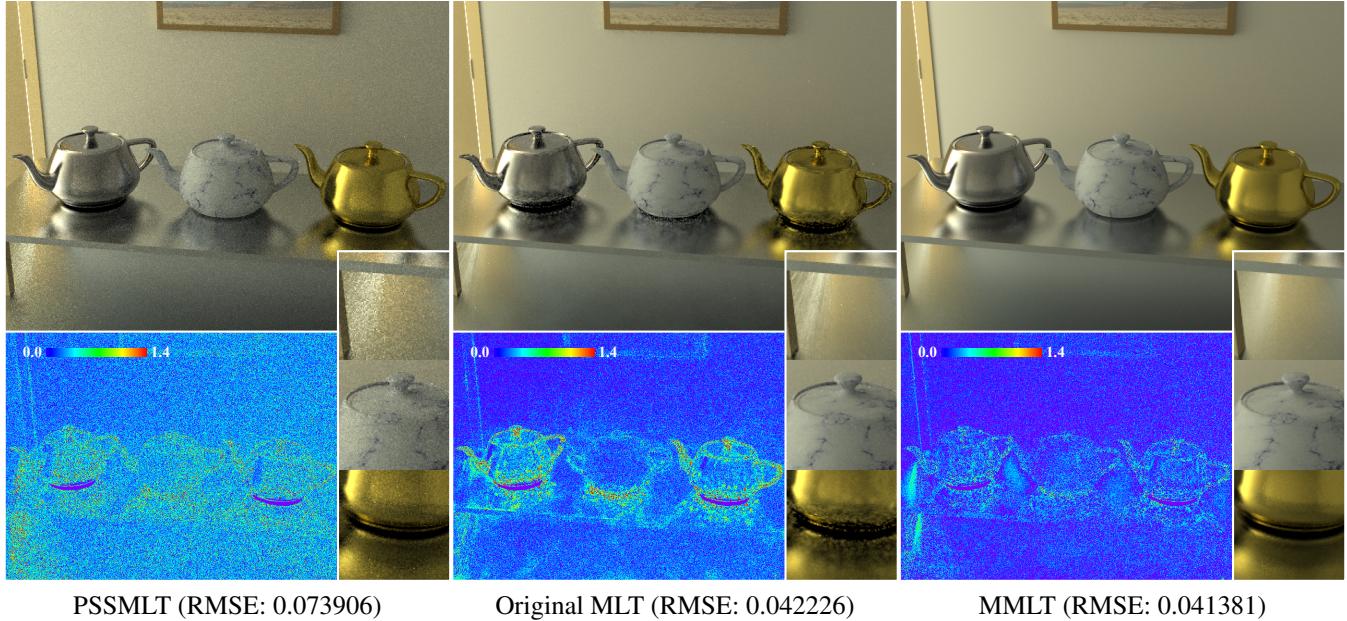


# Multiplexed Metropolis Light Transport

Toshiya Hachisuka<sup>1</sup> Anton S. Kaplanyan<sup>2</sup> Carsten Dachsbaecher<sup>2</sup>  
<sup>1</sup>Aarhus University   <sup>2</sup>Karlsruhe Institute of Technology



**Figure 1:** Door scene with glossy surfaces rendered in 90 min. L2 errors are shown in false color. Primary sample space MLT [Kelemen et al. 2002] (PSSMLT) renders a noisy image as its bidirectional connections cannot efficiently find visible paths in this scene. Original MLT [Veach and Guibas 1997] is efficient at handling such difficult visibility, but suboptimal sampling on glossy surfaces results in distracting artifacts. Our Multiplexed MLT (MMLT), although simple to implement just like PSSMLT, has comparable performance to original MLT on diffuse surfaces and renders glossy surfaces more accurately.

## Abstract

Global illumination algorithms using Markov chain Monte Carlo (MCMC) sampling are well-known for their efficiency in scenes with complex light transport. Samples in such algorithms are generated as a history of Markov chain states so that they are distributed according to the contributions to the image. The whole process is done based only on the information of the path contributions and user-defined transition probabilities from one state to the others. In light transport simulation, however, there is more information that can be used to improve the efficiency of path sampling. A notable example is *multiple importance sampling* (MIS) in bidirectional path tracing, which utilizes the probability densities of constructing a given path with different estimators. While MIS is a powerful ordinary Monte Carlo method, how to incorporate such additional information into MCMC sampling has been an open problem. We introduce a novel MCMC sampling framework, *primary space serial tempering*, which fuses the ideas of MCMC sampling and MIS for the first time. The key idea is to explore not only the sample space using a Markov chain, but also different estimators to generate samples by utilizing the information already available for MIS. Based on this framework, we also develop a novel rendering algorithm, *multiplexed Metropolis light transport*, which automatically and adaptively constructs paths with appropriate techniques as predicted by MIS. The final algorithm is very easy to implement, yet in many cases shows comparable (or even better) performance than significantly more complex MCMC rendering algorithms.

**CR Categories:** I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing

**Keywords:** global illumination, Markov chain Monte Carlo, multiple importance sampling

**Links:** DOI PDF

## 1 Introduction

Rendering photorealistic images in computer graphics is almost exclusively done by using Monte Carlo methods nowadays. They all share the concept of stochastically constructing paths that connect the sensor to a light source and computing the energy reaching the sensor. This process can be done in many ways: sampling only from the sensor or the light sources (path tracing [Kajiya 1986] or light tracing [Arvo 1986]), sampling from both sides with deterministic connections (bidirectional path tracing [Lafortune and Willems 1993; Veach and Guibas 1995]), or density estimation of path vertices (photon mapping [Jensen 1996]).

---

### ACM Reference Format

Hachisuka, T., Kaplanyan, A., Dachsbaecher, C. 2014. Multiplexed Metropolis Light Transport. ACM Trans. Graph. 33, 4, Article 100 (July 2014), 10 pages. DOI = 10.1145/2601097.2601138.  
<http://doi.acm.org/10.1145/2601097.2601138>

### Copyright Notice

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

2014 Copyright held by the Owner/Author. Publication rights licensed to ACM.  
0730-0301/14/07-ART100 \$15.00.  
DOI: <http://dx.doi.org/10.1145/2601097.2601138>

While direct applications of general Monte Carlo methods are feasible, some rendering methods effectively utilize more domain-specific knowledge of light transport simulation than the others. One notable example is multiple importance sampling (MIS) [Veach and Guibas 1995]. MIS uses the extra knowledge of multiple different approaches to construct a given path with different probability densities. This information is then used to define a weighted sum of Monte Carlo estimators that results in less variance than using each estimator alone. MIS has been proven useful in many existing rendering algorithms, such as the recent combinations of density estimation and Monte Carlo integration [Georgiev et al. 2012; Hachisuka et al. 2012], as well as its application to many-lights algorithms [Walter et al. 2012].

On the other hand, Markov Chain Monte Carlo (MCMC) rendering methods (as first introduced to rendering by Veach and Guibas [1997]) take a more direct approach. This class of methods applies MCMC sampling to locally explore the space of transport paths. They obtain paths from a Markov chain, where a new path is generated based *only* on the information about the current path, without any knowledge of probability densities. While the dependency only on the current path is the reason why MCMC methods are so general, they can still benefit from incorporating more domain-specific knowledge in light transport simulation as some ordinary Monte Carlo methods do.

We present a novel MCMC sampling framework, *primary space serial tempering*, which enables us to combine the generality of MCMC sampling with the additional knowledge of multiple different techniques as in MIS. The main idea is to let a Markov chain explore not only the space of transport paths, but also different sampling techniques for each path. To the best of our knowledge, it is the first general MCMC framework that enables us to achieve such a combination. This framework leads to a new MCMC light transport simulation method, *Multiplexed Metropolis Light Transport* (MMLT), which explores the sampling space *and* adaptively changes the path sampling technique using MCMC sampling.

Despite its improved adaptivity, our new algorithm is easy to implement. We demonstrate the efficiency of our algorithm on scenes with complex light transport due to combinations of difficult visibility and highly glossy transport. We show that the performance of our algorithm is comparable to, or even better than, significantly more complex MCMC rendering algorithms (e.g., Fig. 1).

In summary, the contributions of this paper are:

- A theory to bridge multiple importance sampling and MCMC sampling (Sect. 2 and Sect. 3).
- *Primary space serial tempering*, which allows a Markov chain to explore different techniques for sampling a given distribution with additional information available from MIS (Sect. 3.2).
- *Multiplexed Metropolis Light Transport*, a simple-to-implement rendering method built upon our general framework, which is robust to many scene configurations (Sect. 4).

## 2 Connection between MCMC and MIS

In order to establish the connection between MCMC and MIS, we recapitulate important concepts related to our work using a consistent formulation (see Table 1 for notations). Using this formulation, we show how to combine MCMC and MIS in Sect. 3.2. Readers familiar with MCMC and MIS might skip to Sect. 2.6 where we state the problem we solve. We discuss our method and its relation to highly related previous work in Sect. 7.

Symbol	Description
$\bar{x}, \bar{y}$	path defined as a vector of vertices
$f(\bar{x})$	path throughput
$C(\bar{x})$	path contribution
$p(\bar{x}), P(\bar{x})$	probability density function, cumulative distribution
$w(\bar{x})$	multiple importance sampling weighting function
$\bar{u}, \bar{v}$	path defined as a vector of random numbers
$\hat{C}_i(\bar{u}), \hat{p}(\bar{u}), \hat{w}(\bar{u})$	functions in primary space
$f^*(\bar{x}), C^*(\bar{x}), p^*(\bar{x})$	unnormalized target distribution of $\bar{x}$
$b$	normalization factor
$q(\mathbf{x} \rightarrow \mathbf{y})$	transition probability of $\mathbf{y}$ given $\mathbf{x}$
$\min(a(\mathbf{x} \rightarrow \mathbf{y}), 1)$	acceptance probability of the proposal $\mathbf{y}$ given $\mathbf{x}$
$\min(r(t \rightarrow t'), 1)$	acceptance probability of technique $t'$ given $t$
$\Omega, \mathcal{U}$	sampling domain
$N, M$	the number of samples, the number of techniques
$k, t, s$	path length, the numbers of eye/light path vertices

**Table 1:** Notation used in this paper.

### 2.1 (Markov Chain) Monte Carlo Integration

Monte Carlo integration [Metropolis and Ulam 1949] estimates the value of a definite (possibly vector valued) integral with  $N$  samples  $\mathbf{x}_i \in \Omega$  drawn from a probability density function (PDF)  $p(\mathbf{x})$  as

$$\int_{\Omega} g(\mathbf{x}) d\mathbf{x} \approx \frac{1}{N} \sum_{i=1}^N \frac{g(\mathbf{x}_i)}{p(\mathbf{x}_i)}. \quad (1)$$

$p(\mathbf{x})$  should ideally be as proportional to  $g(\mathbf{x})$  as possible, which is the variance reduction technique known as *importance sampling*.

There are many approaches to generate samples from a given PDF [Brooks et al. 2011]. MCMC sampling [Metropolis et al. 1953] is one powerful approach as it can generate samples according to any positive scalar function  $p^*(\mathbf{x}) \propto p(\mathbf{x})$  without knowing the normalization constant  $b = \int_{\Omega} p^*(\mathbf{x}) d\mathbf{x}$  of the PDF. Using samples from MCMC, we can estimate the above integral as:

$$\int_{\Omega} g(\mathbf{x}) d\mathbf{x} \approx \frac{b}{N} \sum_{i=1}^N \frac{g(\mathbf{x}_i)}{p^*(\mathbf{x}_i)}. \quad (2)$$

MCMC sampling itself is not a particularly useful integration method (especially when  $p^*(\mathbf{x}) \propto g(\mathbf{x})$ ) since we still need to know  $b$  to estimate the above integral. However, when we need to estimate many different integrals  $\int_{\Omega} g_0(\mathbf{x}) d\mathbf{x}, \int_{\Omega} g_1(\mathbf{x}) d\mathbf{x}, \dots$  with a *common*  $p^*(\mathbf{x})$  (as in rendering, see Sect. 2.3), MCMC sampling allows us to estimate *all* integrals up to the *common* scaling factor  $b$ . The scaling factor is then estimated separately, for example, by ordinary Monte Carlo integration.

MCMC sampling generates samples as a history of states of a Markov chain. Given  $\mathbf{x}_i$  as the current state of the Markov chain, first a proposal  $\mathbf{y}$  is generated based on a conditional probability density  $q(\mathbf{x}_i \rightarrow \mathbf{y})$  (*proposal density*, often called *mutation* in rendering), and then we set  $\mathbf{x}_{i+1} = \mathbf{y}$  with the probability  $\min(a, 1)$ ;

$$a(\mathbf{x}_i \rightarrow \mathbf{y}) = \frac{p^*(\mathbf{y}) q(\mathbf{y} \rightarrow \mathbf{x}_i)}{p^*(\mathbf{x}_i) q(\mathbf{x}_i \rightarrow \mathbf{y})}. \quad (3)$$

Otherwise, the same state  $\mathbf{x}_{i+1} = \mathbf{x}_i$  is repeated; this is the Metropolis-Hastings (MH) algorithm [Hastings 1970].

### 2.2 Path Integral

Light transport simulation algorithms in rendering essentially compute an estimate of the path integral [Veach 1998]:

$$I_j = \int_{\Omega(\mathcal{M})} h_j(\bar{x}) f(\bar{x}) d\mu(\bar{x}) = \sum_{k=1}^{\infty} \int_{\Omega^k(\mathcal{M})} h_j(\bar{x}) f(\bar{x}) d\mu(\bar{x}), \quad (4)$$

where  $I_j$  is the intensity of the  $j$ -th pixel,  $\Omega^k(\mathcal{M})$  is the space of all possible light paths of length  $k$ , and  $\Omega(\mathcal{M}) = \bigcup_{k=1}^{\infty} \Omega^k(\mathcal{M})$  is the space of all possible paths;  $\bar{x} \in \Omega^k(\mathcal{M})$  is a complete path from a light to the camera in this path space. A path is represented as a vector of points on the scene manifold  $\mathcal{M}$ , i.e.  $\bar{x} = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k)$ .

The path throughput  $f(\bar{x})$  includes a product of reflection operators at all points of a path  $\bar{x}$ , and  $h_j(\bar{x})$  is a pixel filtering function which is non-zero only for the support of the  $j$ -th pixel. Note that we explicitly separated paths of different lengths in Eq. 4 for brevity in later sections. In the following, we show equations only on a specific path length  $k$  whenever we define an estimator of  $I_j$ .

Path tracing [Kajiya 1986] is a direct application of ordinary MC integration (Eq. 1) to the path integral. Each  $I_j$  is independently estimated by sampling paths  $\bar{x}_i \sim p_j(\bar{x}_i)$ :

$$I_j \approx \frac{1}{N} \sum_{i=1}^N \frac{h_j(\bar{x}_i)f(\bar{x}_i)}{p_j(\bar{x}_i)} = \frac{1}{N} \sum_{i=1}^N h_j(\bar{x}_i)C_j(\bar{x}_i), \quad (5)$$

where  $C_j(\bar{x}) = f(\bar{x})/p_j(\bar{x})$  is the path contribution. Sampling a path through each pixel normally leads to a unique probability density  $p_j(\bar{x})$  as a product of PDFs for BSDF, light, and lens sampling.

### 2.3 Original Metropolis Light Transport

MLT [Veach and Guibas 1997] applies Eq. 2 to the path integral as

$$I_j \approx \frac{b}{N} \sum_{i=1}^N \frac{h_j(\bar{x}_i)f(\bar{x}_i)}{f^*(\bar{x}_i)}, \quad (6)$$

where  $f^*(\bar{x})$  is the scalar luminosity of the path throughput  $f(\bar{x})$ , which is used as the target distribution  $p^*(\bar{x})$  in Eq. 2.

Unlike in path tracing, path samples  $\bar{x}_i$  are shared among pixels in MLT: samples are projected to the image plane and contribute to pixels with non-zero  $h_j(\bar{x}_i)$ . By this process, they can be used for estimating the integrals of *all* pixels at the same time. The common scaling factor  $b = \int_{\Omega} f^*(\bar{x})d\mu(\bar{x})$  (related to the brightness of the image) is estimated by separate MC integration such as path tracing. MLT updates the current path  $\bar{x}_i$  using MCMC (Eq. 3)

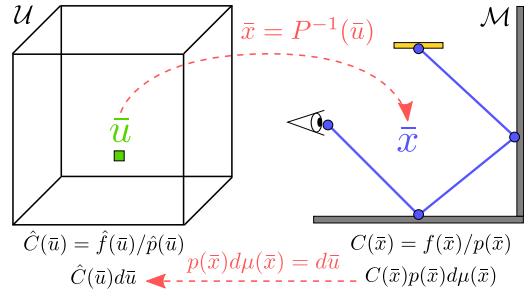
$$a(\bar{x}_i \rightarrow \bar{y}) = \frac{f^*(\bar{y})q(\bar{y} \rightarrow \bar{x}_i)}{f^*(\bar{x}_i)q(\bar{x}_i \rightarrow \bar{y})}. \quad (7)$$

Since well-designed mutation strategies in MLT locally explore the sample space of  $f^*(\bar{x})$  where they are large, MLT is effective at avoiding sampling paths where  $f(\bar{x})$  is zero (thus  $f^*(\bar{x})$  is zero) due to visibility. This consideration of the complete path throughput is a key advantage over ordinary MC methods, where it would be difficult to define such a PDF.

The implementation of original MLT, however, has been commonly considered very challenging due to the intricate MCMC processes such as the mutations and computation of the transition probabilities directly on a path  $\bar{x}$  (Eq. 7). For example, randomly perturbing all the vertex locations of a given path almost always results in an invalid path, since perturbed vertices will likely be off the scene surfaces  $\mathcal{M}$ . Each mutation strategy therefore has to be carefully designed such that it generates valid paths, while making a reasonable change to the current path [Veach and Guibas 1997].

### 2.4 Primary Sample Space Metropolis Light Transport

Primary sample space MLT (PSSMLT) [Kelemen et al. 2002] significantly simplifies the MCMC process compared to the original



**Figure 2:** Primary sample space MLT.  $\bar{u}$  is a vector of random numbers mapped to a path  $\bar{x}$  according to the inverse cumulative distribution function  $P^{-1}(\bar{u})$ . The path contribution in each space is  $C(\bar{x})$  and  $\hat{C}(\bar{u})$ . The differential contribution (used for integration)  $C(\bar{x})p(\bar{x})d\mu(\bar{x})$  is mapped into  $\hat{C}(\bar{u})d\bar{u}$  in the primary space.

MLT method. It utilizes the fact that each path  $\bar{x}$  is uniquely defined by a vector of random numbers  $\bar{u} \in \mathcal{U} = [0, 1]^{O(k)}$  ( $O(k)$  random numbers are normally needed to define a path of length  $k$ ). The mapping from  $\bar{u}$  to  $\bar{x}$  is defined by the (inverse) cumulative distribution function (CDF) as  $\bar{x} = P^{-1}(\bar{u})$  (Fig. 2). This mapping is already used in MC sampling of paths in Eq. 5 to trace a path  $\bar{x}$  given uniform random numbers  $\bar{u}$ .

In order to utilize this mapping in MCMC sampling, Kelemen et al. considered the following decomposition with a given  $p(\bar{x})$ :

$$f(\bar{x}) = \frac{f(\bar{x})}{p(\bar{x})}p(\bar{x}) = C(\bar{x})p(\bar{x}). \quad (8)$$

For the brevity of notation, we define the path throughput and the PDF in primary space as  $\hat{C}(\bar{u}) = C(P^{-1}(\bar{u})) = C(\bar{x})$  and  $\hat{p}(\bar{u}) = p(P^{-1}(\bar{u})) = p(\bar{x})$ . Using this notation, we can observe the following relationship [Kelemen et al. 2002] (see Fig. 2):

$$\begin{aligned} C(\bar{x})p(\bar{x})d\mu(\bar{x}) &= \hat{C}(\bar{u})\hat{p}(\bar{u}) |\frac{d\mu(\bar{x})}{d\bar{u}}| d\bar{u} \\ &= \hat{C}(\bar{u})\hat{p}(\bar{u}) |\frac{dP^{-1}(\bar{u})}{d\bar{u}}| d\bar{u} \\ &= \hat{C}(\bar{u})\hat{p}(\bar{u}) 1/\hat{p}(\bar{u}) d\bar{u} = \hat{C}(\bar{u})d\bar{u}, \end{aligned} \quad (9)$$

which leads to a simplified path integral over the unit hypercube  $\mathcal{U}$  and the respective MCMC estimator:

$$\begin{aligned} I_j &= \int_{\Omega(\mathcal{M})} h_j(\bar{x})C(\bar{x})p(\bar{x})d\mu(\bar{x}) \\ &= \int_{\Omega(\mathcal{U})} \hat{h}_j(\bar{u})\hat{C}(\bar{u})d\bar{u} \approx \frac{b}{N} \sum_{i=1}^N \frac{\hat{h}_j(\bar{u}_i)\hat{C}(\bar{u}_i)}{\hat{C}^*(\bar{u}_i)}. \end{aligned} \quad (10)$$

This algorithm operates on a vector of random numbers  $\bar{u} \in \mathcal{U}$  which is significantly easier to manipulate than a vector of vertices on the scene manifold  $\bar{x} \in \mathcal{M}$ . For example, any random perturbation of  $\bar{u}$  results in  $\bar{x} \in \mathcal{M}$  in a closed scene (recall that almost all random perturbations of  $\bar{x}$  itself will be invalid). Furthermore, it is easy to use a symmetric mutation in the primary space which avoids the computation of the transition probabilities altogether.

The MCMC update process in Eq. 3, now in the primary space with the symmetric mutation  $\bar{v} \sim q(\bar{u}_i \rightarrow \bar{v}) = q(\bar{v} \rightarrow \bar{u}_i)$ , becomes

$$a(\bar{u}_i \rightarrow \bar{v}) = \frac{\hat{C}^*(\bar{v})}{\hat{C}^*(\bar{u}_i)}. \quad (11)$$

In comparison to Eq. 7,  $\hat{C}^*(\bar{x})$  can be significantly flatter and thus easier to explore by MCMC sampling than  $f^*(\bar{x})$ , if we have a

PDF  $p(\bar{x})$  roughly proportional to  $f^*(\bar{x})$ . In this case, we have  $a(\bar{u}_i \rightarrow \bar{v}) = \hat{C}^*(\bar{v})/\hat{C}^*(\bar{u}_i) \approx 1$ , which avoids repeating the same states while allowing large changes in MCMC sampling.

Unfortunately, it is difficult to find such a PDF. In fact, importance sampling in ordinary MC (Eq. 1) also faces the same difficulty, which is inherited here due to the decomposition using  $p(\bar{x})$  in Eq. 8. In addition, since the algorithm indirectly mutates a path by changing its associated vector of random numbers with a fixed mapping defined by  $p(\bar{x})$ , the flexibility of its mutation strategies is limited compared to original MLT.

## 2.5 Multiple Importance Sampling

While it is difficult to define a single  $p(\bar{x})$  that is roughly proportional to  $f^*(\bar{x})$  everywhere, we often know *multiple* different PDFs  $p_t(\bar{x})$  where each PDF *locally* approximates  $f^*(\bar{x})$  better than the others. Multiple importance sampling (MIS) [Veach and Guibas 1995] uses such extra information to generalize the decomposition in Eq. 8 to a weighted sum:

$$f(\bar{x}) = \sum_{t=1}^M w_t(\bar{x}) \frac{f(\bar{x})}{p_t(\bar{x})} p_t(\bar{x}) = \sum_{t=1}^M w_t(\bar{x}) C_t(\bar{x}) p_t(\bar{x}), \quad (12)$$

where  $M$  is the number of PDFs and  $w_t(\bar{x})$  is a weighting function with  $\sum_{t=1}^M w_t(\bar{x}) = 1$  for any given  $\bar{x}$ .

Following the original terminology by Veach [1998, p.299], we define an  $(s, t)$  technique (or  $t$ -th technique given fixed  $k$ ) as a bidirectional path sampling procedure where we trace  $t$  eye-subpath and  $s$  light-subpath vertices to generate a path of length  $k = s + t - 1$ . Each  $p_t(\bar{x})$  then corresponds to a PDF of sampling such a pair of subpaths. In this paper,  $p_t(\bar{x})$  is specifically defined as a product of subpath sampling probability densities with  $t$  eye subpath vertices and  $s = (k+1) - t$  light subpaths vertices for a given path length  $k$ . MIS (and our method) however is not limited to this particular definition of PDFs.

MIS uses ordinary MC integration (Eq. 1) to estimate  $I_j$  by taking  $N_t$  samples  $\bar{x}_{t,i}$  ( $t = 1 \dots M$ ,  $i = 1 \dots N_t$ ) distributed according to  $p_t(\bar{x})$ :

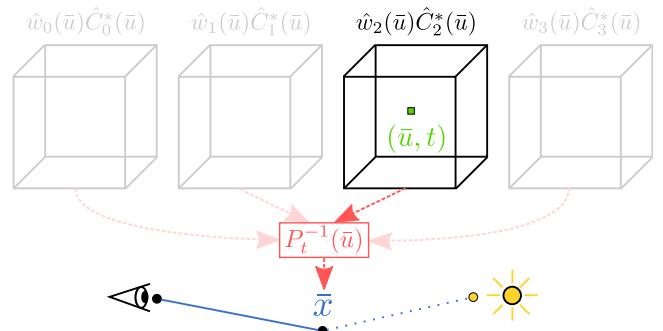
$$I_j \approx \sum_{t=1}^M \frac{1}{N_t} \sum_{i=1}^{N_t} h_j(\bar{x}_{t,i}) w_t(\bar{x}_{t,i}) C_t(\bar{x}_{t,i}), \quad (13)$$

which is a generalization of Eq. 5.

## 2.6 Analysis and Problem Statement

MIS is a powerful technique as each PDF  $p_t(\bar{x})$  can account for a different approach to sample the same path. In contrast, MCMC sampling ignores this extra information and directly tries to distribute paths according to  $f^*(\bar{x})$ . MIS however can be computationally wasteful since samples are weighted *after being generated*, where the weight  $w_t(\bar{x})$  can be extremely small (or even zero) while their probability densities are not. Moreover, the PDFs  $p_t(\bar{x})$  typically do not include visibility as opposed to the target distribution  $f^*(\bar{x})$  in MCMC sampling.

While the relationship between MCMC sampling and MIS has been considered unclear, from Eq. 8 and Eq. 12, we can see that primary sample space MLT is an MCMC sampling approach operating on a special case of MIS, where we use only one technique  $p(\bar{x})$  and  $w(\bar{x}) = 1$  (original importance sampling). As such, it is important to note that the use of bidirectional path tracing in primary sample space MLT *does not* utilize all available information as in Eq. 12.



**Figure 3:** Overview of multiplexed MLT for paths of length two. MMLT utilizes different mappings to the path space by extending the Markov chain state  $(\bar{u}, t)$ . The index  $t$  defines the corresponding bidirectional sampling technique (the illustration shows an example of  $t = 2$ ). MMLT can change the sampling technique by changing  $t$  via MCMC. The target distribution of each primary space is weighted according to  $\hat{w}_t(\bar{u})$  defined via MIS.

Specifically, while the evaluation of  $C(\bar{x})$  might be done by bidirectional path tracing with MIS, there is always a *unique* mapping  $\bar{x} = P^{-1}(\bar{u})$  (Eq. 9). This means that the fact that we have  $M$  *different* mappings  $\bar{x}_t = P_t^{-1}(\bar{u})$  is ignored. Ignoring such information can be wasteful since each  $t$ -th technique can result in a different  $p_t(\bar{x})$  and potentially better approximates  $f^*(\bar{x})$ .

## 3 Primary Space Serial Tempering

We explain serial tempering [Marinari and Parisi 1992; Geyer and Thompson 1995] as the basis in order to explain how we extend it to our novel framework, *primary space serial tempering*. Our framework enables us to use MCMC sampling with the available information for MIS. To be specific, we can sample directly from the distribution defined by Eq. 12 using *MCMC* sampling, while fully utilizing all the information of  $p_t(\bar{x})$  and  $P_t^{-1}(\bar{u})$ .

### 3.1 Serial Tempering

Efficient MCMC sampling from a sum of *parametrized* distributions has been actively investigated in computational statistics [Brooks et al. 2011]. In this problem setting, we generate samples from  $\sum_{t=1}^M f(\mathbf{x}, T_t)$  where  $T_t$  is a parameter of the target distribution  $f(\mathbf{x}, T)$  ( $T$  originally denotes the temperature of a physical process, thus the name “tempering”).

Among many approaches, serial tempering [Marinari and Parisi 1992; Geyer and Thompson 1995] allows us to use a single Markov chain to explore the extended state space  $(\mathbf{x}, T)$ , where  $t$  is the index to the discretized parameter  $T_t$ . In addition to the regular MCMC update of  $\mathbf{x}$  while keeping  $T_t$  (i.e., using Eq. 3 with fixed  $T_t$ ), the extra parameter  $T_t$  is updated to  $T_{t'}$  with the probability  $\min(1, r(T_t \rightarrow T_{t'}))$  where

$$r(T_t \rightarrow T_{t'}) = \frac{f^*(\mathbf{x}, T_{t'}) q(T_{t'} \rightarrow T_t)}{f^*(\mathbf{x}, T_t) q(T_t \rightarrow T_{t'})}. \quad (14)$$

Here,  $q(T_t \rightarrow T_{t'})$  is a transition probability density from  $T_t$  to  $T_{t'}$ . By introducing this update of the parameter, serial tempering makes the chain *automatically* visit parameters  $T$  that result in large  $f(\mathbf{x}, T)$  and generates samples according to  $\sum_{t=1}^M f(\mathbf{x}, T_t)$ .

**Problems of the Direct Application to Original MLT** It might be tempting to apply serial tempering directly to the problem of sampling  $\sum_{t=1}^M w_t(\bar{x})C_t(\bar{x})p_t(\bar{x})$  by defining

$$f^*(\bar{x}, T_t) = w_t(\bar{x})C_t^*(\bar{x})p_t(\bar{x}) = w_t(\bar{x})f^*(\bar{x}). \quad (15)$$

In the context of light transport simulation, this is equivalent to using a path  $\bar{x}$  as the state of a Markov chain as in the original MLT method, except that we now use  $w_t(\bar{x})f^*(\bar{x})$  as the target distribution specified by the extra Markov chain state  $t$ .

There are three major issues with this approach. Firstly, it still ignores the additional knowledge of  $p_t(\bar{x})$  and  $P_t^{-1}(\bar{u})$ , since  $p_t(\bar{x})$  cancels out. Secondly, this approach would still be challenging to implement due to the complexity of the MCMC update process akin to the original MLT method. Lastly, and most importantly, it is not clear how  $w_t(\bar{x})$  (inherited from MIS) should be defined and evaluated in this application, since a path is not generated by the  $t$ -th technique, but by a generic MCMC process. Because of these reasons, this direct application of serial tempering to original MLT is not feasible.

### 3.2 Primary Space Serial Tempering

Our solution is to define each target distribution as  $f^*(\bar{x}, T_t) = w_t(\bar{x})C_t^*(\bar{x})p_t(\bar{x})$  in the primary space according to Eq. 9. Each target distribution is thus “warped” according to the corresponding CDF  $P_t(\bar{x})$ . This results in the target distribution  $\hat{f}^*(\bar{u}, T_t) = \hat{w}_t(\bar{u})\hat{C}_t^*(\bar{u})$  for each technique  $t$  since

$$w_t(\bar{x})C_t^*(\bar{x})p_t(\bar{x})d\mu(\bar{x}) = \hat{w}_t(\bar{u})\hat{C}_t^*(\bar{u})d\bar{u}, \quad (16)$$

where  $\hat{w}_t(\bar{u})$  is defined analogously to  $\hat{C}_t^*(\bar{u})$ . Each  $t$ -th primary space has its own mapping defined as  $P_t^{-1}(\bar{u})$ , thus we can utilize the additional knowledge of  $p_t(\bar{x})$  and  $P_t^{-1}(\bar{u})$ .

The proposed state  $\bar{v}$ , given the current state  $\bar{u}$  and the technique  $t$ , is accepted with the probability  $\min(1, a(\bar{u} \rightarrow \bar{v}))$ ;

$$a(\bar{u} \rightarrow \bar{v}) = \frac{\hat{w}_t(\bar{v})\hat{C}_t^*(\bar{v})q(\bar{v} \rightarrow \bar{u})}{\hat{w}_t(\bar{u})\hat{C}_t^*(\bar{u})q(\bar{u} \rightarrow \bar{v})}. \quad (17)$$

Similar to the argument in primary sample space MLT,  $\hat{w}_t(\bar{u})\hat{C}_t^*(\bar{u})$  can be smooth if  $p_t(\bar{x})$  is approximately proportional to  $f^*(\bar{x})$  (i.e.,  $\hat{C}_t^*(\bar{u})$  becomes approximately constant). The key difference to primary sample space MLT (Eq. 11) is that, because of the additional term  $\hat{w}_t(\bar{u})$ , we do not need  $p_t(\bar{x})$  to fit  $f^*(\bar{x})$  well for all  $\bar{x}$ , but only locally where  $w_t(\bar{x})$  is large enough. Being a primary space approach, our method maintains the ease of implementation.

This extra flexibility is naturally combined with the MIS framework. A provably optimal weighting function such as the balance heuristic [Veach and Guibas 1995] indeed ensures that  $w_t(\bar{x})$  automatically gets small in the region where  $p_t(\bar{x})$  is a bad approximation of  $f^*(\bar{x})$ . Analogous to serial tempering, the probability of switching from the  $t$ -th to the  $t'$ -th target distribution then becomes:

$$r(t \rightarrow t') = \frac{\hat{w}_{t'}(\bar{u})\hat{C}_{t'}^*(\bar{u})q(t' \rightarrow t)}{\hat{w}_t(\bar{u})\hat{C}_t^*(\bar{u})q(t \rightarrow t')}. \quad (18)$$

**Final Algorithm** In practice, we can combine the change of techniques (Eq. 18) and the regular MCMC update (Eq. 17). Given the current state  $(\bar{u}, t)$  and the proposal  $(\bar{v}, t')$ , we accept the proposal with the probability  $\min(1, a((\bar{u}, t) \rightarrow (\bar{v}, t')))$  where

$$a((\bar{u}, t) \rightarrow (\bar{v}, t')) = \frac{\hat{w}_{t'}(\bar{v})\hat{C}_{t'}^*(\bar{v})q((\bar{v}, t') \rightarrow (\bar{u}, t))}{\hat{w}_t(\bar{u})\hat{C}_t^*(\bar{u})q((\bar{u}, t) \rightarrow (\bar{v}, t'))}. \quad (19)$$

Using a symmetric probability density  $q((\bar{v}, t') \rightarrow (\bar{u}, t)) = q((\bar{u}, t) \rightarrow (\bar{v}, t'))$  it further simplifies to

$$a((\bar{u}, t) \rightarrow (\bar{v}, t')) = \frac{\hat{w}_{t'}(\bar{v})\hat{C}_{t'}^*(\bar{v})}{\hat{w}_t(\bar{u})\hat{C}_t^*(\bar{u})}. \quad (20)$$

Our final algorithm thus can be seen as primary sample space MLT with the extended state space  $(\bar{u}, t)$  and the acceptance probability defined as above. Even though they might look similar, primary sample space MLT and our framework are fundamentally different in that an actual sample  $\bar{x}$  is no longer associated with the primary space state  $\bar{u}$  by the *unique* mapping  $P^{-1}(\bar{u})$ , but by the mapping  $P_t^{-1}(\bar{u})$  which *changes* according to the current state  $t$ .

The advantage of our method is that it maintains the simplicity of primary sample space MLT, while effectively utilizing all the available information of  $p_t(\bar{x})$  and  $P_t^{-1}(\bar{u})$ . Primary sample space MLT can be seen a special case of our method with a fixed  $t$ . While the end result might look deceptively trivial, our framework is the first to achieve the exploitation of the extra knowledge available for MIS in MCMC sampling. We would also like to note that bidirectional mutation in original MLT can be viewed as a special case of our formulation with the balance heuristic using an independent mutation. Our formulation however is not limited to the balance heuristic as we discuss later.

## 4 Multiplexed Metropolis Light Transport

We now describe our rendering algorithm, *Multiplexed Metropolis Light Transport* (Multiplexed MLT or MMLT) based on the primary space serial tempering framework. The key characteristic is that an extra Markov chain state determines how the rest of Markov chain states are mapped to a path (Eq. 16 and Fig. 3).

Algorithms 1 and 2 show pseudocode of PSSMLT and MMLT with highlighted differences. Note that the basic structures of the two algorithms are the same. The only major change is that MMLT limits the connection of subpaths to a single technique specified by the current state of the Markov chain. By this change, our algorithm is able to dedicate more computational effort to the techniques that contribute more to the image.

**Initialization** Algorithm 2 shows that we need to select a specific path length  $k$  to choose one of the techniques. To this end, we estimate the average image contribution of paths of length  $k$  using bidirectional path tracing as  $b_k \approx \frac{1}{N_k} \sum_{i=1}^{N_k} \hat{w}_t^*(\bar{u})\hat{C}^*(\bar{u})$ , and select the path length by importance sampling the discrete PDF defined by  $b_k$ . Each  $b_k$  can be estimated either once at the initialization stage or progressively during the integration stage. We used the former approach to produce all the results in the paper for simplicity. For a more robust implementation, the latter approach is less likely to miss rare bounces when an initial estimate of  $b_k$  is zero but  $b_k > 0$ . The contribution from each bounce has to be divided by the probability of the selected path length. The remaining initialization process is the same as for PSSMLT, where we could also employ start-up bias elimination [Veach and Guibas 1997].

**Optimization for Multiple Bounces** While our framework does not limit us to do so, we found that keeping one separate Markov chain for each path length is more efficient. The reason is that paths of different lengths are essentially contributing to different integrals (see also Eq. 4). We thus use  $L$  independent chains for the different path lengths  $k \in [1, L]$ , where  $L$  is the maximum path length.  $L$  can be either specified by a user, or automatically decided by the maximum path length reached during the evaluations of  $b_k$ . It

**Algorithm 1** : PSSMLT: CURRENT stores a vector of random numbers as well as its resulting path contribution. COMBINE returns the weighted contribution of all possible bidirectional connections of the subpaths. MAX returns the maximum luminosity of the weighted bidirectional contributions.

---

```

Proposal. $\bar{u}$   $\leftarrow$  MUTATE(Current. $\bar{u}$ )
EyePath  $\leftarrow$  SAMPLEEYEPATH(Proposal. $\bar{u}$ )
LightPath  $\leftarrow$  SAMPLELIGHTPATH(Proposal. $\bar{u}$ )
Proposal.Contribution  $\leftarrow$  COMBINE(EyePath, LightPath)
 $a \leftarrow \text{MAX}(\text{Proposal.Contribution})/\text{MAX}(\text{Current.Contribution})$ 
if Random  $\leq a$  then
    Current  $\leftarrow$  Proposal
end if
ACCUMULATECONTRIBUTION(Current.Contribution)

```

---

could also be lazily extended during the integration process [Kelemen et al. 2002]. The storage cost for multiple chains is negligible (typically less than 100 KB).

**Path Generation** Analogous to PSSMLT, the state of the Markov chain is a vector of random numbers  $\bar{u}_k \in [0; 1]^{O(k)}$  used for path sampling. The actual number of components in  $\bar{u}_k$  depends on the implementation (e.g., if each vertex uses three random numbers then  $\bar{u}_k \in [0; 1]^{3(k+1)}$ ). We also have the extra state  $u_{k,t} \in [0; 1]$  to specify the sampling technique  $P_t^{-1}(\bar{u})$ .

This extra state determines the number of vertices for the eye and light subpaths for a BPT connection. For paths of length  $k$ , the eye subpath consists of  $t = \lfloor (k+2)u_{k,t} \rfloor \in [0, k+1]$  vertices and the light subpath of  $s = (k+1) - t$  vertices (there are  $k+2$  possible techniques). The two subpaths are then connected deterministically to compute the path contribution. Unlike bidirectional path tracing, we connect subpaths only at their endpoints.

**Mutation and Acceptance Probability** We used the same mutation function for  $\bar{u}_k$  as Kelemen et al. [2002] (i.e., the combination of large-step and regular mutations). The extra state  $u_{k,t}$  to specify the sampling technique is mutated similarly to the other states, while it could potentially use a different mutation function. Since these mutations are all symmetric, we do not need to evaluate transition probability density functions. The acceptance probability  $a$  is simply the quotient of the luminosities of the weighted contributions of the mutated path and the current path (Eq. 20).

**Accumulation of Path Contributions** In addition to the scaling by the reciprocal of the discrete probability density of the selected path length as noted before, we also need to scale each contribution by the number of techniques  $k+2$ . This scaling corresponds to the fact that the chain explores  $k+2$  different spaces. The example code in the supplementary materials provides more details.

## 5 Results

We implemented bidirectional path tracing (BPT), primary sample space MLT (PSSMLT), Veach and Guibas' original MLT, stochastic progressive photon mapping (SPPM) [Hachisuka and Jensen 2009], and our algorithm (MMLT) in a single framework. The reference images are rendered by PSSMLT with days of computation, except for the box scene in Fig. 8 where we had to use SPPM. All comparisons are equal-time using a maximum of eight bounces rendered with a Intel Core i7-3930K at 3.2 GHz using six cores. The parameters for original MLT (such as mutation size for each perturbation strategy and probabilities to delete/insert a certain number of vertices in bidirectional mutation) are tweaked manually to achieve the best results, and its behavior has been validated against the im-

**Algorithm 2** : Differences of Multiplexed MLT to PSSMLT (Alg. 1). CURRENT additionally stores a random number  $u_{k,t}$  to specify a technique (determining subpath lengths  $s$  and  $t$ ). COMBINE connects subpaths only at their endpoints. The acceptance probability  $a$  uses the weighted luminosity as is.

---

```

k  $\leftarrow$  SAMPLEPATHLENGTH()
Proposal. $[\bar{u}_k, u_{k,t}]$   $\leftarrow$  MUTATE(Current. $[\bar{u}_k, u_{k,t}]$ )
t  $\leftarrow$   $\lfloor (k+2)u_{k,t} \rfloor$ 
s  $\leftarrow$   $(k+1) - t$ 
EyePath  $\leftarrow$  SAMPLEEYEPATH(Proposal. $\bar{u}_k$ , t)
LightPath  $\leftarrow$  SAMPLELIGHTPATH(Proposal. $\bar{u}_k$ , s)
Proposal.Contribution  $\leftarrow$  COMBINE(EyePath, LightPath, t, s)
 $a \leftarrow \text{Proposal.Contribution}/\text{Current.Contribution}$ 
if Random  $\leq a$  then
    Current  $\leftarrow$  Proposal
end if
ACCUMULATECONTRIBUTION(Current.Contribution)

```

---

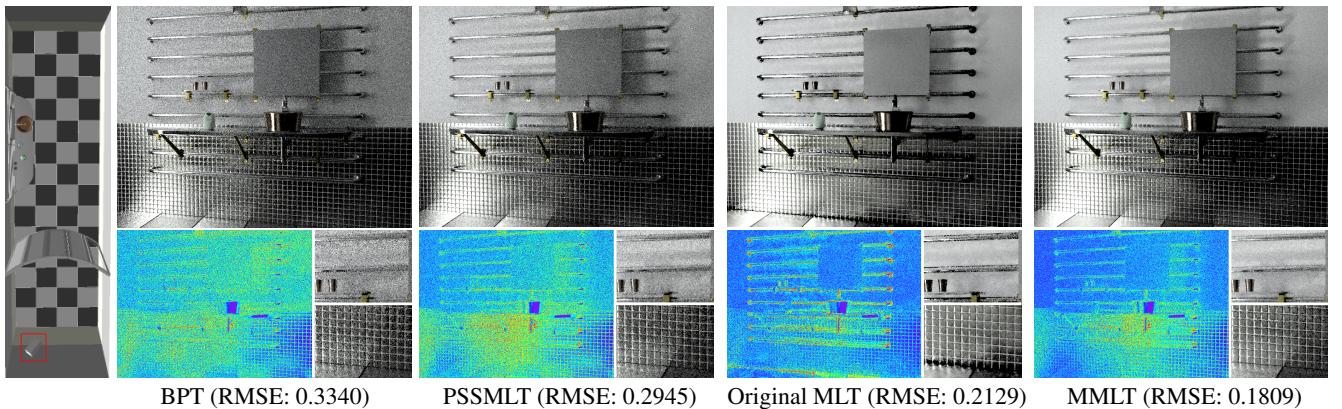
plementation in Mitsuba [Jakob 2010]. We used the same primary sample space mutation parameters for primary sample space MLT and our algorithm. A simplified implementation of our algorithm and PSSMLT in a single C++ file is available on our website.

**Comparison to Other Rendering Algorithms** Fig. 4 demonstrates the robustness of our algorithm. This scene is challenging to render as it contains both highly glossy surfaces and difficult visibility. We compare to BPT, PSSMLT, and original MLT with an equal render time of 60 minutes. BPT and PSSMLT result in noisy images as they cannot efficiently find visible paths in this scene. PSSMLT does not improve the efficiency much over BPT in this scene as the target distribution is defined as the maximum of *all* BPT connections. This wastes computation when only few of the connections yield a non-zero contribution due to visibility.

Fig. 6 compares sequences of images of original MLT and MMLT for the parts of the scene showing diffuse and glossy surfaces. The rendering using the original MLT method is less noisy in some regions, but shows splotchy artifacts on glossy surfaces. This is because no mutation strategy captures these light transport paths well. It is conceivable that adding a specially crafted mutation strategy would improve the performance. However, trying to cover all possible types of paths would require us to design and implement new mutation strategies. Furthermore, adding customized mutation strategies wastes samples when a scene does not include the intended path types (as noted also by Veach and Guibas [1997]).

Our MMLT renders an image that is overall less noisy than BPT and PSSMLT, while capturing glossy transport better than original MLT. MMLT automatically switches to the most appropriate technique as defined by the MIS weights, while difficult visibility is still handled well by MCMC sampling. We emphasize that this result is achieved by a very simple algorithm without any customized mutation for specific types of paths. This automatic adaptation is a consequence of the fusion of MIS and MCMC sampling (Sect. 4).

**Sample Counts and Effective Path Ratio** Fig. 1 shows an equal-time comparison with the well-known door scene [Veach and Guibas 1997] with highly glossy surfaces. For diffuse surfaces such as the back wall, original MLT produces very accurate results. On the other hand, there are distracting artifacts on glossy surfaces similar to the scene in Fig. 4. PSSMLT is more accurate on glossy surfaces but noisy overall. Our algorithm works well in both regions. In order to assess the efficiency of our algorithm over PSSMLT, we computed the ratio of the number of samples with non-zero contributions over the total number of samples. PSSMLT resulted in 11.81% in this scene and MMLT resulted in 29.75%, which is an improvement of approximately 2.5 times.



**Figure 4:** Bathroom scene with glossy surfaces rendered in an hour. The light source (red box in the overview, left) is facing towards the corner of the shower booth. Bidirectional path tracing (BPT) is inefficient due to the complex visibility, and primary sample space MLT (PSSMLT) can only slightly improve the result. While the original MLT method handles complex visibility well, no mutation strategy efficiently captures glossy transport. Our algorithm (MMLT) handles both complex visibility and glossy transport by automatically selecting an efficient sampling technique during MCMC sampling.

Fig. 5 visualizes the individual contributions from each technique for paths of length six and the corresponding sample count distributions for PSSMLT and MMLT. This visualization is not possible with original MLT as there is no concept of sampling techniques.

As we mentioned in Sect. 3, PSSMLT with bidirectional connections is different from our algorithm since PSSMLT relies on a single *unique* mapping of primary space samples to paths, while our algorithm automatically uses *multiple* mappings. This difference is evident in the sample count images (bottom two rows): PSSMLT essentially results in the same number of samples for all the techniques regardless of their weighted contributions, while our algorithm adaptively distributes the samples according to their contributions.

The visualization in the last column further confirms this difference. PSSMLT results in an almost entirely gray image if we visualize relative sample counts of three techniques as RGB values. MMLT results in relative sample counts that roughly correspond to the magnitudes of relative contributions under the same visualization scheme.

**Manifold Exploration** Jakob and Marschner [2012] introduced a mutation strategy, manifold exploration (ME), which is specially designed to improve the handling of specular and highly glossy paths. Like other mutations in original MLT, a path is mutated by directly modifying its vertices. In order to evaluate how our algorithm compares in a scene where manifold exploration is expected to work well, we rendered the Cornell box with glossy surfaces (Fig. 7).

For this comparison, we use the implementation of ME and original MLT (as well as the reference rendering with PSSMLT) available in Mitsuba. We intentionally selected a geometrically simple scene to avoid the influence of low-level optimization (e.g., for ray-triangle intersection) within each framework.

ME effectively compensates the lack of efficient mutation strategy in the original MLT method and resulted in a more accurate image. While our method does not outperform ME, it still works better than original MLT. As such, ME is certainly a good option to improve the efficiency of original MLT for glossy transport. The simplicity and its advantage over original MLT, however, can make our algorithm an attractive alternative in many cases. We however do not claim that our method outperforms ME in general.

## 6 Discussion

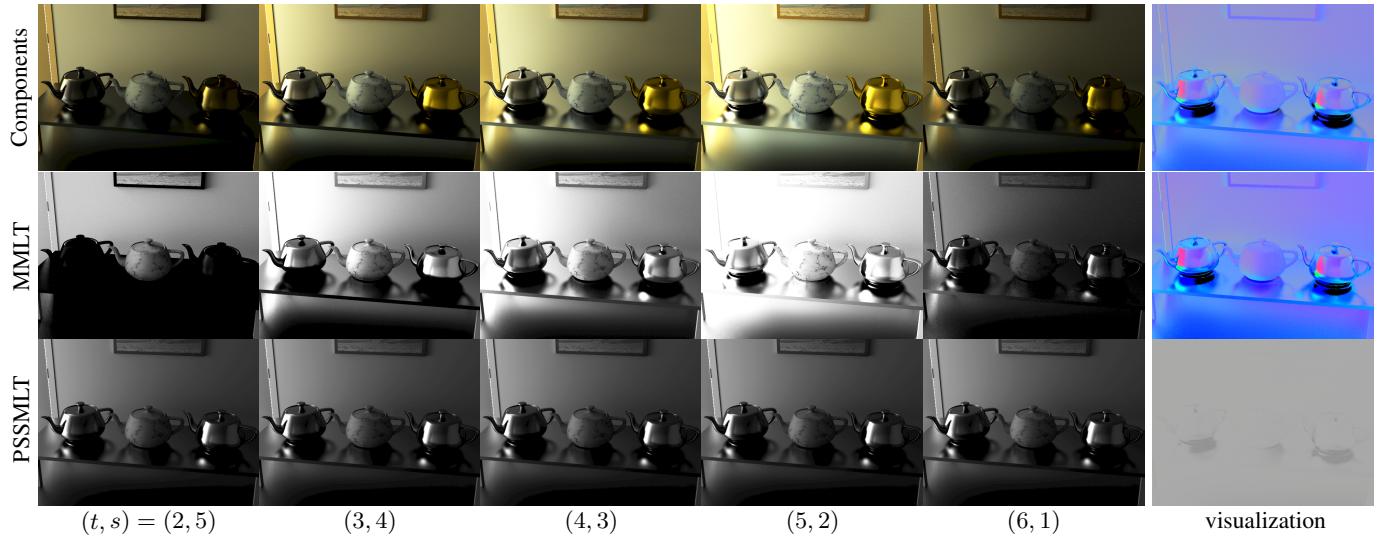
**Limitations** Like any other local path sampling algorithms, MMLT is inefficient at (nearly) non-samplable paths, such as specular-diffuse-specular paths. Fig. 8 compares renderings of the box-scene by Hachisuka et al. [2008] with the original MLT method and MMLT. An equal-time comparison with SPPM revealed that SPPM is significantly more efficient at capturing specular reflections and refractions of caustics than all the algorithms we tested including manifold exploration; note that this scene has been specifically designed to make unbiased path seeding impractical.

In general, we do not claim that MMLT *always* outperforms other MCMC methods. Figure 9 demonstrates such a case using a recreation of the original door scene [Veach and Guibas 1997]. While MMLT still outperforms PSSMLT, original MLT is more efficient than MMLT since its mutation strategies sufficiently cover all the possible types of paths in this scene. Likewise, manifold exploration can work better in some scenes as it is a global path sampling algorithm (Figure 7).

We would like to note that MMLT shares fundamental issues common to all the MCMC rendering algorithms such as a lack of perfect stratification across pixels, potentially distracting artifacts for animation, generally unreliable convergence tests, and remaining bright pixels in otherwise converged renderings.

**Parameters** Similar to any MCMC sampling method, MMLT has parameters that affect its performance. However, unlike the original MLT method where several mutation strategies (bidirectional, caustics, lens, multi-chain etc.) have their own set of parameters to tweak, our algorithm has essentially only two global parameters: mutation size in primary space and large step probability. This is because MMLT uses a single mutation strategy in primary space only, and different types of paths (e.g., caustics and glossy transport) are automatically handled by exploring different techniques.

We can thus make our algorithm *parameter-free* in practice by using adaptive MCMC in primary space [Hachisuka and Jensen 2011] for mutation size adjustment and automatic tuning of the large step probability [Zsolnai and Szirmay-Kalos 2013]. It would however be interesting to see if a customized mutation for our algorithm is beneficial, e.g. for the technique index. All the comparisons in this paper use non-adaptive parameters for both PSSMLT and our algorithm in order to ensure the fairness of comparisons.



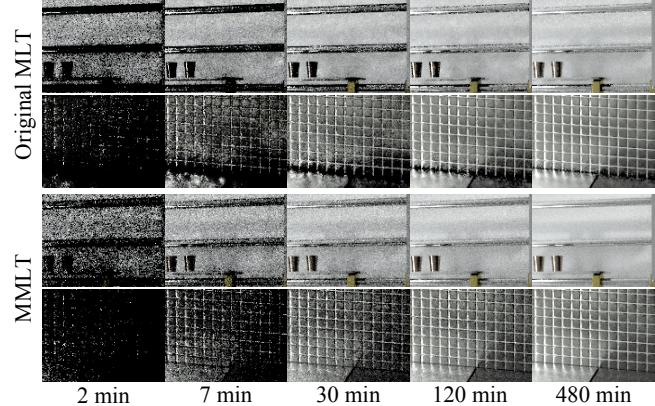
**Figure 5:** Analysis of the usage of different techniques for paths of length six in PSSMLT and MLT: Top row: contributions from each technique to the image (with adjusted brightness and tone mapping). Middle row: the number of samples per pixel in our MMLT is roughly proportional to the contribution. Bottom row: PSSMLT uses bidirectional path tracing as a sampler, but does not adaptively distribute samples to each technique. The right-most column visualizes the dominant techniques (3,4), (4,3), and (5,2) by assigning their relative contributions (top) as well as the sample counts in MMLT and PSSMLT to RGB channels. Sample counts in MMLT closely match the contributions. Note that PSSMLT results in a gray image under the same visualization scheme as it equally distributes samples to techniques.

**Optimal Weighting Functions** MIS has some flexibility in the choice of weighting functions [Veach and Guibas 1995]. Since our framework supports arbitrary weighting functions, different weighting functions potentially result in different performance. We used the balance heuristic for all the results in this paper, but we do not claim that this is the optimal choice. Bidirectional lightcuts [2012] is a prominent example where customizing weighting functions can significantly improve results. While we did not find a weighting function with better performance than the balance heuristic in our experiments so far, it might be interesting future work.

**Possible Optimization** The original MLT method is more effective at path reuse than PSSMLT and MMLT, as it can directly keep a part of the path during the mutation process. It however does not mean that path reusing is impossible in PSSMLT and MMLT. In computational statistics, it is well known that we do not need to update all states of the Markov chain simultaneously. We can use the so-called *block-wise update* [Brooks et al. 2011] and keep a part of a path whenever it is not affected by the mutation. This optimization can be done without any change to the theory. Contrary to the common misconception that parallelization is extremely difficult with MCMC rendering methods, we can simply run multiple Markov chains in parallel and combine the output images. Both Mitsuba and our framework use this approach.

## 7 Comparison to Previous Work

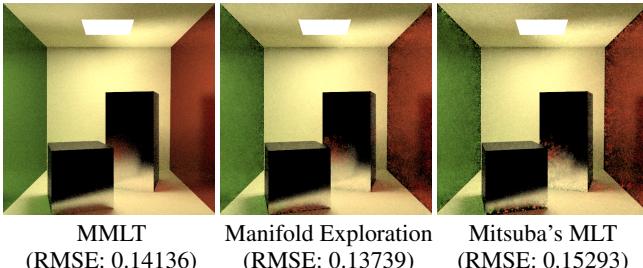
Unified path sampling/vertex connection and merging [Hachisuka et al. 2012; Georgiev et al. 2012] formulates a MIS combination of photon density estimation and Monte Carlo path integration to better handle non-samplable paths. The extra adaptivity *across techniques* in our method could be used to focus samples to a potentially even smaller set of useful techniques in this formulation. It is, however, not entirely clear how this formulation can be efficiently combined with MCMC sampling. Path space regularization [Kaplanyan and Dachsbacher 2013; Bouchard et al. 2013] also addresses the problem of non-samplable paths and it is potentially a viable option for a combination with MCMC sampling like ours.



**Figure 6:** Close-ups of Fig. 4 which highlight the convergence behavior. While the original MLT method renders diffuse surfaces well, its convergence is slower on glossy surfaces. MMLT results in more uniform convergence across different regions.

Variants of MLT and other MCMC techniques have been proposed in graphics literature. Energy redistribution path tracing [Cline et al. 2005] strives to stratify perturbations and to improve exploration of the sample space by using a large number of short Markov chains starting from stratified locations on the image. This idea can potentially be combined with any MCMC rendering algorithm as well as ours (e.g., population Monte Carlo [Lai et al. 2007]).

Kitaoka et al. [2009] applied replica exchange [Swendsen and Wang 1986] using a set of user-defined target distribution for different path types (caustics, specular-diffuse-specular, indirect diffuse illumination, etc). While our idea of using multiple distributions might look deceptively akin to their work, our framework does not rely on a user-defined classification of paths at all. Such a user-defined classification becomes nontrivial with glossy materials or, in general, in any scene configuration that was not originally considered. For example, it is not clear whether reflections from a glossy surface should be classified as caustics or indirect bounce.



**Figure 7:** Glossy Cornell box rendered in three minutes with MMLT and Mitsuba’s implementation of manifold exploration (ME) and the original MLT method. ME improves the efficiency of original MLT in the presence of glossy surfaces. MMLT achieves a comparable result with a much simpler algorithm.

This is the exact ambiguity that MIS resolves in ordinary Monte Carlo via provably optimal weighting. MMLT incorporates this general weighting approach of MIS into MCMC for the first time. Kitaoka et al. also pointed out that, since replica exchange runs a Markov chain for each distribution regardless of its contribution, their algorithm becomes inefficient when many distributions result in small contributions. In contrast, MMLT automatically avoids spending samples on such distributions, since a single extended Markov chain less frequently visits distributions with small contributions.

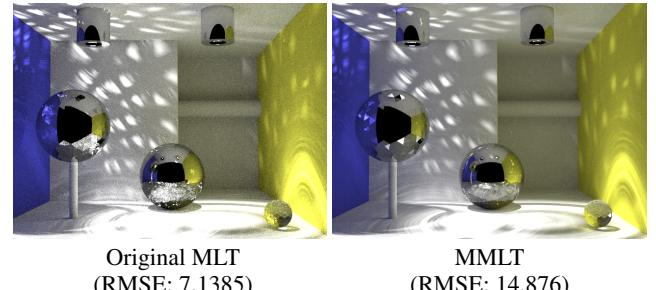
Hachisuka and Jensen [2011] applied an adaptive MCMC sampling method for photon density estimation (Chen et al. [2011] proposed a similar approach without adaptive MCMC). Their application is limited to light subpaths for creating photons, and camera subpaths are sampled by an ordinary Monte Carlo method as in path tracing. This fixed decomposition of complete paths is the reason why their approach is not efficient for glossy surfaces, which our method handles well. Since their formulation uses the primary space, their adaptive MCMC component can be directly applied to MMLT.

Although we believe that the use of the primary space is the key component that makes our algorithm easy to implement, it would be interesting to see if we can combine the original MLT method and MMLT into a single algorithm to improve the performance in some special cases. This combination would potentially make manifold exploration [Jakob and Marschner 2012] accessible to our algorithm. However, we emphasize again that the simplicity of implementation is one of the key benefits of our algorithm, which would be sacrificed in a combination with original MLT or ME.

Gradient-domain MLT [Lehtinen et al. 2013] computes an approximation of the image and its gradients to reconstruct the final image by solving a Poisson equation. In its implementation, gradient-domain MLT uses an extended variant of original MLT with manifold exploration. The basic concept of gradient-domain MLT, however, is orthogonal to, and can thus be combined with, any MCMC sampling technique including ours. The performance of such a direct application, however, would be suboptimal due to the extra difficulty of sampling gradients.

## 8 Conclusion

We introduced a novel Markov chain sampling framework, *primary space serial tempering*, that combines the concepts of multiple importance sampling and Markov chain Monte Carlo sampling. The key idea is that a Markov chain also explores different probability density functions to generate samples from the weighted distributions from MIS. The fusion of MCMC sampling and MIS allows us to locally avoid generating samples from a probability density function that is a bad approximation of the target distribution. Based



**Figure 8:** Box scene with SDS paths rendered in 45 minutes with original MLT and ours. While original MLT works better than our method, SDS paths are better captured by a biased approach such as SPPM (which resulted in RMS error of 1.1506). Note that specular reflections of caustics are quite noisy in both images; the differences in brightness are due to the uneven distributions of samples in MCMC.

on this general framework, we introduced a new rendering algorithm, *multiplexed Metropolis light transport*, which automatically explores and dedicates computation time to different path sampling techniques. The method is very simple to implement, while its performance is comparable to or better than significantly more complex methods across various scenes with complex light transport. In particular, it is effective at handling glossy transport with difficult visibility.

We believe that multiplexed Metropolis light transport is a viable addition to the existing set of MCMC rendering algorithms due to its good performance while being a relatively simple algorithm. While we focused on surface light transport in the paper, its application to volume light transport should be straightforward. Primary space serial tempering can potentially be useful in any problem where MCMC sampling is applied, not only for rendering.

**Acknowledgements** The first author would like to thank Jacopo Pantaleoni for his feedback on initial ideas. The bathroom scene is modeled and provided by Youichi Kimura. We modified and used the door scene originally made by Miika Aittala, Samuli Laine, and Jaakko Lehtinen. We would like to thank Troels Mortensen, Heine Stokholm, and Florian Simon for comments on the draft.

## References

- ARVO, J. 1986. Backward ray tracing. In *Developments in Ray Tracing, ACM SIGGRAPH Course Notes*, 259–263.
- BOUCHARD, G., IEHL, J.-C., OSTROMOUKHOV, V., AND POULIN, P. 2013. Improving Robustness of Monte-Carlo Global Illumination with Directional Regularization. In *SIGGRAPH Asia Technical Briefs*, 22:1–22:4.
- BROOKS, S., GELMAN, A., JONES, G., AND MENG, X.-L. 2011. *Handbook of Markov Chain Monte Carlo*. Taylor & Francis US.
- CHEN, J., WANG, B., AND YONG, J.-H. 2011. Improved stochastic progressive photon mapping with Metropolis sampling. *Computer Graphics Forum* 30, 4, 1205–1213.
- CLINE, D., TALBOT, J., AND EGBERT, P. K. 2005. Energy redistribution path tracing. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 24, 3, 1186–1195.
- GEORGIEV, I., KŘIVÁNEK, J., DAVIDOVIČ, T., AND SLUSALLEK, P. 2012. Light transport simulation with vertex connection and merging. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 31, 6, 192:1–192:10.



**Figure 9:** Comparison of original MLT (left), primary sample space MLT (center), and multiplexed MLT (right) for the door scene without glossy materials (90 min). Original MLT works quite well in this scene since their mutation strategies are well suited to handle paths with Lambertian and specular materials. While multiplexed MLT is slightly outperformed by original MLT, it still improves upon primary sample space by automatically distribute samples across different techniques according to their weighted contributions.

- GEYER, C. J., AND THOMPSON, E. A. 1995. Annealing Markov chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association* 90, 431, 909–920.
- HACHISUKA, T., AND JENSEN, H. W. 2009. Stochastic progressive photon mapping. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 28, 5, 141:1–141:8.
- HACHISUKA, T., AND JENSEN, H. W. 2011. Robust adaptive photon tracing using photon path visibility. *ACM Transactions on Graphics* 30, 5, 114:1–114:11.
- HACHISUKA, T., OGAKI, S., AND JENSEN, H. W. 2008. Progressive photon mapping. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 27, 5, 130.
- HACHISUKA, T., PANTALEONI, J., AND JENSEN, H. W. 2012. A path space extension for robust light transport simulation. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 31, 6, 191:1–191:10.
- HASTINGS, W. K. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57, 1, 97–109.
- JAKOB, W., AND MARSCHNER, S. 2012. Manifold exploration: a Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 31, 4, 58:1–58:13.
- JAKOB, W., 2010. Mitsuba renderer. <http://www.mitsuba-renderer.org>.
- JENSEN, H. W. 1996. Global illumination using photon maps. In *Proc. Eurographics Workshop on Rendering*, 21–30.
- KAJIYA, J. T. 1986. The rendering equation. *Computer Graphics (Proceedings of SIGGRAPH '86)*, 143–150.
- KAPLANYAN, A. S., AND DACHSBACHER, C. 2013. Path space regularization for holistic and robust light transport. *Computer Graphics Forum (Proc. of Eurographics)* 32, 2.
- KELEMEN, C., SZIRMAKALOS, L., ANTAL, G., AND CSONKA, F. 2002. A simple and robust mutation strategy for the Metropolis light transport algorithm. *Computer Graphics Forum* 21, 3, 531–540.
- KITAOKA, S., KITAMURA, Y., AND KISHINO, F. 2009. Replica exchange light transport. *Computer Graphics Forum* 28, 8, 2330–2342.
- LAFORTUNE, E. P., AND WILLEMS, Y. D. 1993. Bi-directional path tracing. In *Compugraphics '93*, 145–153.
- LAI, Y.-C., FAN, S. H., CHENNEY, S., AND DYER, C. 2007. Photorealistic image rendering with population Monte Carlo energy redistribution. In *In Rendering Techniques 2007 (Proceedings of the Eurographics Symposium on Rendering)*, 287–295.
- LEHTINEN, J., KARRAS, T., LAINE, S., AITTALA, M., DURAND, F., AND AILA, T. 2013. Gradient-domain Metropolis light transport. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 32, 4.
- MARINARI, E., AND PARISI, G. 1992. Simulated tempering: a new Monte Carlo scheme. *EPL (Europhysics Letters)* 19, 6, 451.
- METROPOLIS, N., AND ULAM, S. 1949. The Monte Carlo method. *Journal of the American Statistical Association* 44, 247, 335–341.
- METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H., AND TELLER, E. 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* 21, 1087.
- SWENDSEN, R. H., AND WANG, J.-S. 1986. Replica Monte Carlo simulation of spin glasses. *Physical Review Letters* 57, 21, 2607–2609.
- VEACH, E., AND GUIBAS, L. J. 1995. Optimally combining sampling techniques for Monte Carlo rendering. In *SIGGRAPH '95*, 419–428.
- VEACH, E., AND GUIBAS, L. 1997. Metropolis light transport. In *SIGGRAPH '97*, 65–76.
- VEACH, E. 1998. *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford University, USA. AAI9837162.
- WALTER, B., KHUNGURN, P., AND BALA, K. 2012. Bidirectional lightcuts. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 31, 4, 59:1–59:11.
- ZSOLNAI, K., AND SZIRMAKALOS, L. 2013. Automatic parameter control for Metropolis light transport. In *Eurographics short papers*, 53–56.