Charted Metropolis Light Transport

Jacopo Pantaleoni*
NVIDIA

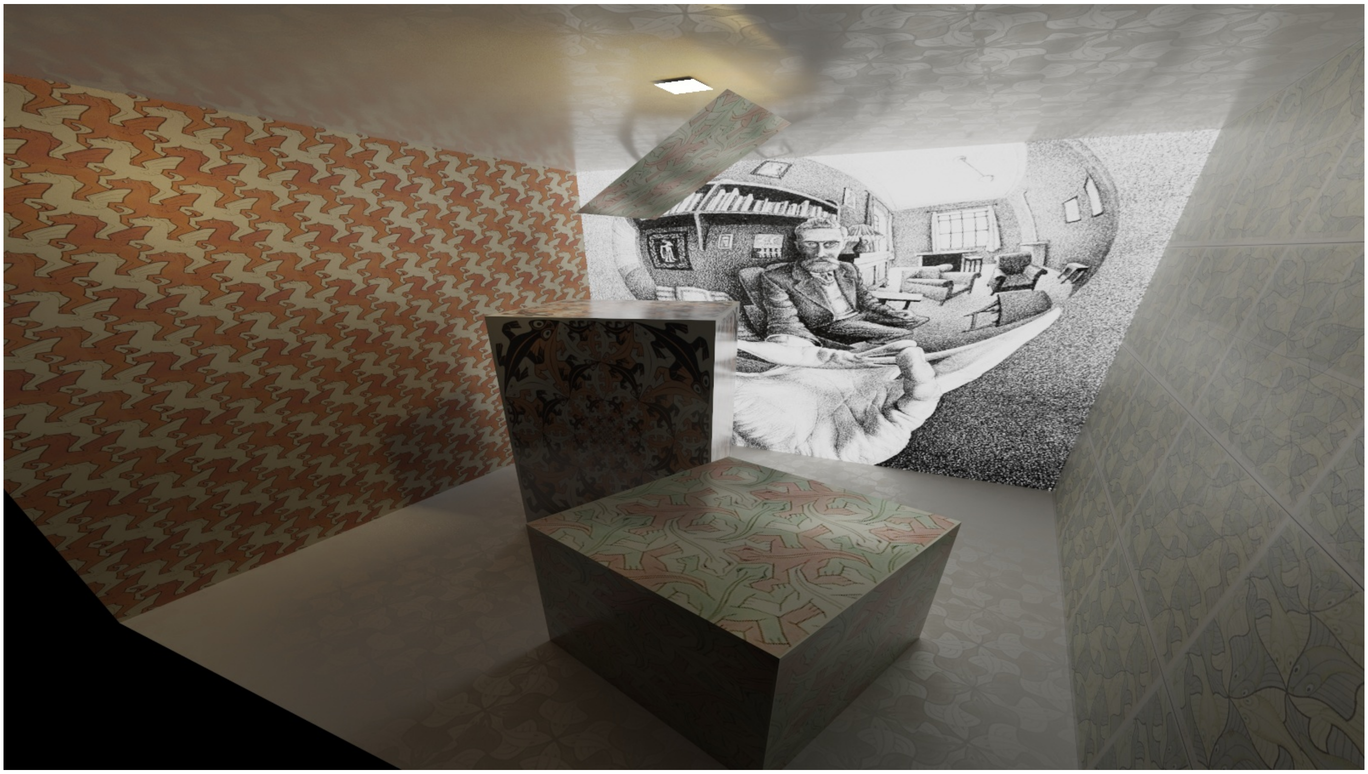


Figure 1: Escher's Box. *Charted Metropolis light transport considers path sampling methods and their primary sample space coordinates as charts of the path space, allowing to easily jump between them. In particular, it does so without requiring classical invertibility of the sampling methods, making the algorithm practical even with complex materials.*

Abstract

In this manuscript, inspired by a simpler reformulation of primary sample space Metropolis light transport, we derive a novel family of general Markov chain Monte Carlo algorithms called *charted Metropolis-Hastings*, that introduces the notion of *sampling charts* to extend a given sampling domain and make it easier to sample the desired target distribution and escape from local maxima through coordinate changes. We further apply the novel algorithms to light transport simulation, obtaining a new type of algorithm called *charted Metropolis light transport*, that can be seen as a bridge between primary sample space and path space Metropolis light transport. The new algorithms require to provide only right inverses of the sampling functions, a property that we believe crucial to make them practical in the context of light transport simulation. We further propose a method to integrate density estimation into this framework through a novel scheme that uses it as an independence sampler.

CR Categories: I.3.2 [Graphics Systems C.2.1, C.2.4, C.3)]: Stand-alone systems—; I.3.7 [Three-Dimensional Graphics and Realism]: Color,shading,shadowing, and texture—Raytracing;

Keywords: global illumination, light transport simulation, Markov Chain Monte Carlo

1 Introduction

Light transport simulation can be notoriously hard. The main problem is that forming an image requires evaluating millions of infinite dimensional integrals, whose integrands, while correlated, may contain an infinity of singularities and different modes at disparate frequencies. Many approaches have been proposed to solve the rendering equation, though most of them rely on variants of Monte Carlo integration. One of the most robust algorithms, Metropolis light trans-

*e-mail: jpantaleoni@nvidia.com

port (MLT), has been proposed by Veach and Guibas in 1997 [Veach and Guibas 1997] and has been later extended in many different ways. One of the most commonly used variants is primary sample space MLT [Kelemen et al. 2002], partly because in some scenarios it is more efficient (though not always), partly because it is generally considered simpler to implement. However, both variants are still considered relatively complex compared to other algorithms that are not based on Markov chain Monte Carlo (MCMC) methods, or that employ a simplified target distribution [Hachisuka and Jensen 2011].

In this paper we show that the original primary sample space MLT uses a suboptimal target distribution, and that fixing the problem makes the algorithm more efficient while also greatly simplifying it at the same time.

Inspired by this simpler formulation, we then propose a novel family of general Markov chain Monte Carlo algorithms called *charted Metropolis-Hastings* (CMH). The core idea is to extend the concept of primary sample spaces into that of *sampling charts* of the target space, extending the domain of the desired target distribution and introducing novel mutation types that swap charts and perform coordinate changes (analogous to those found in regular tensor calculus) in order to craft better proposals.

We then apply the new MCMC algorithm to light transport simulation, obtaining a type of algorithms called *charted Metropolis light transport* (CMLT), that considers all local path sampling methods as parameterizations of the path space manifold, and employs stochastic path inversion as a way to perform coordinate transformations between charts. Our algorithm is made practical by avoiding the requirement to use fully invertible path sampling methods, and only requiring stochastic right inverses, a property we believe fundamental. This new type of algorithms can be seen as fundamentally bridging the difference between the original formulation of path space MLT and the primary sample space version, allowing to easily combine both.

Finally, we propose a novel scheme to integrate density estimation inside MCMC frameworks that exploits its robustness with respect to sampling near-singular and singular paths while mantaining overall simplicity and efficiency of implementation.

A preprint of this work has been published on ArXiv [Anonymous 2016].

2 Preliminaries

Veach [1997] showed that light transport simulation can be expressed as the solution of per-pixel integrals of the form:

$$I_j = \int_\Omega f_j(\mathbf{x}) d\mu(\mathbf{x}) \tag{1}$$

where $\Omega = \bigcup_{k=1}^{\infty} \Omega_k$ represents the space of light paths of all finite lengths k and μ is the area measure.

For a path $\mathbf{x} = x_0 \rightarrow x_1 \cdots \rightarrow x_k$, the integrand is defined by the *measurement contribution function*:

$$\begin{aligned} f_j(\mathbf{x}) &= L_e(x_0 \rightarrow x_1) \\ &\cdot \prod_{i=0}^{k-1} \left[f_s(x_{i-1} \rightarrow x_i \rightarrow x_{i+1}) G(x_i \leftrightarrow x_{i+1}) \right] \\ &\cdot W_e^j(x_{k-1} \rightarrow x_k) \end{aligned} \tag{2}$$

where L_e is the surface emission, W_e^j is the pixel response (or emitted importance), f_s denotes the local BSDF and G is the geometric term. To simplify notation, in the following we will simply omit the pixel index and consider the positions $f = f_j$ and $I = I_j$.

He further showed that if one employs a family $\mathcal{F}_k = \{s, t : s + t - 1 = k\}$ of *local path sampling* techniques to sample subpaths $\mathbf{y} = y_0 \ldots y_{s-1}$ and $\mathbf{z} = z_0 \ldots z_{t-1}$ from the light and the eye respectively, and build the joined path $\mathbf{x} = y_0 \ldots y_{s-1} z_{t-1} \ldots z_0$, an unbiased estimator of I can be obtained as a *multiple importance sampling* combination:

$$F = \sum_{s,t} C_{s,t}(\mathbf{x}) \tag{3}$$

with the following definitions:

$$C_{s,t}(\mathbf{x}) = w_{s,t} C^*_{s,t} \tag{4}$$

$$C^*_{s,t}(\mathbf{x}) = \frac{f(\mathbf{x})}{p_{s,t}(\mathbf{x})} \tag{5}$$

$$p_{s,t}(\mathbf{x}) = p_s(\mathbf{x}) p_t(\mathbf{x}) \tag{6}$$

$$w_{s,t} = \frac{p_{s,t}(\mathbf{x})}{\sum_{(i,j) \in \mathcal{F}_k} p_{i,j}(\mathbf{x})} \tag{7}$$

While a complete analysis of the above formulas is beyond the scope of this paper (we refer the reader to [Veach 1997]), we feel it is important to make the following remarks:

remark 1: the multiple importance sampling weight $w_{s,t}$ can also be expressed as:

$$\begin{aligned} w_{s,t} &= \frac{\frac{p_{s,t}}{f}}{\sum_{(i,j) \in \mathcal{F}_k} \frac{p_{i,j}}{f}} \\ &= \frac{1}{C^*_{s,t}} \left(\sum_{(i,j) \in \mathcal{F}_k} \frac{1}{C^*_{i,j}} \right)^{-1} \end{aligned} \tag{8}$$

remark 2: if importance sampling is used, the connection term $C^*_{s,t}$ effectively contains only the parts of f which have not been importance sampled; particularly, if p_s and p_t importance sample all terms of the measurement contribution function up to the s-th and t-th light and eye vertex respectively, $C_{s,t}$ will be proportional to the BSDFs at the connecting vertices times the geometric term $G(y_{s-1}, z_{t-1})$. This is the only remaining singularity, which gets eventually suppressed in $C_{s,t}$ by the multiple importance sampling weight $w_{s,t}$. In fact, simplifying equation (4), one gets:

$$C_{s,t}(\mathbf{x}) = \frac{f(\mathbf{x})}{\sum_{(i,j) \in \mathcal{F}_k} p_{i,j}(\mathbf{x})}$$

2.1 The Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is a Markov-Chain Monte Carlo method that, given an arbitrary target distribution $\pi(x)$, builds a chain of samples $X_1, X_2, \ldots$ that have π as the stationary distribution, i.e. $\lim_{n \to \infty} p(X_n) = \pi(X_n)$. The algorithm is based on two simple steps:

proposal: a new sample Y is obtained from $X = X_i$ by means of a *transition kernel* $K(Y|X)$

acceptance-rejection: X_{i+1} is set to Y with probability:

$$A(Y|X) = \min\left(1, \frac{\pi(Y)K(X|Y)}{\pi(X)K(Y|X)}\right) \qquad (9)$$

and to X_i otherwise.

Importantly, note that π can be defined up to a constant. In other words, if $\int \pi(x)dx = c$, the algorithm will simply admit π/c as its stationary distribution.

Finally, it is also possible to use mutations in which the proposal $K(Y|X) = K(Y)$ depends only on Y: in this case, the mutation type is called an independence sampler [Tierney 1994].

2.2 Primary Sample Space Metropolis Light Transport, done right

Kelemen et al [2002] showed that if one considers the transformation $T : U \to \Omega$ that is typically used to map random numbers to paths when performing forward and backward path tracing (i.e. when sampling eye and light subpaths), one can handily apply the Metropolis-Hastings algorithm on the unit hypercube U instead of bothering about the more complex path space. The advantage is that crafting mutations in U is much easier to implement - a simple Gaussian kernel will do - and will often lead to better mutations, since they will naturally follow the local BSDFs.[1] The only thing that needs to be taken care of is pulling back the desired measure from Ω to U, which is easily achieved by multiplying by the Jacobian of the transformation T, which is nothing more than the reciprocal of path probability:

$$I = \int_U f(T(u)) \left|\frac{dT(u)}{du}\right| du = \int_U \frac{f(T(u))}{p(T(u))} du \qquad (10)$$

Very unfortunately, they chose a suboptimal mapping and a suboptimal target distribution.

In fact, what they did was to consider a mapping from the product of two infinite-dimensional unit hypercubes (which itself poses some interesting definition challenges), to the product space of light and eye subpaths sampled using Russian Roulette terminated path tracing. Furthermore, instead of simply considering the single path obtained by joining the two endpoints of the respective subpaths, and using the measurement contribution function as the target distribution, they considered the sum of the MIS weighted contributions from all paths obtained joining any two vertices of the light and eye subpaths. The reason why they did this can be understood: this was the historical way to perform bidirectional path tracing. In order not to *waste* any vertex, one would *reuse* all of them at the expense of some added correlation and some added shadow rays. However, it turns out this is undesirable for several reasons:

reason 1: by joining all vertices in the generated subpaths, and summing up all the weighted contributions from the obtained paths (which are in fact truly different paths, except for the fact they share their light and eye prefixes), they were using *a target distribution which was no longer proportional to path throughput* (or rather, the measurement contribution function we are finally interested in). In other words, the obtained paths have a *skewed* distribution which is not necessarily optimal.[2]

reason 2: dealing with the infinite dimensional unit hypercubes introduces some unnecessary algorithmic complications, including the need for lazy coordinate evaluations.

reason 3: by joining all vertices in the generated subpaths, we are introducing some additional sample correlation that might not necessarily improve the per-sample efficiency. In some situations, for example in the presence of incoherent transport, it will in fact reduce it.

We now propose a much simpler variant. Let's for the moment consider the space of paths of length k, and a single technique $i \in \mathcal{F}_k$ to generate them, where i defines the number of light vertices and the number of eye vertices is given as $j = k + 1 - i$. If sampling n vertices through path tracing requires $m{\cdot}n$ random numbers, we will consider the following definition of the primary sample space:

$$U_i = [0,1]^{m\cdot i} \times [0,1]^{m\cdot(k+1-i)}. \qquad (11)$$

The transformation $T = T_i : U \to \Omega_k$ will have the following Jacobian:

$$\left|\frac{dT(u)}{du}\right| = \frac{1}{p_i(T(u))}. \qquad (12)$$

We now have two options for the choice of our target distribution. The simplest is to set:

$$\pi_i(u) = \frac{f(T(u))}{p_i(T(u))}. \qquad (13)$$

This choice keeps the corresponding path space distribution invariant relative to the area measure μ, as we have:

$$\begin{aligned} \pi_i(u)du &= \pi_i(u)p_i(T(u))|d\mu(T(u))/du|du \\ &= \pi_i(u)p_i(T(u))d\mu(T(u)) \\ &= f(T(u))d\mu(T(u)) \\ &= \bar{\pi}(T(u))d\mu(T(u)). \end{aligned} \qquad (14)$$

In other words, it ensures that all our distributions $\pi_i(u)$ are designed to have a distribution in their primary space U_i that becomes the same distribution $\bar{\pi}(\mathbf{x}) = f(\mathbf{x})$ in path space.

The second choice is to set:

$$\pi_i(u) = w_i(T(u))\frac{f(T(u))}{p_i(T(u))}, \qquad (15)$$

exploiting the fact that, while now the corresponding path space distributions $\bar{\pi}_i(\mathbf{x}) = w_i(\mathbf{x})f(\mathbf{x})$ are biased,[3] our desired path space distribution f is obtained as their sum:

$$\sum_{i\in\mathcal{F}_k} \bar{\pi}_i(\mathbf{x}) = \sum_{i\in\mathcal{F}_k} w_i(\mathbf{x})f(\mathbf{x}) = f(\mathbf{x}). \qquad (16)$$

[1] This can, however, be detrimental in cases of complex occlusion, where the original path space MLT is generally superior. The reason is that the BSDF parameterizations might squeeze unoccluded, off-specular directions into vanishingly small regions of the primary sample space.

[2] One can consider their technique to generate *path bundles* and in this sense their target distribution is optimal for the constructed bundles, relative to the overall bundle contribution, but not for the individual paths.

[3] In practice instead of sampling f, they are sampling a version downscaled locally according to the efficiency of p_i

This definition leads to some interesting properties. First and foremost, we have the following simplifications:

$$\pi_i(u) = \frac{f(T(u))}{\sum_{j \in \mathcal{F}_k} p_j(T(u))} \tag{17}$$

Second, in each primary sample space the target distribution depends only on the path $\mathbf{x} = T(u)$, but not on the particular choice of technique i used to generate it. In other words, if $u^i \in U_i$ and $u^j \in U_j$ map to the same path $\mathbf{x} = T_i(u^i) = T_j(u^j)$, we have:

$$\pi_i(u^i) = \pi_j(u^j) \tag{18}$$

In particular, the target distribution depends only on how well the *sum*[4] of the individual pdfs p_i approximate f. This is an interesting result, as we will see later on.

Third, notice that if all bidirectional techniques are included in $\mathcal{F}_k$, the target distribution does not contain any of the weak singularities induced by the geometric terms. This is the case because each pdf includes all but one of the geometric terms: thus their sum will contain all of them, and counterbalance those in the numerator of (18). In particular, this means there will be no singular concentration of paths near geometric corners.[5] Notice that this would have not been the case if we simply adopted $\pi = f/p_i$, omitting the multiple importance sampling weight.

2.3 Auxiliary distributions

Šik et al [2016] proposed using an auxiliary distribution in conjunction with replica exchange [Swendsen and Wang 1986] to help the primary MLT chain escape from local maxima. The auxiliary distribution is designed to be easier to sample, and hence favor exploration. Given they were working in the context of the original PSSMLT formulation where all connections are performed, they proposed using an auxiliary distribution with a target defined as 1 if any of the paths formed provides a non-zero contribution, and 0 otherwise.

A similar but even easier objective can be achieved by simply dropping all connection terms except for visibility, i.e. the only terms which are not sampled by our local path sampling technique, giving:

$$\pi'(u) = V(y_{s-1} \leftrightarrow z_{t-1}) \tag{19}$$

which in path space becomes:

$$\bar{\pi}'(x) = V(x_{s-1} \leftrightarrow x_s) p_i(x) \tag{20}$$

Notice that due to our use of primary sample space mutations, this function is very easy to sample, as our base sampling technique already generates samples distributed according to p_i. Importantly, we might not even need Metropolis at all, as we could simply use our path generation technique as an independence sampler, akin to the *large steps* in the original work of Kelemen et al [2002]. However, using Metropolis with local perturbations might still help in regions with difficult visibility.

2.4 Handling color

In the above we treated f as a scalar, though in practice it is actually a color represented either in RGB or with some other spectral sampling. While handling spectral rendering in all generality can require custom techniques [Wilkie et al. 2014] and is beyond the scope of this paper, for RGB (and even in many cases of spectral transport) it is sufficient to use the maximum of the components $f^* = \max_i\{(f)_i\}$ when constructing the target distribution, and weighting the resulting color samples accordingly before final image accumulation.

3 Charted Metropolis-Hastings

Before introducing our light transport algorithm, we introduce a novel family of general Markov chain Monte Carlo algorithms inspired by the primary sample space MLT formulation we just described. The idea is that we want to allow *jumping* between different primary sample spaces.

Suppose in all generality that we have an arbitrary *target space* (Ω, μ), a function $f : \Omega \to \mathbb{R}$ we are interested in sampling, and a parametric family of *sampling charts* $\mathcal{F} = (U_i, T_i, R_i)_{i=0,\ldots,n-1}$ such that:

U_i is a measured *primary sample space*;

T_i is a *forward map*, $T_i : U_i \to \Omega$ with density $p_i : \Omega \to \mathbb{R}$;

R_i is a *reverse map*, $R_i : \Omega \to U_i$ with density $r_i : U_i \to \mathbb{R}$;

Now, consider again the distributions defined by:

$$\pi_i(u) = \frac{f(T_i(u))}{\sum_i p_i(T_i(u))} \tag{21}$$

The idea is that we could use the reverse maps R_i, which can be interpreted as inverse sampling functions, to perform the desired jumps between primary sample spaces, e.g performing swaps in the context of a replica exchange framework where we run n chains, each sampled according to a different π_i. We now show how to achieve it.

Given two states, u_1^i, generated by the i-chain, and u_2^j, generated by the j-chain, consider their target space mappings:

$$x_1 := T_i(u_1^i)$$
$$x_2 := T_j(u_2^j)$$

and their *reverse* mappings:

$$u_1^j := R_j(x_1)$$
$$u_2^i := R_i(x_2)$$

if we wanted to perform a swap, preserving detailed balance between the chains requires accepting the swap with probability:

$$A = \min\left(1, \frac{\pi_i(u_2^i)\pi_j(u_1^j)r_i(u_1^i)r_j(u_2^j)}{\pi_i(u_1^i)\pi_j(u_2^j)r_i(u_2^i)r_j(u_1^j)}\right) \tag{22}$$

This can be proven by looking at the two chains as an ensemble in the space $U_i \times U_j$, with target distribution $\pi_i \cdot \pi_j$. Equation (22) is then obtained from equation (9) following the usual Metropolis-Hastings rule described in section 2.1, viewing (u_1^i, u_2^j) as the current state and (u_2^i, u_1^j) as the proposal.

[4] Equivalently, their average, since π is here defined up to a constant.

[5] The only sources of singularities can be the Diracs in unsampled specular BSDFs in SDS paths (not containing any DD edge).

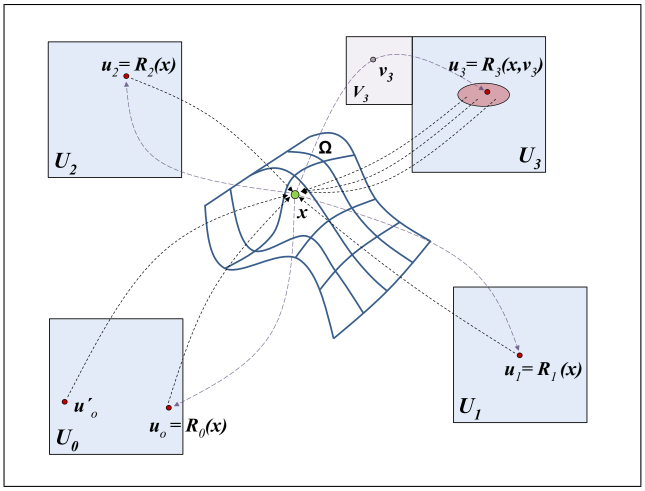


Figure 2: *Charted Metropolis-Hastings allows performing coordinate changes between the target space Ω and its sampling charts. When multiple points of a given sampling domain map to a single point in Ω, it's sufficient for the right inversion mappings to return one of them (as for the case of u_0), or return one picked at random inside the basin (as for the case of u_3) with the help of an additional sampling domain (V_3, light violet box).*

In the previous section we saw that our target distributions π_i assume the same value on the same points of Ω, independently of the underlying technique i used to generate it. Now suppose R_i was a right inverse of T_i, i.e:

$$T_i(R_i(x)) = x \quad \forall x \in \Omega; \tag{23}$$

this would guarantee that, if $u^j = R_j(T_i(u^i))$, we would have:

$$\pi_j(u^j) = \pi_i(u^i). \tag{24}$$

This property is essentially stating that our target distribution is invariant under a change of charts of the target space.

In this case, equation (22) simplifies to:

$$A = \min\left(1, \frac{r_i(u_1^i) r_j(u_2^j)}{r_i(u_2^i) r_j(u_1^j)}\right) \tag{25}$$

without requiring any evaluation of the target distributions. Notice that we don't require the transformations T_i to be fully invertible: if the fiber of $T_i^{\leftarrow}(x)$ contains several points, it's sufficient that R_i returns one of them. In fact, this can be made even more general by *randomizing* the selection of the point in the fiber.

We do so by extending the domains in which the functions R_i operate. I.e. we consider the family $\mathcal{F} = (U_i, V_i, T_i, R_i)_{i=0,\ldots n-1}$ where U_i and T_i are defined as before, but:

V_i is a measured *reverse sampling space*, and

R_i is an *extended right-inversion map*, $R_i : \Omega \times V_i \to U_i$, such that:

$$T_i(R_i(x, v)) = x \quad \forall x \in \Omega \quad \text{and} \quad \forall v \in V_i.$$

With these definitions, we can draw two uniform random variables $v_1 \in V_i$ and $v_2 \in V_j$, and replace the reverse mappings u_1^j and u_2^i with:

$$u_1^j := R_j(x_1, v_1)$$
$$u_2^i := R_i(x_2, v_2)$$

which can now be tested for acceptance with the same acceptance ratio:

$$A = \min\left(1, \frac{r_i(u_1^i) r_j(u_2^j)}{r_i(u_2^i) r_j(u_1^j)}\right).$$

This construction is depicted in Figure 2, where:

a) the chart U_0 contains two points, u_0 and u_0', that map to the same point $\mathbf{x} \in \Omega$, but $R_0(\mathbf{x})$ selects just one of them, in this case u_0;

b) the chart U_3 contains an entire basin that maps to x, but its points are identified by means of points of the reverse sampling domain V_3.

A similar mathematical framework can be used in the context of serial (or simulated) tempering [Marinari and Parisi 1992]. In this context, one could run a single chain $u^i = (u, i)$ in an extended state space $U \times \mathcal{F}$, where i denotes the technique used to map the chain to target space. Drawing a uniform random variable $v \in V_i$ and swapping from i to j through the transformation:

$$u^j = R_j(u^i, v)$$

would then require accepting the swap with probability:

$$\min\left(1, \frac{r_i(u^i)}{r_j(u^j)}\right) \tag{26}$$

and rejecting it otherwise. Once again, no evaluation of the target distributions is required. We call both this and the above mutations *chart swaps* or *coordinate changes.*

Notice that if there is a way to craft mutations in the target space itself, it is always possible to add the identity chart to $\mathcal{F}$:

$U_n = \Omega$, $V_n = \emptyset$

$T_n(x) = R_n(x) = x$;

care must only be taken in adding the probability $p_n = 1$ to the denominator of all the distributions π_i in equation (21).

Finally, we consider a third type of mutations, *inverse primary space perturbations*, which can be in a sense considered the dual of the above. Suppose we are now running a chain in the target space Ω, distributed according to $\pi(\mathbf{x})$. We can then use inversion to momentarily parameterize the target space through a given technique i and take a detour or *move down* from Ω to U_i to perform a symmetric primary sample space perturbation there, before finally getting back to Ω. With this scheme, given a state $\mathbf{x}$ and a uniform random variable $v \in V_i$, applying the transformation R_i to obtain $u = R_i(\mathbf{x}, v)$ and the perturbation kernel K to obtain the proposal $u' = K(u)$ and $\mathbf{y} = T_i(u')$, would result in the following acceptance ratio:

$$A(\mathbf{y}|\mathbf{x}) = \min\left(1, \frac{\pi(\mathbf{y}) K(u|u') r_i(u')}{\pi(\mathbf{x}) K(u'|u) r_i(u)}\right) \tag{27}$$

which ends up being the standard primary sample space formula if $r_i(u) = 1/p_i(T(u))$ and K is symmetric:

$$A(\mathbf{y}|\mathbf{x}) = \min\left(1, \frac{\pi(\mathbf{y})}{p_i(\mathbf{y})} \cdot \frac{p_i(\mathbf{x})}{\pi(\mathbf{x})}\right). \tag{28}$$

The only requirement would be to have R_i respect exactly the reciprocal of the density of T_i - although we stress that equation (27) is valid even if this condition is not met.

We call this family of MCMC algorithms that jump between charts of the target space *charted Metropolis-Hastings*, or CMH.

4 Charted Metropolis Light Transport

It should now be clear how the above algorithms can be applied to light transport simulation. If we consider the framework for primary sample space MLT outlined in section 2.2, it is sufficient to add functions for *path sampling inversion* to be able to apply our new charted Metropolis-Hastings replica exchange or serial tempering mutations in conjunction with the standard set of primary sample space perturbations. The advantage of these mutations is that they will allow to more easily escape from local maxima when the current sampling technique is not locally the best fit for f. The mutations are relatively cheap, as they don't require any expensive evaluations of the target distribution.

Moreover, and very importantly, the algorithm is made practical by not requiring the path sampling functions T_i to be classically invertible. In the context of light transport simulation this property is crucial, as BSDF sampling is seldom invertible: in fact, with layered materials often a random decision is taken to decide which layer to sample, but the resulting output directions could be equally sampled (with different probabilities) by more than one layer. Our framework requires to return just one of them, but it also allows selecting which one at random with a proper probability. All is needed is the ability to compute the density of the resulting transformation. This construction is illustrated in Figure 3, where the chart $U_{3,2}$ contains two points, $u_{3,2}$ and $u'_{3,2}$, that map to the same path $\mathbf{x}$, but $R_{3,2}(\mathbf{x})$ selects just one of them, in this case $u'_{3,2}$.

Further on, by adding the identity target space chart, we can also add the original path space mutations proposed by Veach and Guibas [1997], potentially coupled with the new inverse primary space perturbations.

We call the family of such algorithms *charted Metropolis light transport*, or CMLT.

4.1 Connection to path space MLT

The new algorithms can be considered as a bridge between primary sample space MLT and the original path space MLT proposed by Veach and Guibas [1997]. In fact, one of the advantages of the original formulation over Kelemen's variant [2002] was its ability to *break the path in the middle* and resample the given path segment with any arbitrary bidirectional technique. This ability was entirely lost in primary sample space, as the bidirectional sampling technique was implicitly determined by the sample coordinates (or needed to be chosen ahead of time in the version we outlined in section 2.2). While Multiplexed Metropolis Light Transport (MMLT) [Hachisuka et al. 2014] added the ability to change technique over time, as the coordinates u were kept fixed, such a scheme was leading to swap proposals that sample unrelated paths: in fact, two techniques i and j map the same coordinates u to different paths $T_i(u) \neq T_j(u)$ that share only a portion of their prefixes (in other words, the two resulting paths are *spuriously* correlated by the algorithm, whereas in fact there is no reason for them to be). Our coordinate changes, in contrast, *preserve* the path while changing its parameterization, thus allowing to simply perturb it later on with a different bidirectional sampler.

Adding the identity path space chart and inverse primary space perturbations makes the connection even tighter, allowing to smoothly integrate the original bidirectional mutations and perturbations with an entirely new set of primary sample space perturbations.

Notice that while inverse primary space perturbations could also be applied to a single path space chain, the advantage of also incorporating primary space chains in a replica exchange or serial tempering context is that the target distributions (defined by equation 21) become generally smoother due to the implicit use of the multiple importance sampling weight, raising the acceptance rate.

4.2 Alternative parameterizations

While the original primary sample space Metropolis used path space parameterizations based on plain BSDF sampling, it is also possible to use other parameterizations that can provide further advantages: for example the half vector space parameterizations that have been recently explored [Kaplanyan et al. 2014; Hanika et al. 2015].

4.3 Density Estimation

So far, we have concentrated on standard bidirectional path tracing with vertex connections. However, all the above extends naturally to density estimation methods, using the framework outlined in [Hachisuka et al. 2012]. The only major difference is the computation of the subpath probabilities.

However, in order to combine density estimation into our new framework, we here suggest a novel approach. Instead of using density estimation as an additional technique, applying multiple importance sampling to combine it into a unique estimator, we use it only to craft additional proposals. In other words, we use density estimation as another independence sampler. Suppose again at some point in time our chain is in the state u^i. We can then try to build a candidate path through density estimation with the $(s+1,t)$-technique and, if the resulting path has non-zero contribution, we can drop one light vertex (and the corresponding primary sample space coordinates) and consider it as a new proposal u^i_{de}. Notice that in doing so, we have to adjust the acceptance ratio for the actual proposal distribution. For clarity, we will now omit the superscripts s, t, and obtain:

$$A(u_{de}|u) = \min\left(1, \frac{\pi(u_{de})p_{de}(T(u))}{\pi(u)p_{de}(T(u_{de}))}\right) \tag{29}$$

where $p_{de}(x)$ is the probability of sampling the path x by density estimation (which can be approximated at the cost of some bias as described in [Hachisuka et al. 2012] or estimated unbiasedly as in [Qin et al. 2015]).

If we want to further raise the acceptance rate, we can also mix this proposal scheme with an independence sampler

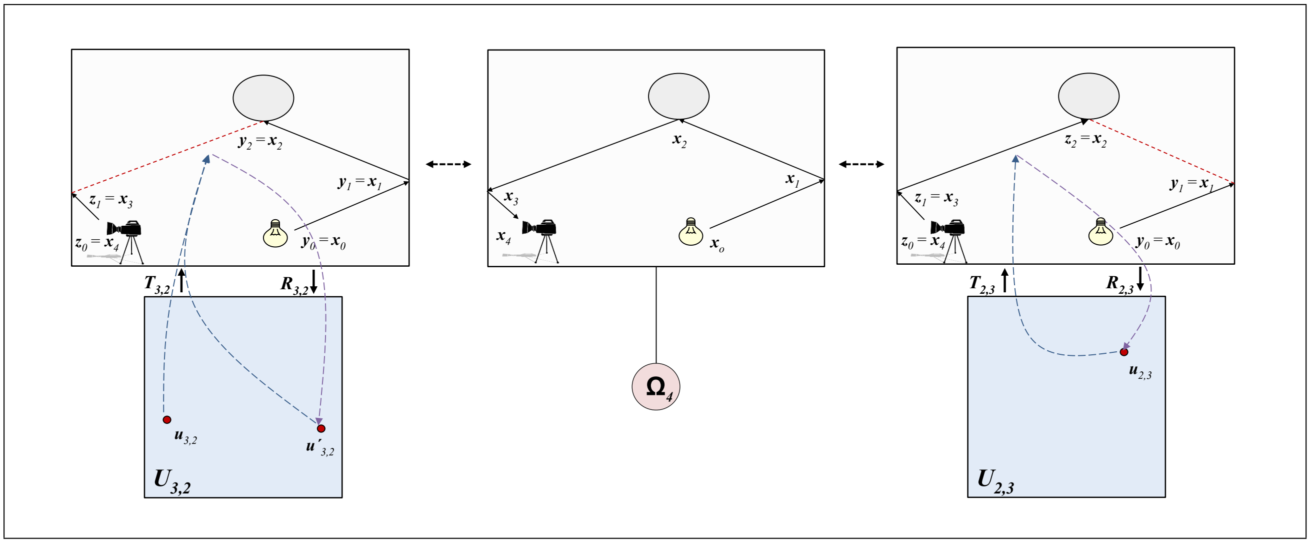


Figure 3: *A visualization of the path space charts, where multiple primary space coordinates may map to the same path.*

based on bidirectional connections and combine the two, calculating the total expected probability to make both more robust:

$$A(u'|u) = \min\left(1, \frac{\pi(u')(p_{de}(T(u)) + p_{bc}(T(u)))}{\pi(u)(p_{de}(T(u')) + p_{bc}(T(u')))}\right). \quad (30)$$

Notice that this formula is now agnostic of how the samples were generated in the first place, i.e. whether the candidate u' was proposed by density estimation or bidirectional connections: this is a positive side-effect of using expectations.[6]

4.4 Designing a complete algorithm

So far we have only constructed a theoretical background to build novel algorithms, but we didn't prescribe practical recipes. The way we combine all the above techniques into an actual algorithm is described here.

First of all, we start by estimating the total image brightness with a simplified version of bidirectional path tracing. The algorithm first traces N_{init} light subpaths in parallel and stores all generated light vertices. Then it proceeds tracing N_{init} eye subpaths, and connects each eye vertex to a single light vertex chosen at random among the ones we previously stored. At the same time, the EDF at each eye vertex is considered, forming pure path tracing estimators with light subpaths with zero vertices. All evaluated connections (both implicit and explicit) with non-zero contribution (which represent entire paths, each with a different number of light and eye vertices s and t) are stored in an unordered list.

Second, in order to remove startup bias, we resample a population of N seed paths for a corresponding amount of chains. In order to do this, we build the cumulative distribution over the scalar contributions of the previously stored paths, and resample N of them randomly.

Notice that the N seed paths will be distributed according to their contribution to the image. Particularly, the number of paths sampled with technique i will be proportional to the overall contribution of that technique, and similarly for path length. At this point, though not crucial for the algorithm, we sort the seeds by path length k so as to improve execution coherence in the next stages. In practice, sorting divides the N seeds into groups of N_k paths each, such that $\sum_k N_k = N$.

Finally, we run the N Markov chains in parallel using both classic primary sample space perturbations and the novel simulated tempering or replica exchange mutations described in sections 3 and 4. As the new mutations have a low cost compared to performing actual perturbations, they can be mixed in rather frequently.

5 Implementation

We implemented our algorithm, together with MMLT and bidirectional path tracing (BPT) in CUDA C++, exposing massive parallelism in every single stage, including ray tracing, shading, cdf construction (prefix sum), resampling and sorting (radix sort). The basic bidirectional path tracing algorithm is constructed as a pipeline of kernels (also known as wavefront tracing), and relies on the OptiX Prime library for ray tracing. For the sake of our experiments, we ran all tests on an NVIDIA Maxwell Titan X GPU.

Both our CMLT and MMLT implementations run several thousand chains in parallel, using the seeding algorithm described in section 4.4. Besides being strictly necessary to scale to massively parallel hardware, we found this to produce some additional image stratification. The CMLT implementation is based on the serial tempering formulation.

Our framework employs a layered material system that combines a diffuse BSDF (Lambertian) and rough glossy reflec-

[6] while this looks similar to multiple importance sampling, it is not quite the same: multiple importance sampling is a more general technique used to combine estimators, whereas here we are just interested in computing an expected probability density, using so called state-independent mixing [Geyer 2011]. However, multiple importance sampling using the balance heuristic is equivalent to using an estimator based on the average of the probabilities, which is exactly the expected probability we need: hence the reason of the similarity. This approach is the same used in the original MLT to compute the expected probability of bidirectional mutations.

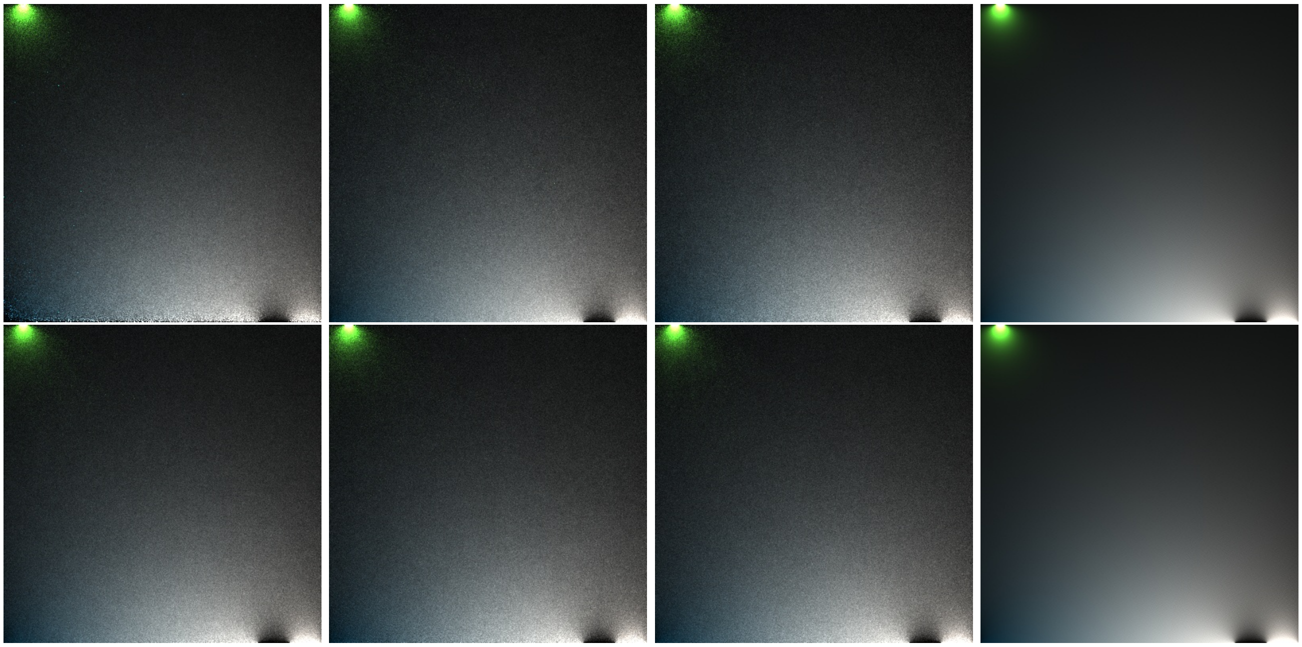


Figure 4: *A simple test scene featuring direct lighting from two equally sized area lights: the first, with spatially varying emission properties, partially blocked by a thin black strip; the second only reachable through a tiny hole. From left to right, first row: PSS-MLT-1, PSS-MLT-2, PSS-MLT-AVG, reference image. Second row: PSS-MLT-MIX, CMLT-IPSM, CMLT, reference image.*

tion and transmission BSDFs (GGX) using a Fresnel weighting. Sampling of the glossy component is implemented using the distribution of visible normals [Heitz and D'Eon 2014], and selection between the diffuse and glossy components is performed based on Fresnel weights. Clearly, this path sampling scheme is not invertible, as both the diffuse and glossy components can map different primary sample space values to the same outgoing directions. Hence, we used the machinery described in section 3.3 to enable randomized inversion.

Our implementation of CMLT doesn't include yet our new density estimation based proposals, which we leave as future work.

5.1 Chart swaps and path inversion

Given a bidirectional path generated by the technique (s, t) using coordinates u, in order to perform a chart swap we propose a new pair (s', t') distributed according to the total energy of the techniques (i.e. the normalization constants of the target distributions). After the candidate is sampled, path inversion needs to be performed using the transformation $u' = R_{s',t'}(T_{s,t}(u))$. This transformation can be widely optimized noticing that there are only two cases:

$s' > s$: in this case it is only necessary to invert the coordinates of the light subpath vertices $\{y_s, ..., y_{s'-1}\}$.

$t' > t$: in this case it is only necessary to invert the coordinates of the eye subpath vertices $\{z_t, ..., z_{t'-1}\}$.

Computing the inverse pdf $r_{s,t}$ can be optimized analogously.

6 Results

We performed two sets of tests. The first is aimed at testing the many possible algorithmic variations of CMLT on a simplified light transport problem. The second, using full light transport simulation, compares a single CMLT variant against MMLT, which could be currently considered state-of-the-art in primary sample space MLT.

6.1 Simplified light transport tests

This test consists of rendering an orthographic projection of the XY plane directly lit by two area light sources. The first light is a unit square on the plane $Y = 0$, with a spatially varying emission distribution function changing color and increasing in intensity along the X axis. The light source is partially blocked by a thin black vertical strip near its area of strongest emission. The second light is another unit square on the plane $Y = 1$, with uniform green emission properties. This light is completely blocked except for a tiny hole.

In this case, our path space consists of two three-dimensional points: the first on the ground plane, the second on the light source. As charts, we used two different parameterizations:

1. generating a point uniformly on the visible portion of the ground plane and a point on the light sources distributed according to their spatial emission kernels (corresponding to path tracing with next-event estimation, i.e. the bidirectional path tracing technique $(s, t) = (1, 1)$);
2. generating a point uniformly on the visible portion of the ground plane, sampling a cosine distributed direction, and intersecting the resulting ray with the scene geometry to obtain the second point (corresponding to

Algorithm	n = 16 · 10e6	n = 128 · 10e6
`PSS-MLT-1`	8.287 · 10e-2	3.212 · 10e-2
`PSS-MLT-2`	4.073 · 10e-2	1.488 · 10e-2
`PSS-MLT-AVG`	4.106 · 10e-2	1.587 · 10e-2
`PSS-MLT-MIX`	3.663 · 10e-2	1.405 · 10e-2
`CMLT-IPSM`	3.924 · 10e-2	1.467 · 10e-2
`CMLT`	**3.502 · 10e-2**	**1.374 · 10e-2**

Table 1: *Root mean square error of the images computed by the various algorithms we tested in figure 4.*

pure path tracing, i.e. the bidirectional path tracing technique $(s, t) = (0, 2)$).

Both charts have a four dimensional domain, and in both cases we used exact inverses of the sampling functions.

We tested six different MCMC algorithms:

PSS-MLT-1: a single PSS-MLT chain using the first parameterization;

PSS-MLT-2: a single PSS-MLT chain using the second parameterization;

PSS-MLT-AVG: two PSS-MLT chains using both the first and the second parameterization, both distributed according to eq. (13), where the accumulated image samples are weighted (i.e. averaged) through multiple importance sampling with the balance heuristic;

PSS-MLT-MIX: two PSS-MLT chains using both the first and the second parameterization, each distributed according to eq. (15);

CMLT-IPSM: a single CMLT chain in path space alternating inverse primary space mutations using the first and the second parameterizations;

CMLT: two CMLT chains using both the first and the second parameterization as charts, coupled with replica-exchange swaps performed every four iterations;

Results are shown in Figure 3, while their root mean square error (RMSE) is reported in Table 1. All images except for the reference were produced using the same total amount of samples $n = 16 \cdot 10^6$: PSS-MLT-1, PSS-MLT-2 and CMLT-IPSM running a single chain of length n, whereas PSS-MLT-AVG, PSS-MLT-MIX and CMLT running two chains of length $n/2$. In table 1 we further report RMSE values for $n = 128 \cdot 10^6$. The reference image has been generated by plain Monte Carlo sampling.

As can be noticed, our PSS-MLT-MIX formulation using the distributions defined by equation (15) is superior to simply averaging two PSS-MLT chains using multiple importance sampling (PSS-MLT-AVG), which is in fact worse than PSS-MLT using a single chain according to the second distribution (PSS-MLT-2).

CMLT-IPSM produces results that are just slightly worse than PSS-MLT-MIX, but still superior to all other PSS-MLT variants. The reason why CMLT-IPSM is inferior to PSS-MLT-MIX is that while the target distribution for CMLT-IPSM is proportional to f, the target distributions of the chains in PSS-MLT-MIX are smoother due to the embedded multiple importance sampling weights, and contain no singularities. Finally, CMLT produces the best results among all algorithms.

6.2 Full light transport tests

For these tests we compared the CMLT implementation described in section 5 against our own implementation of MMLT. We provide four test scenes representative of different transport phenomena:

Escher's Box: an M.C. Escher themed adaptation of the Cornell Box, featuring both diffuse and glossy surfaces. This scene contains two light sources: the large back wall, with a variable Lambertian emission distribution displaying a famous painting by the artist, and a smaller area light on the ceiling. The smaller light is partially blocked by a rough glossy reflector, which causes a blurry caustic on the partially glossy ceiling.

Escher's Glossy Box: a variation of the above scene in which all surfaces are glossy (with no diffuse component), with variable roughness (with GGX exponents ranging between 5 and 100). Notice that this scene contains a variety of caustics of all frequencies (in a sense, all lighting is due to caustics). This scene contains an additional large off-screen area light mimicking the sky.

Escher's Textured Glossy Box: same as above, with textures that modulate the amount of reflection.

Escher's Box with Wall Ajar: another variation of the above scene mimicking Eric Veach's famous scene *the door ajar*. Most of the lighting in the scene comes from an area light hidden beneath the sliding back wall. Again, the area light has variable surface emission properties. The ceiling area light source is also considerably smaller, and the surfaces are roughly half diffuse half glossy. The short red box also features rough transmission.

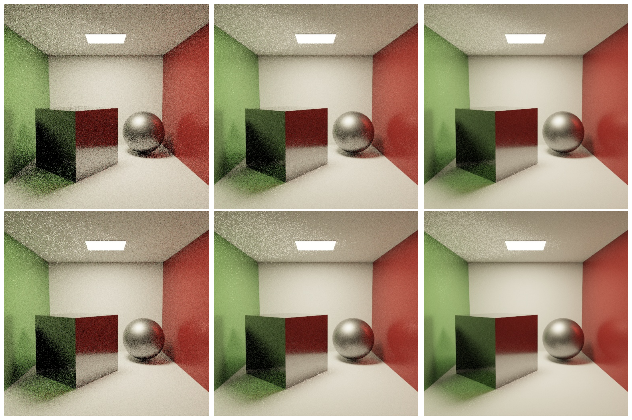
It is important to note that while the scenes look superficially similar, they stress entirely different transport phenomena. Moreover, all of them are very complex to render, requiring between $16 \cdot 10^3$ and $128 \cdot 10^3$ samples per pixel (spp) for bidirectional path tracing to converge.

Figure 6 shows equal-time comparisons of MMLT and CMLT on all scenes. Except for the last row, both the MMLT and CMLT renders were generated using 64 spp, taking roughly the same computation time, whereas the reference images have been rendered with bidirectional path tracing using 32K spp. The images in the last row used 256 spp for MMLT and CMLT, and 128K spp for the reference image.

CMLT produces considerably less noise on all test scenes. Notice that while the caustic on the ceiling in the last scene is hardly visible in both the MMLT and CMLT renders, this is only due to the relatively low number of samples: both do converge to the correct result after a sufficiently long time (with CMLT, the caustic starts to be fully visible at 512 spp).

Figure 7 shows the convergence of MMLT and CMLT on the hardest of the four scenes. Notice how MMLT needs almost twice as many samples as CMLT to get approximately the same RMSE.

Figure 8 shows a similar graph comparing also to PSSMLT. Since each PSSMLT sample requires both more shadow rays



(a) *32 spp* **(b)** *128 spp* **(c)** *512 spp*

Figure 5: *Parallel CMLT convergence using respectively 32K (top row) and 256K chains (bottom row). From left to right: 32, 128 and 512 samples per pixel.*

and BSDF evaluations, in our implementations PSSMLT can perform roughly one half the mutations as CMLT in the same time.

Finally, Figure 5 shows the effect of varying the number of chains run in parallel, trading it against chain length to keep the total number of samples fixed. The images in the top row are obtained running 32K chains in parallel, whereas the ones in the bottom row are obtained using 256K chains. It can be seen that using more, shorter chains generally improves stratification. The exception is the caustic on the ceiling that benefits from the higher adaptation of the longer chains. In practice, this parameter trades exploration for stratification.

In all cases, for CMLT we used one chart swap proposal every 32 mutations, resulting in negligible overhead. On our system, the 1k spp CMLT and MMLT images take roughly 2 minutes to render at a resolution of 1600 x 900.

7 Discussion

We proposed a novel family of MCMC algorithms that use sampling charts to extend the sampling domain and allow better exploration. We applied the new scheme to propose a new type of light transport simulation algorithms that bridge primary sample space and path space MLT.

We also showed that the new algorithms arising from this framework require to implement only a new set of relatively cheap mutations that can be constructed using simple, stochastic right inverses of the path sampling functions: particularly, the fact our framework requires only such type of probabilistic inversion is what makes the algorithm practical, as classical BSDF inversion with layered material models is generally impossible. We believe this to be a major strength of our work.

We implemented both the old and new methods exposing massive parallelism at all levels, and showed how increasing the number of chains that run in parallel can increase stratification.

Finally, we suggested a novel, simpler method to integrate path density estimation into MCMC light transport algorithms as a mechanism to craft independent proposals.

7.1 Future work

There are multiple venues in which this work could be extended. The first is testing all possible variants of our new algorithmic family more thoroughly. In such context, it will be particularly interesting to test the combination with the original path space MLT mutations, which might provide some advantages in regions with complex visibility. Similarly, it would be interesting to test the new technique for including path density estimation as an independence sampler.

Another potential venue is considering *dimension jumps* to switch between the charts underlying different path spaces Ω_k and $\Omega_{k'}$. This could be achieved using the *Metropolis-Hastings-Green with Jacobians* algorithm as described in [Geyer 2011].

Finally, it would be interesting to integrate half vector space light transport [Kaplanyan et al. 2014; Hanika et al. 2015] as yet another path space chart.

References

Anonymous. 2016. Charted Metropolis Light Transport. *arXiv:1612.05395 - non-peer-reviewed prepublication by the authors* (Dec.).

Geyer, C. J. 2011. Introduction to Markov Chain Monte Carlo. In *Handbook of Markov Chain Monte Carlo*. Chapman and Hall CRC, ch. 1, 3–47.

Hachisuka, T., and Jensen, H. W. 2011. Robust adaptive photon tracing using photon path visibility. *ACM Trans. Graph. 30*, 5 (Oct.), 114:1–114:11.

Hachisuka, T., Pantaleoni, J., and Jensen, H. W. 2012. A path space extension for robust light transport simulation. *ACM Trans. Graph. 31*, 6 (Nov.), 191:1–191:10.

Hachisuka, T., Kaplanyan, A. S., and Dachsbacher, C. 2014. Multiplexed Metropolis light transport. *ACM Trans. Graph. 33*, 4 (July), 100:1–100:10.

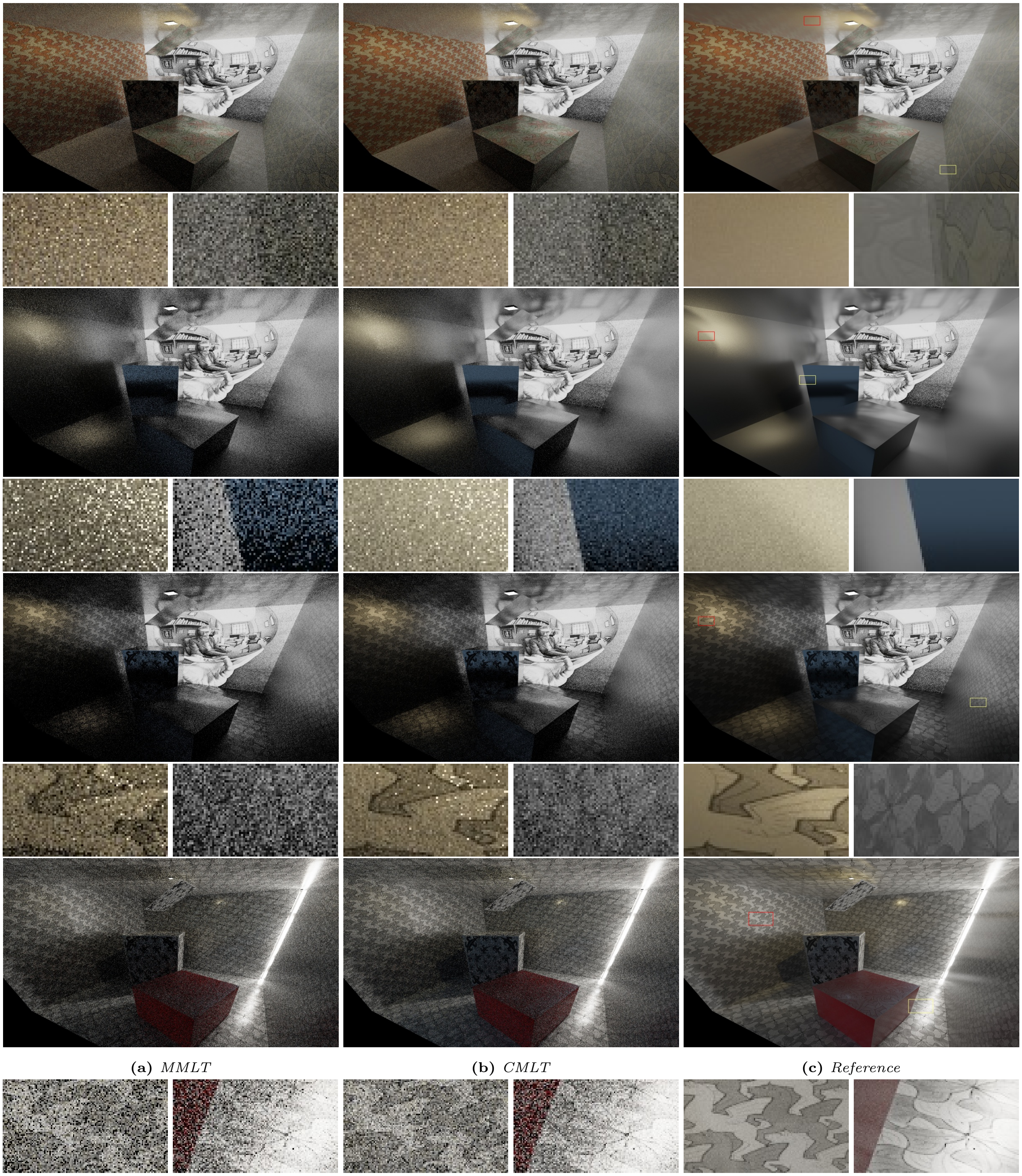
Hanika, J., Kaplanyan, A., and Dachsbacher, C. 2015. Improved half vector space light transport. *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering) 34*, 4 (June), 65–74.

Heitz, E., and D'Eon, E. 2014. Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals. *Computer Graphics Forum 33*, 4 (July), 103–112.

Kaplanyan, A. S., Hanika, J., and Dachsbacher, C. 2014. The natural-constraint representation of the path space for efficient light transport simulation. *ACM Transactions on Graphics (Proc. SIGGRAPH) 33*, 4.

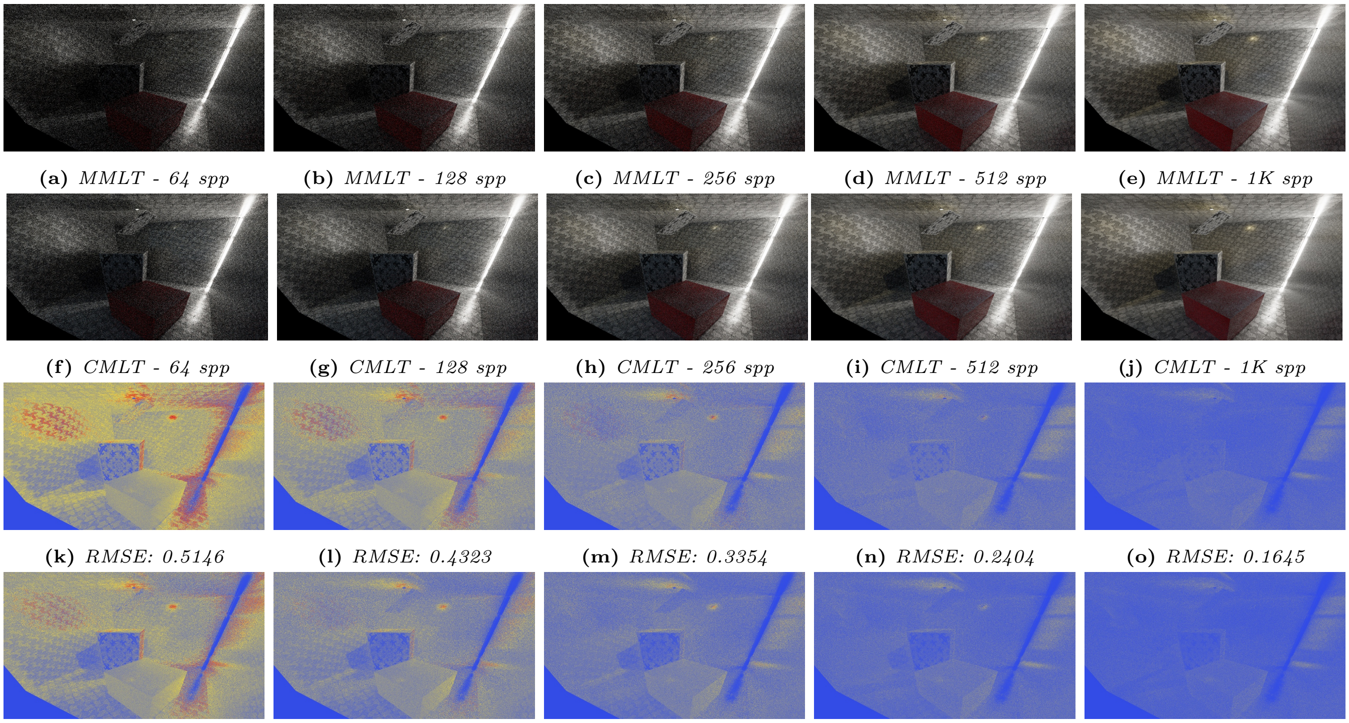
Kelemen, C., Szirmay-Kalos, L., Antal, G., and Csonka, F. 2002. A simple and robust mutation strategy for the Metropolis light transport algorithm. In *Computer Graphics Forum*, 531–540.

Marinari, E., and Parisi, G. 1992. Simulated tempering: A new Monte Carlo scheme. *Europhysics Letters*, 19, 451–458.



(a) *MMLT* **(b)** *CMLT* **(c)** *Reference*

Figure 6: *Equal-time comparisons of our CMLT and MMLT on four scenes testing different transport phenomena.*



(a) *MMLT - 64 spp* **(b)** *MMLT - 128 spp* **(c)** *MMLT - 256 spp* **(d)** *MMLT - 512 spp* **(e)** *MMLT - 1K spp*

(f) *CMLT - 64 spp* **(g)** *CMLT - 128 spp* **(h)** *CMLT - 256 spp* **(i)** *CMLT - 512 spp* **(j)** *CMLT - 1K spp*

(k) *RMSE: 0.5146* **(l)** *RMSE: 0.4323* **(m)** *RMSE: 0.3354* **(n)** *RMSE: 0.2404* **(o)** *RMSE: 0.1645*

(p) *RMSE: 0.4368* **(q)** *RMSE: 0.3492* **(r)** *RMSE: 0.2651* **(s)** *RMSE: 0.1934* **(t)** *RMSE: 0.1375*

Figure 7: *RMSE comparison of CMLT (bottom) and MMLT (top) at 64, 128, 256, 512 and 1K spp.*

Qin, H., Sun, X., Hou, Q., Guo, B., and Zhou, K. 2015. Unbiased photon gathering for light transport simulation. *ACM Trans. Graph. 34*, 6 (Oct.), 208:1–208:14.

Swendsen, R. H., and Wang, J.-S. 1986. Replica Monte Carlo simulation of spin-glasses. *Phys. Rev. Lett.*, 57, 2607–2609.

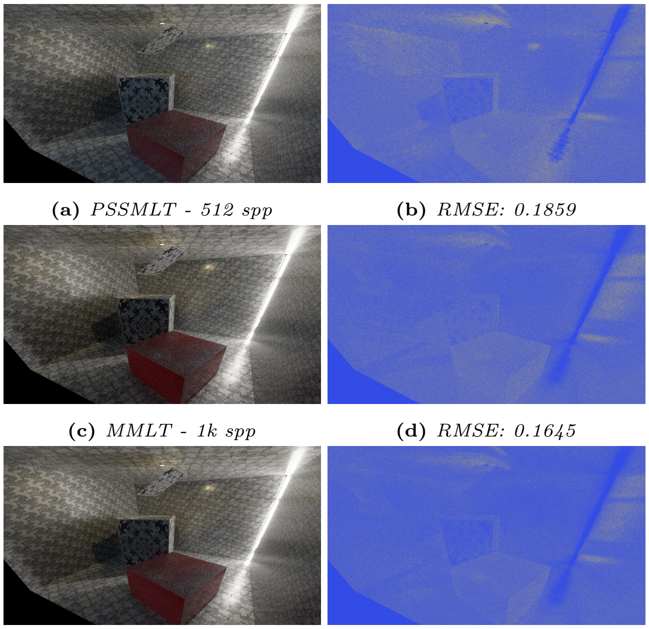
Tierney, L. 1994. Markov chains for exploring posterior distributions. *Annals of Statistics 22*, 1701–1762.

Veach, E., and Guibas, L. J. 1997. Metropolis light transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '97, 65–76.

Veach, E. 1997. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University.

Šik, M., Otsu, H., Hachisuka, T., and Křivánek, J. 2016. Robust light transport simulation via Metropolised bidirectional estimators. *ACM Trans. Graph. 35*, 6 (Nov.), 245:1–245:12.

Wilkie, A., Nawaz, S., Droske, M., Weidlich, A., and Hanika, J. 2014. Hero wavelength spectral sampling. In *Proceedings of the 25th Eurographics Symposium on Rendering*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, EGSR '14, 123–131.



(a) *PSSMLT - 512 spp* **(b)** *RMSE: 0.1859*

(c) *MMLT - 1k spp* **(d)** *RMSE: 0.1645*

(e) *CMLT - 1k spp* **(f)** *RMSE: 0.1375*

Figure 8: *RMSE comparison of PSSMLT (top), MMLT (middle) and CMLT (bottom) at equal computation time.*