

Replica Exchange Light Transport

Shinya Kitaoka, Yoshifumi Kitamura and Fumio Kishino

Graduate School of Information Science and Technology, Osaka University, Osaka, Japan
kitaoka.shinya@ist.osaka-u.ac.jp

Abstract

We solve the light transport problem by introducing a novel unbiased Monte Carlo algorithm called replica exchange light transport, inspired by the replica exchange Monte Carlo method in the fields of computational physics and statistical information processing. The replica exchange Monte Carlo method is a sampling technique whose operation resembles simulated annealing in optimization algorithms using a set of sampling distributions. We apply it to the solution of light transport integration by extending the probability density function of an integrand of the integration to a set of distributions. That set of distributions is composed of combinations of the path densities of different path generation types: uniform distributions in the integral domain, explicit and implicit paths in light (particle/photon) tracing, indirect paths in bidirectional path tracing, explicit and implicit paths in path tracing, and implicit caustics paths seen through specular surfaces including the delta function in path tracing. The replica-exchange light transport algorithm generates a sequence of path samples from each distribution and samples the simultaneous distribution of those distributions as a stationary distribution by using the Markov chain Monte Carlo method. Then the algorithm combines the obtained path samples from each distribution using multiple importance sampling. We compare the images generated with our algorithm to those generated with bidirectional path tracing and Metropolis light transport based on the primary sample space. Our proposing algorithm has better convergence property than bidirectional path tracing and the Metropolis light transport, and it is easy to implement by extending the Metropolis light transport.

Keywords: global illumination, Metropolis light transport, replica exchange Monte Carlo method, multiple importance sampling, population annealing

ACM CCS: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism; I.3.3 [Computer Graphics]: Picture/Image Generation

1. Introduction

Photorealistic images with global illumination are achieved by precisely simulating various light phenomena in a virtual scene by solving the light transport problem, which is an integral of light power. Many algorithms have been proposed to solve the integral. Most are based on numerical integration and ray tracing since the integrand is evaluated by ray tracing. They are statistically consistent and are divided into two classes: biased and unbiased.

The biased algorithms are efficient solutions of photorealistic rendering; however statistical biases always exist in their results. Since the errors of the results come from diverse causes that differ for each algorithm, their effi-

ciency cannot be measured, and there is no guarantee that their results are correct due to unintended biases. Nevertheless these algorithms are commonly used because they rapidly produce impressive results. On the other hand, the advantage of unbiased algorithms is that they provide the expected value of the result equals the true value. This means that images rendered by unbiased algorithms have artefacts caused only by estimation variance and that their efficiency can be quantitatively measured. An unbiased algorithm computes the correct answer, on average. A biased algorithm computes the wrong answer, on average. However, if a biased estimator is also consistent, then the average error can be made arbitrarily small by increasing the sample size.

We argue that unbiased algorithms are essential so as to solve the light transport problem to be robust, because they make it far easier to estimate the error in a calculated result. Unbiased algorithms are often used to generate reference images, against which other rendering algorithms can be compared. Because unbiased methods have strong guarantees about the kinds of errors, they are useful for measuring the artefacts introduced by approximations.

Therefore, we focus on unbiased algorithms and design a novel unbiased global illumination algorithm based on a statistical approach to improve their efficiency.

1.1. Related Work

[Kaj86] formulated photorealistic rendering as the rendering equation and then presented the classical path-tracing algorithm, which is the first unbiased Monte Carlo rendering algorithm. Since then many refinements have been proposed, e.g., [AK90] and [SKAS05]. Often these improvements have been adapted from neutron transport and radiative heat transfer literatures, which have a long history of solving similar transport problems.

The naive path-tracing algorithm is commonly inefficient for general lighting situations, because it traces paths starting only at the eye to evaluate the integrand and has difficulty finding significant light transport paths. [LW93] and [VG94] independently developed Bidirectional Path Tracing (BPT), which generates paths starting not only from the eyes but also from light sources. This algorithm handles indirect illumination better. [LW96] extended the algorithm to handle such participating media as smoke and fog; however, some parts of the path space are not sampled.

Techniques in biased rendering algorithms, on the other hand, cache and interpolate (ir)radiance to sample the entire path space. Allowing errors of path connection using statistically consistent ways, they build paths that transport the radiant flux from light sources to the eyes. Irradiance caching [WRC88], density estimation [SWH*95], photon mapping [Jen96], and ray mapping [HBHS05] all take this approach.

Metropolis Light Transport (MLT) is an advanced unbiased global illumination algorithm introduced in [VG97]. It uses Metropolis-Hastings (MH) sampling, which is a scheme of dynamic Monte Carlo methods. The algorithm utilizes the coherence of the integrand by using path mutation to explore path space in a localized way. Thus, when high contribution paths are found, nearby paths will likely be explored as well. This strategy works well for most scenes. The MLT algorithm is the first application of Markov chain Monte Carlo methods in computer graphics and transport problems.

The original MLT algorithm has been extended in many ways since it was proposed. [PKK00] added new mutation strategies to it for handling participating media, and [KSKGC02] simplified it by defining the problem on an ab-

stract space of random numbers called the primary sample space. It increased the algorithm's mutation acceptance rate and robustness. Other works clarified its statistical properties. [SKDP99] characterized its start-up bias problem, and [APSS01] analysed its variance.

More recent improvements of MLT also exist. [CTE05] presented Energy Redistribution Path Tracing (ERPT) that combines path tracing and MLT in an energy redistribution sampling framework. It can easily stratify and control integration. [LFCD06] improved the ERPT algorithm's efficiency by applying a population Monte Carlo method to energy redistribution. [SIP07] introduces coherent ray tracing for lens subpath mutation in MLT with the multiple-try MH algorithm.

1.2. Extended ensemble Monte Carlo method

Our motivation for applying the extended ensemble Monte Carlo methods [Iba01a] to the light transport problem springs from the fact that there are realistically impossible situations to satisfy the ergodicity of the Markov chain Monte Carlo method. This is known as the slow mixing problem. A multimodal Probability Density Function (PDF) of the integrand causes this problem. Rendering caustics in a mirror/glass or on a pool bottom is especially difficult because specular reflection/refraction is defined by Dirac's delta function, and it derives an integrand with a multimodal PDF.

We propose a novel global illumination algorithm for rendering a general scene called Replica-Exchange Light Transport (RELT), which is an extension of MLT based on the primary sample space [KSKGC02]. The algorithm draws samples based on a family of distributions by the Replica-Exchange Monte Carlo (REMC) method. Therefore, our algorithm is unbiased, handles general lighting configurations, and renders caustics particularly well in a mirror/glass or robustly on a pool bottom. The MLT based on the primary sample space is a robust algorithm; however it does not work well in such situations, because they often lose important contributions along the paths with low probability of occurrence. To our knowledge, this is the first explicit application of the extended (generalized) ensemble Monte Carlo method in computer graphics and transport problems of any kind (e.g. neutron or heat transport problems).

Extended ensemble Monte Carlo is a generic term that indicates a set of algorithms for improving the slow mixing problem. These algorithms, which are included in the extended ensemble Monte Carlo method, are now popular and useful tools in the fields of computational physics and statistical information processing. The REMC method used by our algorithm is one such method.

1.3. Paper organization

The remainder of this paper is organized as follows. In Section 2, we give a high-level overview of RELT and

describe its components in detail. Section 3 summarizes the light transport problem as an integral and solves it by describing the framework of the Monte Carlo estimator. Section 4 presents a RELT algorithm, followed by comparisons among the RELT algorithm, the standard BPT, and MLTs in Section 5. Finally, Section 6 concludes and suggests extensions to improve the RELT algorithm.

2. Overview of the RELT Algorithm

To synthesize an image, we sample points on the primary sample space defined as a unit super-cube. These sampled points are transformed to the paths from the light sources to the eye and then are evaluated by calculating the light flow along the paths.

We define function g on primary sample space such that $\int_{\mathcal{U}} g(\mathbf{u}) d\mathbf{u}$ represents the flux from the light sources to the image plane along a set of paths transformed from points in primary sample space \mathcal{U} . We call g the image contribution function.

The key idea of our algorithm is to sample points in the primary sample space not only with the PDF of integrand g but also with multiple distributions (replicas). To draw samples based on each distribution and to satisfy the detail balance condition for each replica, we use the REMC method that generates a sequence of points for each distribution p_k , $(\mathbf{U}_{k,0}, \mathbf{U}_{k,1}, \dots, \mathbf{U}_{k,N})$ using Metropolis sampling to which the regular exchange is applied. The regular exchange switches sampling states between neighbouring distributions.

As each sample of the k th replica is evaluated, we update the current k th image, which is a two-dimensional array of pixel values. We first find the image location (x, y) corresponding to each sample $\mathbf{U}_{k,i}$, and then a sample is weighted by a multi-sample estimator and a pixel filter function. Next we update the values of those pixels whose filter support contains (x, y) in the k th image. The final image is obtained by adding those K images.

The RELT algorithm is summarized in Algorithm 1. In the following sections, we will describe it in more detail.

3. Light Transport Problem

In this section, we present a short overview of the light transport problem and introduce the light transport integral on primary sample space. Then we describe the Monte Carlo estimator that calculates the integral as an expected value. Since our algorithm is based on this formalism and the notations, they will be used in Section 4.

3.1. Light Transport Integral

The light transport problem is formulated as follows [VG97]:

$$m_{j,\lambda} = \int_{\Omega} f_{j,\lambda}(\bar{x}) d\mu(\bar{x}), \quad (1)$$

where $m_{j,\lambda}$ is the j th pixel value of wavelength λ , Ω is the domain of the path space in a scene, \bar{x} is a path on Ω , $f_{j,\lambda}(\bar{x})$ is a spectral measurement contribution function, and $\mu(\bar{x})$ is a measure on Ω .

Algorithm 1 Pseudocode of RELT table

```

1:   for  $k = 1$  to  $K$  do
2:      $image_k \leftarrow \{ \text{array of zeros} \}$ 
3:      $\mathbf{u}_k \leftarrow \text{InitialSample}(k)$ 
4:   end for
5:   for  $i = 1$  to  $N$  do
6:     for  $k = 1$  to  $K$  do
7:        $\mathbf{v}_k \leftarrow \text{Mutate}(\mathbf{u}_k)$ 
8:        $a \leftarrow \text{AcceptProbability}(\mathbf{v}_k | \mathbf{u}_k)$ 
9:       if  $\text{Random}() < a$  then
10:         $\mathbf{u}_k \leftarrow \mathbf{v}_k$ 
11:      end if
12:       $\text{RecodeSample}(image_k, \mathbf{u}_k)$ 
13:    end for
14:     $k \leftarrow \text{RandomInteger}(1, K - 1)$ 
15:     $e \leftarrow \text{ExchangeProbability}(\mathbf{u}_k, \mathbf{u}_{k+1})$ 
16:    if  $\text{Random}() < e$  then
17:       $\text{Exchange}(\mathbf{u}_k, \mathbf{u}_{k+1})$ 
18:    end if
19:  end for
20:  return  $\text{CompositeImages}(images)$ 

```

To maximize the acceptance probability of Metropolis sampling, we use integral reformulation as an integral on primary sample space \mathcal{U} using a change of variables [KSKGC02]. Point \mathbf{U} on the primary sample space is transformed to path \bar{x} on the path space by $\bar{x} = S(\mathbf{U})$ (see Figure 1), where S builds a path by ray tracing using importance sampling, e.g. BRDF sampling, light source sampling, etc. The integral is given by

$$\begin{aligned} m_{j,\lambda} &= \int_{\mathcal{U}} f_{j,\lambda}(S(\mathbf{U})) \left| \frac{d\mu(S(\mathbf{U}))}{d\mathbf{U}} \right| d\mathbf{U} \\ &= \int_{\mathcal{U}} g_{j,\lambda}(\mathbf{U}) d\mathbf{U}, \end{aligned} \quad (2)$$

where

$$\left| \frac{d\mu(S(\mathbf{U}))}{d\mathbf{U}} \right| = \frac{1}{p_S(\mathbf{U})} \quad (3)$$

is the Jacobian determinant of the transformation,

$$g_{j,\lambda}(\mathbf{U}) = \frac{f_{j,\lambda}(S(\mathbf{U}))}{p_S(\mathbf{U})} \quad (4)$$

is the integrand on primary sample space, and $p_S(\mathbf{U})$ is the probability density of $\bar{x} = S(\mathbf{U})$.

This reformulation explicitly classifies the problem into two portions: point sampler and path builder. The point sampler samples points on the primary sample space with a sampling strategy. The path builder creates a path starting at the

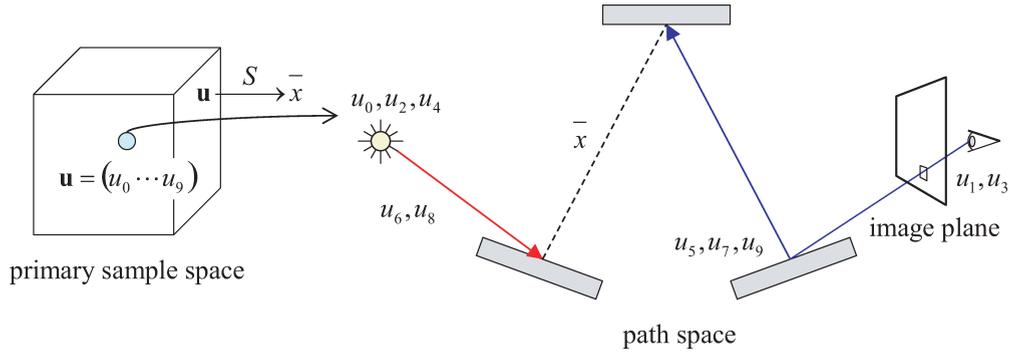


Figure 1: Path \bar{x} built by pure BPT from a point \mathbf{U} in primary sample space: Red arrow shows a light path. Blue arrows show an eye path. u_0 is used for choosing a wavelength of the light. u_2 and u_4 choose a position on the light source. u_6 and u_8 choose an emitting direction from the light source. u_1 and u_3 choose a position on the image plane; the position on the image plane decides a direction of the primary ray in the pinhole camera model. u_5 is used for selecting an elementary BSDF, e.g. BRDF or BTDF. u_7 and u_9 choose a scattered direction by the elementary BSDF.

light sources to the eye from a point on the primary sample space by ray-tracing. For example, standard BPT uses uniform sampling as the sampler and a bidirectional path construction technique as the builder, while MLT based on the primary sample space [KSKGC02] uses MH sampling as the sampler and a (bidirectional) path-tracing technique as the builder.

To design an efficient rendering algorithm, we have to choose a good path builder whose integrand $g(\mathbf{U})$ becomes approximately constant for general scenes. However, such a builder does not exist for unbiased rendering algorithms, e.g. handling caustics in a mirror/glass or on a pool bottom is quite difficult. So we focus on improving the sampler's efficiency.

3.2. Monte Carlo Estimator

The sampler simultaneously estimates all pixel values m_j for efficiency. Observe that each integrand $g_{j,\lambda}(\mathbf{U})$ has the form

$$g_{j,\lambda}(\mathbf{U}) = h_{j,\lambda}(\mathbf{U})g(\mathbf{U}), \quad (5)$$

where $h_{j,\lambda}(\mathbf{U})$ represents the filter function for pixel j and wavelength λ , and $g(\mathbf{U})$ represents all the other factors of $g_{j,\lambda}(\mathbf{U})$. We call g the image contribution function. An image can be computed by sampling N samples \mathbf{U}_i based on distribution $p(\mathbf{U})$ and using identity

$$m_{j,\lambda} = E \left[\frac{1}{N} \sum_{i=1}^N \frac{h_{j,\lambda}(\mathbf{U}_i)g(\mathbf{U}_i)}{p(\mathbf{U}_i)} \right]. \quad (6)$$

The replica-exchange Monte Carlo method generates samples based on multiple distributions, so we use a multiple importance sampling technique [VG95] to compute the expected value. To do this, a multi-sample estimator that uses

K distributions is given by

$$m_{j,\lambda} = E \left[\sum_{k=1}^K \frac{1}{n_k} \sum_{i=1}^{n_k} w_k(\mathbf{U}_{k,i}) \frac{h_{j,\lambda}(\mathbf{U}_{k,i})g(\mathbf{U}_{k,i})}{p_k(\mathbf{U}_{k,i})} \right], \quad (7)$$

where $w_k(\mathbf{U})$ is a weighting function, and each sample $\mathbf{U}_{k,i}$ is drawn from $p_k(\mathbf{U})$. Weighting function $w_{k,i}(\mathbf{U})$ must satisfy the following two conditions:

- $\sum_{k=1}^K w_k(\mathbf{U}) = 1$ whenever $g(\mathbf{U}) \neq 0$, and
- $w_k(\mathbf{U}) = 0$ whenever $p_k(\mathbf{U}) = 0$.

4. Replica-Exchange Light Transport

This section gives the details of our RELT algorithm, which we will refer to as RELT. First, we present a short overview of the REMC method in Section 4.1. Then we choose four replicas that compose a family of distributions to adapt the REMC method to the light transport problem. Next, we describe in Section 4.3 how to initialize samples for each replica without start-up bias. Finally, we design a multi-sample estimator to give a completely progressive algorithm in Section 4.4.

4.1. Replica-Exchange Monte Carlo Method

The REMC method seems to have been independently discovered by several different groups in the early 1990s, and as a result, it has a variety of names: Metropolis-coupled chain algorithm, time-homogeneous parallel annealing, multiple Markov chain algorithm, and parallel tempering (see [Iba01a]). In each case, this method was developed to introduce annealing in the optimization to sampling.

The REMC method considers a set of distributions $\{p(\mathbf{u}_k|\theta_k)\}$ with different abstract hyperparameters $\{\theta_k\}$, $k = 1, \dots, K$. These distributions are called replicas. Their

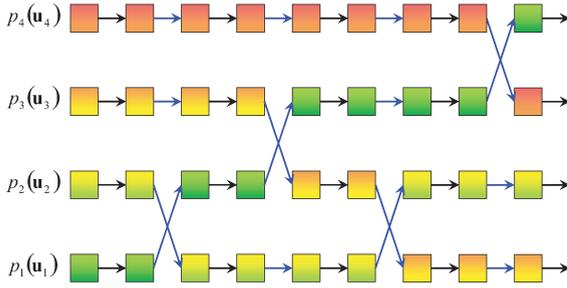


Figure 2: Replica-exchange Monte Carlo method. Black and blue arrows show conventional and exchange updates, respectively. Squares of each row are samples from each distribution p_k . In this case, we obtain four samples for each replica, since the method takes a sample after the conventional updates.

abstract hyperparameters, which represent the difficulty of sampling based on each distribution (We can easily draw samples from the uniform distribution; however, we cannot easily draw samples from multimodal probabilistic density functions. This is the difficulty of sampling), are incrementally ordered in the set as $\theta_1 \leq \dots \leq \theta_K$ by difficulty. The method samples simultaneous distribution $\tilde{p}(\{\mathbf{u}_k\})$ of $\{\mathbf{u}_k\}$ as stationary distribution for performing such sampling as annealing by the Markov chain Monte Carlo method:

$$\tilde{p}(\{\mathbf{u}_k\}) = \prod_{k=1}^K p(\mathbf{u}_k | \theta_k). \quad (8)$$

The method uses two types of updates for sampling: conventional and replica-exchange. A conventional update is independently defined in each replica k and has to satisfy the detailed balance condition for $p(\mathbf{u}_k | \theta_k)$. This means that this update equals the conventional Markov chain Monte Carlo method within each replica. On the other hand, the replica-exchange update exchanges state \mathbf{u} of the replicas with neighbouring parameters θ_k and θ_{k+1} based on exchange probability $\max\{1, r\}$, where

$$\begin{aligned} r &= \frac{p(\mathbf{u}_1, \dots, \mathbf{u}_{k+1}, \mathbf{u}_k, \dots, \mathbf{u}_K)}{p(\mathbf{u}_1, \dots, \mathbf{u}_k, \mathbf{u}_{k+1}, \dots, \mathbf{u}_K)} \\ &= \frac{p_k(\mathbf{u}_{k+1})p_{k+1}(\mathbf{u}_k)}{p_k(\mathbf{u}_k)p_{k+1}(\mathbf{u}_{k+1})}, \end{aligned} \quad (9)$$

and $p_k(\mathbf{u}_k) = p(\mathbf{u}_k | \theta_k)$. The simultaneous distribution is invariant under the exchange, and the detailed balance condition for the simultaneous distribution is satisfied. Obtained samples $\mathbf{u}_{k,i}$ from each replica are considered samples drawn from $p_k(\mathbf{u}_k)$ (Figure 2).

We have to calculate a normalization constant $b_k = \int_{\mathcal{U}} q_k(\mathbf{u}) d\mathbf{u}$ for each distribution because distributions for the REMC method are often unnormalized, where $q_k(\mathbf{u}) =$

$b_k p_k(\mathbf{u})$ is an unnormalized distribution. If we know b_1, b_k can recursively compute a normalization constant for each distribution $p_k(\mathbf{u})$ by Equation (10):

$$b_k = b_{k-1} \int_{\mathcal{U}} \frac{q_k(\mathbf{u})}{q_{k-1}(\mathbf{u})} p_{k-1}(\mathbf{u}) d\mathbf{u}. \quad (10)$$

4.2. Designing a Set of Replicas for Solving the Light Transport Problem

In this section, we describe how to design a set of replicas for solving the light transport problem. First, we discuss the properties of an integrand of the light transport integration. We use Equation (11) as an image contribution function that was defined for bidirectional path tracing [VG95]:

$$g(\mathbf{u}) = \sum_{s,t \geq 0} w_{s,t}(S(\mathbf{u})) \frac{f(S(\mathbf{u}))}{p_{s,t}(S(\mathbf{u}))}, \quad (11)$$

where $p_{s,t}(\mathbf{u})$ is a PDF in a bidirectional sampling strategy, s is the number of vertices of a light subpath, and t is the number of vertices of an eye subpath. This function is evaluated by generating path samples that are transformed from random numbers in primary sample space (as shown in Section 4.1).

Even though Equation (11) is useful as an integrand for rendering direct/indirect illumination by bidirectional path tracing, it often suffers from multi-modal probability density because that function is composed of several types of paths. For examples, generated paths by $p_{0,t}, t \geq 0$, are directly reached to a light source called an implicit lighting path. On the other hand, paths by $p_{s,1}, s \geq 0$, are explicitly connected to the eye called an explicit caustic path.

The main advantage of the Metropolis method is using the correlation of consecutive samples. So when we used the Metropolis method, we assumed the PDF of the integrand implicitly has a continuous shape, i.e., the near points of a high contribution point also have high contributions. However, Equation (11) does not satisfy such a property for the above reasons.

Now we classify Equation (11) based on its features for designing a set of replicas.

4.2.1. Classification of Paths

The integrand of the light transport problem is a linear combination of different path sampling strategies classified into five types: implicit lighting paths based on $p_{0,t}, t \geq 0$, explicit lighting paths based on $p_{1,t}, t \geq 0$, implicit caustic paths based on $p_{s,0}, s \geq 0$, explicit caustic paths based on $p_{s,1}, s \geq 0$, and indirect lighting paths based on $p_{s,t}, s, t \geq 2$ [VG97]. We also classify implicit lighting paths into two parts to handle difficult lighting configurations: implicit lighting paths that contain one or more specular scatterings and

implicit lighting paths without specular scatterings. The former based on $p_{0,t}^+$ and the latter on $p_{0,t}^-$ for ordering them. Each path type is designed to satisfy the coherence of contribution in mutation.

Contributions from those types of paths are driven from Equation (11): a contribution obtained through the implicit lighting paths is given by $L_0(\mathbf{u})$ or $L_1(\mathbf{u})$, a contribution obtained through the explicit lighting paths is given by $L_2(\mathbf{u})$, a contribution obtained through the implicit caustic paths is given by $L_3(\mathbf{u})$, a contribution obtained through the explicit caustic paths is given by $L_4(\mathbf{u})$, and a contribution obtained through the indirect lighting path is given by $L_5(\mathbf{u})$. These details are summarized below

$$L_0(\mathbf{u}) = \sum_{t \geq 0} w_{0,t}(S(\mathbf{u})) \frac{f(S(\mathbf{u}))}{p_{0,t}^+(S(\mathbf{u}))}, \quad (12)$$

$$L_1(\mathbf{u}) = \sum_{t \geq 0} w_{0,t}(S(\mathbf{u})) \frac{f(S(\mathbf{u}))}{p_{0,t}^-(S(\mathbf{u}))}, \quad (13)$$

$$L_2(\mathbf{u}) = \sum_{t \geq 0} w_{1,t}(S(\mathbf{u})) \frac{f(S(\mathbf{u}))}{p_{1,t}(S(\mathbf{u}))}, \quad (14)$$

$$L_3(\mathbf{u}) = \sum_{s \geq 0} w_{s,0}(S(\mathbf{u})) \frac{f(S(\mathbf{u}))}{p_{s,0}(S(\mathbf{u}))}, \quad (15)$$

$$L_4(\mathbf{u}) = \sum_{s \geq 0} w_{s,1}(S(\mathbf{u})) \frac{f(S(\mathbf{u}))}{p_{s,1}(S(\mathbf{u}))}, \quad (16)$$

$$L_5(\mathbf{u}) = \sum_{s,t \geq 2} w_{s,t}(S(\mathbf{u})) \frac{f(S(\mathbf{u}))}{p_{s,t}(S(\mathbf{u}))}. \quad (17)$$

Using these descriptions, the image contribution function is given by

$$g(\mathbf{u}) = \sum_{n=0}^5 L_n. \quad (18)$$

4.2.2. Replica Design

We built a replica for Metropolis sampling by linear combination of the contributions from different path strategies. We use Equation (19) as a k th replica:

$$\begin{aligned} p_k(\mathbf{u}) \propto q_k(\mathbf{u}) &= \sum_{n=0}^5 d_{k,n} L_n(\mathbf{u}) \\ &= \mathbf{d}_k \cdot \mathbf{L}(\mathbf{u}), \end{aligned} \quad (19)$$

where $q_k(\mathbf{u})$ is an unnormalized PDF, and $d_{k,n}$ is a weight that is a user specified parameter to build replica.

4.2.3. Building a Set of Replicas

We built a set of replicas for replica-exchange sampling by choosing four types of replicas: uniform distribution, full path, indirect path, and specific path.

Uniform distribution replica $p_1 = 1$ is the foundation of the RELT algorithm since it allows the algorithm to satisfy ergodicity. This replica draws samples from the uniform distribution on the primary sample space, which is a unit hypercube. So the integration value of this replica is one, i.e. $\int_{\mathcal{U}} p_1(\mathbf{u}) d\mathbf{u} = 1$. This replica is not based on Equation (19), which is a special case of our replica design.

Full path replica $p_2 \propto q_2$ is a PDF of image contribution function $g(\mathbf{u})$; here we remove implicit lighting paths that do not have specular scattering $p_{0,t}(\mathbf{u})$ to focus on large contribution portions. Such paths often make small contributions in multiple importance sampling. This replica, which is a case of $\mathbf{d}_2 = (101111)$ in Equation (19), was designed with reference to [KSKGC02].

Indirect path replica $p_3 \propto q_3$ is a case of $\mathbf{d}_3 = (100111)$. We designed this replica for focusing on indirect illumination.

Specific path replica $p_4 \propto q_4$ is a case of $\mathbf{d}_4 = (10000\delta)$, where δ is a small value for sampling robustness. If $\delta = 0$, q_4 is often zero in an integration domain, so δ prevents such situations. This replica is for handling difficult lighting configurations including specular-diffuse-specular paths.

All unnormalized distributions, $\{q_k\}$, $k = 1, 2, 3, 4$, are evaluated simultaneously by building a path of each replica from a bidirectional path [VG95].

The MLT algorithm based on the primary sample space [KSKGC02] can be considered as a subset of our RELT algorithm. It uses uniform distribution for large perturbations and the PDF of the image contribution function for small perturbations. That is, our algorithm is formulated by an simple extension of the MLT, and the algorithm improves sampling efficiency by introducing additional sampling densities.

4.3. Eliminating Start-Up Bias

This section shows how the RELT algorithm, which uses extended distribution, can be initialized to avoid start-up bias for each replica. In practice, initializing the RELT algorithm with standard MLT initializing techniques [VG97] works poorly. If we generate samples for each replica by resampling only once, there are often few samples from $p_K(\mathbf{u})$.

The key idea for unbiased initialization to avoid start-up bias is to adapt a sequential Monte Carlo method to a family of distributions instead of a time-variable distribution. This sampling technique is called population annealing in the field of statistical physics [Iba01b].

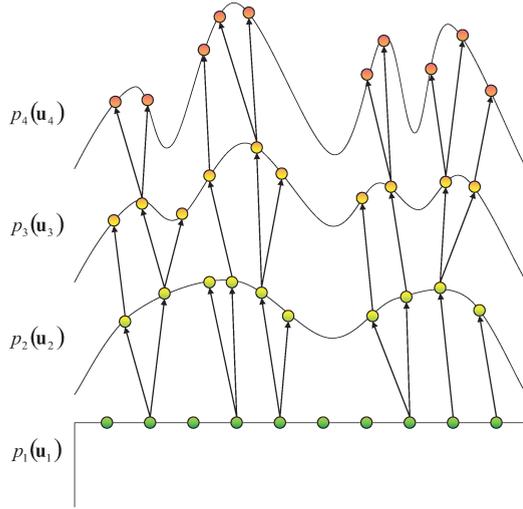


Figure 3: Population annealing: Curves show abstract shapes of each PDF. Particles are propagated from p_1 to p_K by resampling and Markov chain Monte Carlo transition. Then important particles are split and increased, and unimportant particles are deleted.

The sequential Monte Carlo method, which is a technique for sampling from a time-variable distribution, has two steps: propagating and resampling. The propagating step creates temporary samples in the current time step from the previous time step and assigns weights to the created samples. Then the resampling step draws samples from the temporary samples based on their weights, and these resampled samples are used as current time step samples. [GDH06] efficiently obtained samples from the BRDF and illumination products of dynamic environment maps using the sequential Monte Carlo method. On the other hand, several algorithms use only one time resampling. [LW95] use a resampling to reduce the number of shadow rays in bidirectional path tracing. [TCE05, BGH05] generate a sample set based on the product distribution of BRDF and environment map illumination by resampling.

Population annealing uses multiple copies called particles to represent distributions. These particles have weights c_i , which are used for resampling. Weighted particles are commonly used in such population Monte Carlo methods as the sequential Monte Carlo algorithm. Unnormalized incremental weight c_i of every particle for population annealing at replica k is given by the following ratio:

$$c_i = \frac{q_k(\mathbf{u}_i)}{q_{k-1}(\mathbf{u}_i)}. \quad (20)$$

These particles are propagated from p_1 to p_K by resampling and Markov chain Monte Carlo transition (Figure 3). After resampling, the particle weight is set to one.

The pseudocode of the initialization using population annealing for any family of distributions parameterized by a parameter is shown in Algorithm 2. This algorithm can also calculate the expected values of each replica at each iteration k by Equation (21):

$$E \left[\frac{q_{k+1}(\mathbf{u})}{q_k(\mathbf{u})} \right] = \sum_{i=1}^N c_i \left(= \int_{\mathcal{U}} \frac{q_{k+1}(\mathbf{u})}{q_k(\mathbf{u})} q_k(\mathbf{u}) \right). \quad (21)$$

Algorithm 2 Pseudocode of the unbiased initialization using population annealing

```

1: set each weight of particles to 1
2: for  $k = 1$  to  $K$  do
3:   if  $k = 1$  then
4:     draw particles from a uniform distribution and
       evaluate them
5:   else
6:     resample particles using their weights
7:     set their weight to one:  $c_i = 1$ 
8:     mutate particles based on  $p_k$ 
9:   end if
10:  randomly take samples as initial state  $\mathbf{u}$  of  $p_k$  from
      current particles
11:  if  $k \neq K$  then
12:    compute each particle weight:  $c_i = \frac{q_{k+1}}{q_k}$ 
13:    compute an expected value:  $E \left[ \frac{q_{k+1}}{q_k} \right]$ 
14:  end if
15: end for

```

4.4. Progressive Rendering

The progressive rendering algorithm is useful for synthesizing images with global illumination because the number of samples N to get a fully converged result is unknown before rendering. In addition, if the algorithm is multi-pass, adjusting the number of samples becomes more difficult. Thus we construct a single-pass and progressive rendering algorithm.

We obtain samples from each distribution of replicas chosen as above with the REMC method. To compute expected value by Equation (7), we cannot use a standard balance heuristic, a power heuristic, or any other heuristic because they need to evaluate the normalized PDF. Instead we use an extended power heuristic (Equation (22)) as a weight function to calculate the expected value:

$$w_k(\mathbf{u}_k) = \frac{b_k \left(\frac{q_k(\mathbf{u}_k)}{b_k} \right)^{\beta-1} p_k(\mathbf{u}_k)}{\hat{b}_k \sum_{l=1}^K \left(\frac{q_l(\mathbf{u}_k)}{b_l} \right)^{\beta}}, \quad (22)$$

where \hat{b}_k is a roughly estimated integral value calculated in the initialization phase by Equation (10), and $\beta \geq 1$ is a user-specified parameter that is identical as the parameter in

the power heuristic. This heuristic works well, since often $\frac{b_k}{\hat{b}_k} \approx 1$.

In our implementations, each replica has the same number of samples $n_k = N$ (This does not introduce any bias), and the estimator is obtained as follows:

$$m_{j,\lambda} = E \left[\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K \frac{b_k}{\hat{b}_k} \frac{h_{j,\lambda}(\mathbf{u}_{k,i}) \left(\frac{q_k(\mathbf{u}_{k,i})}{b_k} \right)^{\beta-1} g(\mathbf{u}_{k,i})}{\sum_{l=1}^K \left(\frac{q_l(\mathbf{u}_{k,i})}{b_l} \right)^{\beta}} \right]. \quad (23)$$

Since this shows that the evaluated values of the samples from each replica can be calculated independently, we use K buffers to accumulate the evaluated values. Then these buffers are composited to generate a final image. As a result, we can get an image progressively.

5. Implementation and Results

This section describes the results and discussion. We first describe the details of our implementation and then discuss some results.

We rendered test images to compare our proposed RELT with standard BPT and two types of MLT algorithms: MLT based on the primary sample space [KSKGC02] and MLT with bidirectional mutation and lens- and caustic-perturbations. To distinguish them, we call the first simplified MLT (SMLT) and the second method full MLT (FMLT). Here, the PDF of FMLT excludes direct lighting paths for focusing on indirect lighting calculation.

5.1. Implementation Details

We present the implementation details of our RELT algorithm in this section. Suppose that we already have a path builder implementation, which is a bidirectional path tracing with multiple importance sampling based on path density. It uses a finite maximum path length for path building to avoid unsuitable long length sample generation. That is, we solve a subproblem of infinite domain of integration (This is unbiased.) [VG97]. Additionally, our ray-tracing implementations support kd-tree with surface area heuristics (SAH-kd-tree) as spatial data structure, importance-sampled BSDFs, direct lighting calculations, and several other optimizations.

5.1.1. Spectral Sampling

We represent BSDFs and light sources as point-sampled spectra. Given a path, we compute the energy delivered to the camera film at the sampled wavelength. The obtained contributions have three parameters: two-dimensional coordinates on the image-plane and wavelength in a spectral range of 400–700 nm. The resulting spectrum is then converted to a tristimulus colour value (we use XYZ) before it is accumu-

lated in the current image. The PDF of the image contribution function is defined as the sum of the tristimulus colour value without using the luminance of the colour value. Human eye is substantially more sensitive to luminance differences than other colour variations; however, for rendering such spectral effects as dispersion, luminance cannot be used for the PDF definition. Images obtained by MLT with a small number of samples are often yellow-tinged, because yellow is much brighter than other colours.

5.1.2. Initialization Cost

The initialization phase is a negligible part of the total computation time. For example, even 10,000 bidirectional path samples typically constitute less than one sample per pixel. The computation time for initialization is less than one second in our following results.

5.1.3. Mutation Strategy

We use the formula of Equation (24) for the mutations in the primary sample space [SKDP99]:

$$s = s_2 \exp \left(- \log \left(\frac{s_2}{s_1} \xi \right) \right), \quad (24)$$

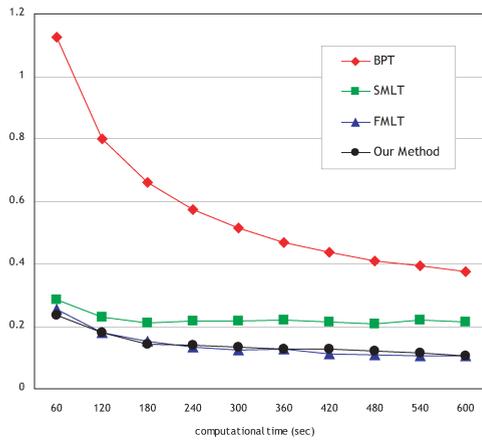
where ξ is a uniformly distributed random variable in $[0, 1)$ and the samples are expected in $[s_1, s_2)$. The actual mutation function equals [SKDP99], and we use different s_1 and s_2 for each replica. Parameter s_1 for the mutation of each replica is always $s_2 = 16s_1$, and s_1 for the full, the indirect, and the specific path replicas is $\frac{1}{61}$, $\frac{1}{61}$, and $\frac{1}{96}$, respectively.

5.1.4. Multiple Importance Sampling

We use multiple importance sampling (MIS) to combine the samples from each replica. Here, we cannot simply use Equation (23), because the naive application of MIS increases errors. In the naive application of MIS, samples from p_K often provide impulse noises. Instead we compute contributions from implicit lighting paths that contain one or more specular scattering events by only using specific path replicas. Furthermore, due to combining samples from three other replicas, we calculate the remaining terms. That is, we have four image buffers for accumulating contributions from each replica: one is for the implicit lighting paths that contain one or more specular scattering events, and the other is for the remaining terms.

5.2. Results

First, we compare the results of simple lighting configurations whose spectral range is 400–700 nm. All results were rendered on a 2.13 GHz Intel Core 2 CPU 6400. We compare our technique with BPT and MLTs.



(a) Root Mean Square Error



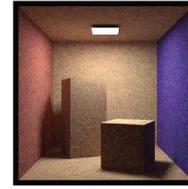
(b) Reference



(c) BPT



(d) SMLT

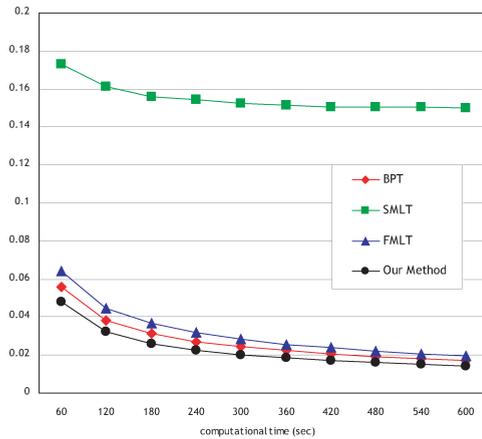


(e) FMLT



(f) Our Method

Figure 4: Comparison of a simple direct lighting scene: Rendering times of those images are approximately 10 min, and all images are 400×400 pixels.



(a) Root Mean Square Error



(b) Reference



(c) BPT



(d) SMLT



(e) FMLT



(f) Our Method

Figure 5: Comparison of a simple indirect lighting scene: Rendering times of those images are approximately 10 min, and all images are 400×400 pixels.

5.2.1. Simple Lighting Configuration

We measured the rendering errors using the Root Mean Square Error (RMSE) as measurement:

$$\sqrt{\frac{\sum_{i=1}^n (\hat{x}_i - x_i)^2}{n}}, \quad (25)$$

where \hat{x} is a test vector and x is a reference vector. We computed RMSE using a rendered image as a $n (= QR)$ -dimensional vector, where Q is the number of pixels and R is the number of pixel components. We use XYZ colour space $R = 3$. Reference images were calculated by BPT with a large number of samples.

Figure 4 compares our method to a BPT that uses uniform sampling as a sampler and two types of MLTs (SMLT and FMLT) that use Metropolis sampling as a sampler in a direct lighting configuration scene. SMLT (Simplified MLT) is MLT based on the primary sample space [KSKGC02], and FML (Full MLT) is MLT with bidirectional mutation and lens- and caustic-perturbations. At any time, our method, which can produce an image with lower error than BPT and SMLT, has an error level equivalent to the FMLT; however FMLT's result contains visually conspicuous noise, because its PDF does not have a direct lighting term for focusing on indirect illumination.

In Figure 5, we applied our method to indirect lighting configuration in a scene that mainly contains indirect

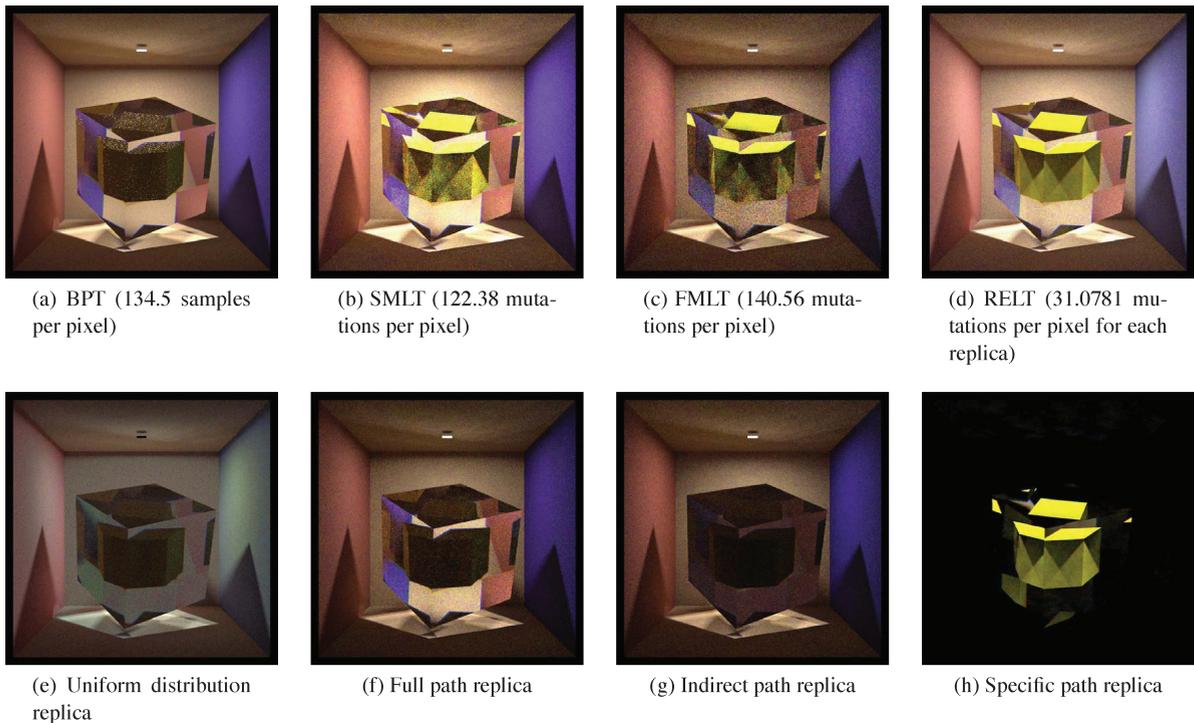


Figure 6: Comparison of caustics seen through a glass: Upper row images are results produced by each algorithm. Lower row images show results by individual replicas in the RELT algorithm. All images are 800×800 pixels, generated in identical computation time (20 min). Large portion of illumination in this scene comes from caustics. Illumination in the glass, which needs to find the specular-diffuse-specular transport paths, is a particularly difficult caustic lighting situation.

illumination. At any time, our method can produce an image with lower error than the other methods. The SMLT has larger errors than the BPT, the FMLT and the RELT due to its initialization phase. The SMLT only computes the normalization term for MIS in the initialization phase. This produces estimation biases.

MLT provides benefits for rendering common scenes by applying Metropolis sampling; however, mutation strategies need to be designed and set carefully. Our approach, which can straightforwardly design a mutation strategy by combining simple mutation strategies, is robust. Furthermore, as compared with SMLT, our approach has good convergence characteristic (Figure 4 and 5) and resulting in fewer visual artefacts more as compared with FMLT (Figure 5).

5.2.2. Complex Lighting Configuration

In the following results, large parts of the lighting contain caustics seen through a surface with BSDF composed of the Dirac delta function, including mirrors, glasses, and mirror-varnished surfaces. Their caustics are calculated by light contributions along implicit caustics paths. Note that these lighting situations are exceedingly difficult for unbiased algorithms.

Figure 6 shows a test scene with difficult caustic lighting seen through a glass. A yellow cube is in a glass cube, and a small light source is on the ceiling. This scene has diffuse, specular, and transparent surfaces that are untextured to clearly reveal the noise levels. The path space of this scene contains several types of light transport paths, i.e. direct lighting, indirect lighting, caustic lighting, and caustic direct lighting seen through a mirror/glass. Computing the results with approximately equal computation time, RELT creates better results than BPT, SMLT, and FMLT. Image (a), produced by BPT, cannot sample the specular-diffuse-specular paths. Images (b,c) by MLTs are much better, but still the results are noisy. SMLT produced a much brighter result. Image (d), our result, has a much smoother appearance, because our sampling strategy defined the separate roles over the path space by four replicas. Images (e-h) show the results from individual replicas; image (d) was obtained by summing these images. Notice that these results show different portions of light transport over the path space. Image (e) was obtained by the uniform distribution replica, which rendered the scene's darker areas. Image (f), a result of the full path replica, shows the scene's brighter areas. Image (g) by the indirect path replica rendered indirect illuminations in the scene. This replica bridges the bidirectional path and specific replicas. Image (h) by the specific replica shows caustic direct/indirect

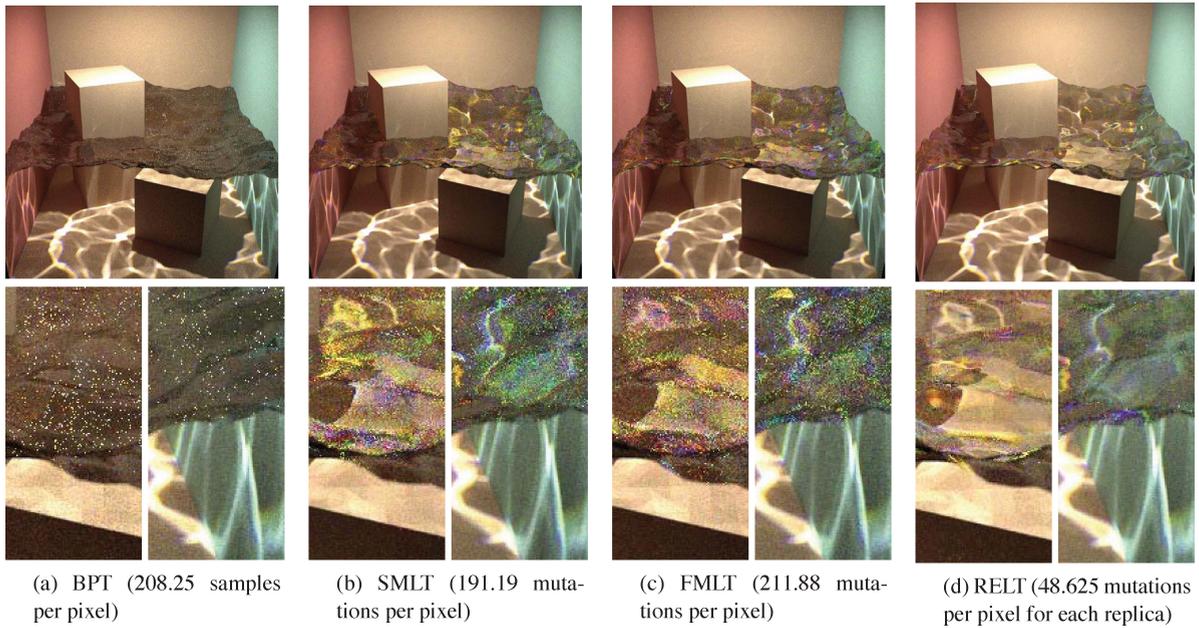


Figure 7: *Water-filled Cornell box: A small light source is on the ceiling (identical as scene in Figure 6). All images are 800×800 pixels, generated in identical computation time (60 min). These upper images show the difficult caustic lighting situations, and lower row images are their close-ups.*

lighting seen through specular reflection/refraction (i.e. in the glass).

Figure 7 compares RELT to BPT and MLTs for a more difficult caustic lighting situation with direct/indirect illumination. This is a water-filled Cornell box, and there is a small area light source on the ceiling identical with the scene in Figure 6. Note that in this scene the water surface radiation comes from the caustics on the floor. As shown in image (a), BPT produces a dark water surface, since it could not implicitly find a light source. The MLTs shown in images (b,c) outperformed BPT; however the result is still noisy. On the other hand, the RELT (image (d), our result) obviously provided a smoother result than the other three algorithms.

Figure 8 shows another result produced by incomplete RELT in the same computation time as Figures 6 and 7. Their results, which are rendered by RELT without an exchange process, are obviously imperfect. The yellow box in Image (a) is dark, and the water surface in Image (b) is also dark with excessively bright areas produced by the slow mixing. This is because the exchange process in RELT is for satisfying the ergodicity of the Markov chain. So traditional sampling strategies that simply use two or more distributions could not be applied to rendering by the Markov chain Monte-Carlo method.

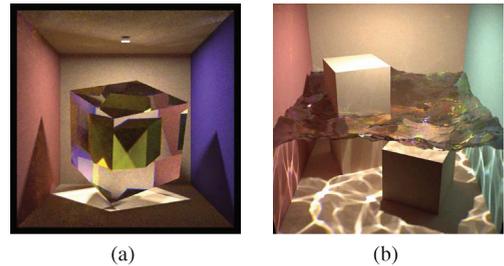


Figure 8: *Results produced by RELT without exchange process.*

6. Conclusion and Future Work

We have proposed an unbiased global illumination algorithm by adapting the REMC method to the light transport problem. Our algorithm starts from a few seed light transport paths and applies a sequence of conventional and replica-exchange updates to them. This algorithm is very easy to implement, improves sampling efficiency for general scenes, and can render progressively without determining the number of samples. Experimental results showed that our algorithm has better convergence properties than BPT and MLTs, the existing algorithms.

Even though our algorithm is efficient for difficult lighting situations, it has some limitations. The set of replicas proposed in this paper is inefficient for lighting situations that do not have specular-diffuse-specular paths since the distribution of the specific path replica is always very small. To handle this problem, the algorithm uses a set of replicas without a specific path replica in such lighting situations and determines whether the specific replica is removed based on \hat{b}_4 in the initialization step. As an extension of MLT, unfortunately, our algorithm also inherits some of its drawbacks. For example, although it is efficient for difficult lighting situations, another algorithm is more efficient for simple lighting situations, e.g. a classical path-tracing algorithm is most efficient for rendering, if all of surfaces have a Lambert BRDF and a large portion of the lighting is direct illumination.

Many extensions and refinements of this basic idea are possible. First, for example, a partially ordered set of replicas can be used for the sampling. Our RELT algorithm uses a totally ordered set of replicas for the sampling, because if we use a set that includes path and light tracing replicas to trace a path starting from a light source, their product set is empty. Second, the dimension of the primary sample space can be extended to render a motion blur effect caused by finite shutter speed, lens flare, and depth-of-field effects caused by lens. These are easy to implement. Moreover, a rendering algorithm that uses only population annealing can be designed without the REMC method to evaluate its efficiency. We will address some of these refinements and extensions in the future.

References

- [AK90] ARVO J., KIRK D.: Particle transport and image synthesis. In *Proceedings of ACM SIGGRAPH '90* (1990), pp. 63–66.
- [APSS01] ASHIKHMIN M., PREMOZE S., SHIRLEY P., SMITS B.: A variance analysis of the metropolis light transport algorithm. *Computers and Graphics* 25, 2 (2001), 287–294.
- [BGH05] BURKE D., GHOSH A., HEIDRICH W.: Bidirectional importance sampling for direct illumination. In *Proceedings of Eurographics Symposium on Rendering '05* (2005), pp. 147–156.
- [CTE05] CLINE D., TALBOT J., EGBERT P.: Energy redistribution path tracing. In *Proceedings of ACM SIGGRAPH '05* (2005), pp. 1186–1195.
- [GDH06] GHOSH A., DOUCET A., HEIDRICH W.: Sequential sampling for dynamic environment map illumination. In *Proceedings of Eurographics Symposium on Rendering '06* (2006), pp. 115–126.
- [HBHS05] HAVRAN V., BITTNER J., HERZOG R., SEIDEL H.-P.: Ray maps for global illumination. In *Proceedings of Eurographics Symposium on Rendering '05* (2005), pp. 43–54.
- [Iba01a] IBA Y.: Extended ensemble Monte Carlo. *International Journal of Modern Physics C* 12 (2001), 623–656.
- [Iba01b] IBA Y.: Population Monte Carlo algorithms. *Transactions of the Japanese Society for Artificial Intelligence* 16, 2 (2001), 279–286.
- [Jen96] JENSEN H. W.: Global illumination using photon maps. In *Proceedings of Eurographics Workshop on Rendering '96* (1996), pp. 21–30.
- [Kaj86] KAJIYA J. T.: The rendering equation. In *Proceedings of ACM SIGGRAPH '86* (1986), pp. 143–150.
- [KSKGC02] KELEMEN C., SZIRMAY-KALOS L., GYÖRGY A., CSONKA F.: A simple and robust mutation strategy for metropolis light transport algorithm. *Computer Graphics Forum* 21, 3 (2002), 1–10.
- [LFCD06] LAI Y.-C., FAN S., CHENNEY S., DYER C.: Photorealistic image rendering with population Monte Carlo energy redistribution. In *Proceedings of Eurographics Symposium on Rendering '07* (2006), pp. 287–296.
- [LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bi-directional path tracing. In *Proceedings of Compugraphics '93* (1993), pp. 145–153.
- [LW95] LAFORTUNE E. P., WILLEMS Y. D.: Reducing the number of shadow rays in bidirectional path tracing. In *Proceedings of the Winter School of Computer Graphics and CAD Systems '95* (1995), pp. 384–392.
- [LW96] LAFORTUNE E. P., WILLEMS Y. D.: Rendering participating media with bidirectional path tracing. In *Proceedings of Eurographics Workshop on Rendering '96* (1996), pp. 92–101.
- [PKK00] PAULY M., KOLLIG T., KELLER A.: Metropolis light transport for participating media. In *Proceedings of Eurographics Workshop on Rendering '00* (2000), pp. 11–22.
- [SIP07] SEGOVIA B., IEHL J.-C., PÉROCHE B.: *Coherent Metropolis Light Transport with Multiple-Try Mutations*. Tech. Rep., LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/Ecole Centrale de Lyon, 2007.
- [SKAS05] SZIRMAY-KALOS L., ANTAL G., SBERT M.: Go with the winners strategy in path tracing. In *WSCG (Journal Papers)* (2005).
- [SKDP99] SZIRMAY-KALOS L., DORNBACH P., PURGATHOFER W.: On the start-up bias problem of metropolis sampling. In *Proceedings of WSCG '99* (1999), pp. 273–280.

- [SWH*95] SHIRLEY P., WADE B., HUBBARD P. M., ZARESKI D., WALTER B., GREENBERG D. P.: Global illumination via density estimation. In *Proceedings of Eurographics Workshop on Rendering '95* (1995), pp. 219–230.
- [TCE05] TALBOT J. F., CLINE D., EGBERT P. K.: Importance resampling for global illumination. In *Proceedings of Eurographics Symposium on Rendering '05* (2005), pp. 139–146.
- [VG94] VEACH E., GUIBAS L.: Bidirectional estimators for light transport. In *Proceedings of Eurographics Workshop on Rendering '94* (1994), pp. 147–162.
- [VG95] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of ACM SIGGRAPH '95* (1995), pp. 419–428.
- [VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. In *Proceedings of ACM SIGGRAPH '97* (1997), pp. 65–76.
- [WRC88] WARD G. J., RUBINSTEIN F. M., CLEAR R. D.: A ray tracing solution for diffuse interreflection. In *Proceedings of ACM SIGGRAPH '88* (1988), pp. 85–92.