Simple and Robust Mutation Strategy for Metropolis Light Transport Algorithm

Csaba Kelemen and László Szirmay-Kalos

Department of Control Engineering and Information Technology, Technical University of Budapest
Budapest, Magyar Tudósok krt. 2, H-1117, HUNGARY
Email: szirmay@iit.bme.hu

Abstract

The paper presents a new mutation strategy for the Metropolis light transport algorithm, which works in the space of uniform random numbers used to build up paths. Thus instead of mutating directly in the path space, mutations are realized in the infinite dimensional unit cube of pseudo-random numbers and these points are transformed to the path space according to BRDF sampling, light source sampling and Russian roulette. This transformation makes the integrand and the importance function flatter and thus increases the acceptance probability of the new samples in the Metropolis algorithm. Higher acceptance ratio, in turn, reduces the correlation of the samples, which increases the speed of convergence. When mutations are calculated, a new random point is selected usually in the neighborhood of the previous one, but according to our proposition called "large steps", sometimes an independent point is obtained. Large steps greatly reduce the start-up bias and guarantee the ergodicity of the process. Due to the fact that some samples are generated independently of the previous sample, this method can also be considered as a combination of the Metropolis algorithm with a classical random walk. Metropolis light transport is good in rendering bright image areas but poor in rendering dark sections since it allocates samples proportionally to the pixel luminance. Conventional random walks, on the other hand, have the same performace everywhere, thus they are poorer than Metropolis method in bright areas, but are better at dark sections. In order to keep the merits of both approaches, we use multiple importance sampling to combine their results, that is, the combined method will be as good at bright regions as Metropolis and at dark regions as random walks. The resulting scheme is robust, efficient, but most importantly, is easy to implement and to combine with an available random-walk algorithm.

1. Introduction

The fundamental idea of Monte-Carlo quadrature is to convert an integral to an expected value, which is then estimated by the average of samples:

$$\int_{\mathcal{P}} F(\mathbf{z})\,d\mathbf{z} = \int_{\mathcal{P}} \frac{F(\mathbf{z})}{p(\mathbf{z})} \cdot p(\mathbf{z})\,d\mathbf{z} = E\left[\frac{F(\mathbf{z})}{p(\mathbf{z})}\right] \approx$$

$$\frac{1}{M} \cdot \sum_{i=1}^{M} \frac{F(\mathbf{z}_i)}{p(\mathbf{z}_i)} \pm \frac{\sigma}{\sqrt{M}}, \qquad (1)$$

where $p(\mathbf{z})$ is a probability density in $\mathcal{P}$, the $\mathbf{z}_i$ points are selected according to this probability density, and σ is the variance of random variable $F(\mathbf{z})/p(\mathbf{z})$. Probability density $p(\mathbf{z})$ should be selected to minimize the variance. As can be shown easily, the variance can be minimized if $p(\mathbf{z})$ is proportional to the integrand $F(\mathbf{z})$. Thus in Monte-Carlo integration it is worth applying probability distributions that are large where the integrand is large and small where the integrand is negligible. This variance reduction technique is called the *importance sampling* [8].

In the context of the rendering equation, importance sampling prefers useful paths along which significant radiance is transferred. Since a path carries radiance on several different wavelengths, the integrand $F(\mathbf{z})$ is a vector, thus the selection of a "proportional" probability density requires further considerations. In order to express where the elements of vector $F(\mathbf{z})$ are large, a scalar importance function $\mathcal{I}(\mathbf{z})$

presented at Dagstuhl Seminar on Monte-Carlo Methods in Rendering 2001.

is defined. This importance function can, for example, represent the luminance of the carried light.

Although the contribution on the image is a function of the complete path, it is not possible to construct an explicit sampling scheme that samples with this probability, thus random walk algorithms usually assign estimated importance to individual steps of this path. In a single step the importance is usually selected according to the BRDF [3, 4], or according to the direction of the direct lightsources [7].

The adaptive character of Metropolis sampling, however, offers a different approach. Unlike other methods, Metropolis sampling can assign importance to a complete walk not just to the steps of this walk, and it explores important regions of the domain adaptively [11]. Thus no a-priori knowledge is required about the important rays to construct a probability density function in advance. Instead, the algorithm converges to this probability density automatically.

1.1. Metropolis sampling

The Metropolis algorithm [5] converges to the optimal probability density that is proportional to the importance, that is in the limiting case $\mathcal{I}(\mathbf{z}) = b \cdot p(\mathbf{z})$. Scalar b comes from the requirement of normalization:

$$b = \int_{\mathcal{P}} \mathcal{I}(\mathbf{z})\, d\mathbf{z}.$$

However, this probability density cannot be stored, thus in the Monte-Carlo formula the importance should be used instead, in the following way:

$$\Phi = \int_{\mathcal{P}} \frac{F(\mathbf{z})}{\mathcal{I}(\mathbf{z})} \cdot \mathcal{I}(\mathbf{z})\, d\mathbf{z} = b \cdot \int_{\mathcal{P}} \frac{f(\mathbf{z})}{\mathcal{I}(\mathbf{z})} \cdot p(\mathbf{z})\, d\mathbf{z} =$$

$$b \cdot E\left[\frac{F(\mathbf{z})}{\mathcal{I}(\mathbf{z})}\right] \approx \frac{b}{M} \cdot \sum_{i=1}^{M} \frac{F(\mathbf{z}_i)}{\mathcal{I}(\mathbf{z}_i)} \qquad (2)$$

To generate samples according to $p(\mathbf{z}) = 1/b \cdot \mathcal{I}(\mathbf{z})$, a Markovian process is constructed whose stationary distribution is just $p(\mathbf{z})$. The next state $\mathbf{z}_{i+1}$ of this process is found by letting an almost arbitrary *tentative transition function* $T(\mathbf{z}_i \to \mathbf{z}_t)$ generate a *tentative sample* $\mathbf{z}_t$ which is either accepted as the real next state or rejected making the next state equal to the actual state. The decision uses the "*acceptance probability*" $a(\mathbf{z}_i \to \mathbf{z}_t)$ that expresses the increase of the importance (if this "acceptance probability" is greater than 1, then the sample is accepted deterministically). The formal definition of this Markovian process $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_i, \ldots\}$ is as follows:

```
for $i = 1$ to $M$ do
    Based on $\mathbf{z}_i$, choose a tentative point $\mathbf{z}_t$ using $T(\mathbf{z}_i \to \mathbf{z}_t)$
    $a(\mathbf{z}_i \to \mathbf{z}_t) = \frac{\mathcal{I}(\mathbf{z}_t) \cdot T(\mathbf{z}_t \to \mathbf{z}_i)}{\mathcal{I}(\mathbf{z}_i) \cdot T(\mathbf{z}_i \to \mathbf{z}_t)}$
                                // accept with probability $a(\mathbf{z}_i \to \mathbf{z}_t)$
    Generate random number $r$ in $[0,1]$.
    if $r < a(\mathbf{z}_i \to \mathbf{z}_t)$ then $\mathbf{z}_{i+1} = \mathbf{z}_t$
    else                          $\mathbf{z}_{i+1} = \mathbf{z}_i$
endfor
```

When it comes to the global illumination problem, $\mathbf{z}$ is a path connecting the light sources to the eye, possibly including several reflections or refractions, the integration domain $\mathcal{P}$ is the space of light paths and $F(\mathbf{z})$ is the contribution of path $\mathbf{z}$ onto a given pixel.

Veach and Guibas [11] recognized that the basic Metropolis algorithm needs to be modified to make it suitable for the solution of the global illumination problem. Such modifications allowed the integrals of all pixels to be simultaneously evaluated, the reduction of the start-up bias and the utilization of the rejected samples, and also guaranteed the required ergodicity of the Markov chain. Let us examine these modifications separately.

The global illumination solution requires the evaluations of integrals for each pixel of the screen. For a single pixel j, this integrand is the product of the measurement function $W_j(\mathbf{z})$ of this pixel and the radiance function $L(\mathbf{z})$ that is also obtained as a large dimensional integral of the Neumann series:

$$\Phi_j = \int_{\mathcal{P}} W_j(\mathbf{z}) \cdot L(\mathbf{z})\, d\mathbf{z}.$$

The radiance is independent of the measurement function of the given pixel, thus can be reused for all pixels. To exploit this, importance $\mathcal{I}$ is made "proportional" only to $L(\mathbf{z})$, and for pixel j formula 2 becomes:

$$\Phi_j = \int_{\mathcal{P}} W_j(\mathbf{z}) \cdot \frac{L(\mathbf{z})}{\mathcal{I}(\mathbf{z})} \cdot \mathcal{I}(\mathbf{z})\, d\mathbf{z} \approx \frac{b}{M} \cdot \sum_{i=1}^{M} W_j(\mathbf{z}) \cdot \frac{L(\mathbf{z}_i)}{\mathcal{I}(\mathbf{z}_i)}.$$

Metropolis sampling adaptively converges to the desired probability distribution. It also means that at the beginning of the process, the samples are not selected with the required probabilities, which introduces some error in the estimation. This error is generally called as the *start-up bias*[9]. In their original paper Veach proposed the following solution of the problem. In a preprocessing phase random samples are generated and the initial seed of the Metropolis algorithm is selected from this random population with a probability that is proportional to the importance. Since in this case even the first sample follows the desired destribution, the start-up bias problem is said to be eliminated. Of course, this is only true in the statistical sense, i.e. when many Metropolis algorithms are initiated from different seeds and their results are averaged. If the Metropolis algorithm is started from a single or a few samples, the start-up bias occurs no matter how the initial path has been selected.

The next problem comes from the requirement that the Markovian process should be ergodic, i.e. the stationary distribution should exists and be independent of the initial seed.

This requires the mutations to lead to everywhere. This problem is proposed to be handled by using a candidate generated from scratch when the process identified a path of zero contribution.

Finally, it is worth using also the rejected samples since they also provide illumination information. Note that the tentative sample is accepted with probability a, while the original sample is kept with probability $1-a$. Replacing this random variable by its mean, both locations can be contributed but the contributions of the tentative sample and the old sample should be weighted with a and $1-a$, respectively.

Summarizing, the pseudo-code of the Metropolis light transport algorithm is as follows:

```
Generate path seeds
Approximate $b = \int \mathcal{I}\, d\mathbf{z}$ from the seeds
Find $\mathbf{z}_1$ from the seeds using $\mathcal{I}(\mathbf{z}_i)$
for $i = 1$ to $M$ do
    Based on $\mathbf{z}_i$, choose a tentative point $\mathbf{z}_t$ using $T(\mathbf{z}_i \to \mathbf{z}_t)$
    $a(\mathbf{z}_i \to \mathbf{z}_t) = \min\left\{\frac{\mathcal{I}(\mathbf{z}_t)\cdot T(\mathbf{z}_t \to \mathbf{z}_i)}{\mathcal{I}(\mathbf{z}_i)\cdot T(\mathbf{z}_i \to \mathbf{z}_t)}, 1\right\}$
    Select pixel $j$ to which $\mathbf{z}_i$ contributes
    $\Phi_j \mathrel{+}= \frac{b}{M} \cdot W_j(\mathbf{z}_i) \cdot \frac{L(\mathbf{z}_i)}{\mathcal{I}(\mathbf{z}_i)} \cdot (1 - a(\mathbf{z}_i \to \mathbf{z}_t))$
    Select pixel $k$ to which $\mathbf{z}_t$ contributes
    $\Phi_k \mathrel{+}= \frac{b}{M} \cdot W_k(\mathbf{z}_t) \cdot \frac{L(\mathbf{z}_t)}{\mathcal{I}(\mathbf{z}_i)} \cdot a(\mathbf{z}_i \to \mathbf{z}_t))$
                                    // accept with probability $a(\mathbf{z}_i \to \mathbf{z}_t)$
    Generate random number $r$ in $[0,1]$.
    if $r < a(\mathbf{z}_i \to \mathbf{z}_t)$ then $\mathbf{z}_{i+1} = \mathbf{z}_t$
    else                           $\mathbf{z}_{i+1} = \mathbf{z}_i$
endfor
```

When one implements this seemingly simple algorithm, he has to face crucial design decisions and several dangers. The vital part of a Metropolis algorithm is the used mutation strategy and setting their parameters. Regions of low importance, i.e. which has positive but very little contribution, might also pose problems. These are accepted by some low probability, thus in theory the Metropolis algorithm is able to walk through them and find other important regions. However, the low acceptance probability will greatly decrease the speed, and we always be far from the asymptotically correct result.

Metropolis light transport method places samples proportionally to the importance, i.e. to the luminance or brighness of the pixel. This seems reasonable if some absolute error measure is used. However, the human eye is sensitive to relative errors, which means that targeting perceptual accuracy we should use approximately the same number of samples no matter how bright the pixel is. This requirement is met by conventional random walk algorithms, thus the combination of them with Metropolis would be desireable.

In the following subsections, we briefly review the main problems and address what kind of mutation strategies can eliminate them.

1.2. Correlated samples

Unlike other Monte-Carlo algorithms, Metropolis sampling generates not statistically independent, but correlated samples. The statistical independent sampling of Monte-Carlo quadrature guarantees that if the standard deviation of random variable $F(\mathbf{z})/p(\mathbf{z})$ is $\sigma_{primary}$, then the standard deviation of the Monte-Carlo quadrature will be $\sigma_{primary}/\sqrt{M}$ after evaluating M samples (the standard deviation is a good measure of the integration error). Since Metropolis method uses statistically correlated samples, the standard deviation of the quadrature can be determined using the Bernstein theorem [6], which states that the standard deviation of the difference between the average and the mean is bounded by

$$\sigma_{primary} \cdot \sqrt{\frac{1 + 2\sum_{k=1}^{M} R(k)}{M}} \tag{3}$$

where $\sigma_{primary}$ is the standard deviation of the primary estimator and $R(k)$ is an upperbund of the correlation between $F(\mathbf{z}_i)/p(\mathbf{z}_i)$ and $F(\mathbf{z}_{i+k})/p(\mathbf{z}_{i+k})$. This formula clearly shows that the correlation of the samples increase the error.

Taking into account the specific properties of Metropolis light transport Ashikhmin et. al. [1] obtained the following formula:

$$\sigma \leq \sqrt{\frac{bQ\mathcal{I}}{M}\left(1 + \frac{2R(1)}{p(1-p) - R(1)}\right)} \tag{4}$$

where Q is the number of pixels, $\mathcal{I}$ is the average importance of the paths associated with this pixel, p is the probability that the path contributes to the pixel under consideration and $R(1)$ is the correlation between random variables indicating that two subsequent paths go through the same pixel.

According to the analysis, strong correlation increases the variance of the integral estimate. Let us consider what it means from the point of view of good mutation strategies. Clearly, if the mutations are small, then the next sample has no chance to be relatively independent of the previous one, thus the correlation will be high. Interestingly, large mutations can also lead to highly correlated samples (figure 1). Suppose that the process has found a peak of the integrand. Having made a large perturbation, the importance of the tentative sample will be much lower, thus the chance of accepting it will also be low. The point on the peak remains to be the sample points for many steps, which is responsible for the high correlation.

Considering these, we can conlcude that a single, constant mutation strategy cannot provide an effective algorithm. Instead we need a mutation strategy that is generally large but gets smaller around the peaks of the integrand. The original Metropolis light transport algorithm proposes the random combination of several strategies, each of them is tailored for a particular type of light transfer [11]. The construction of a single strategy requires care and usually involves the

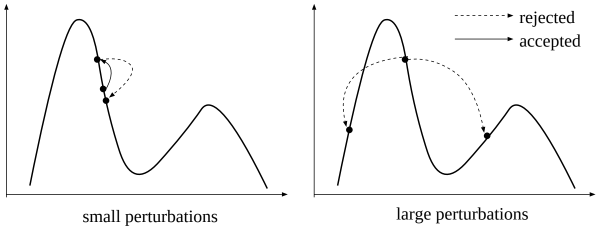


Figure 1: *Both large and small mutations can result in large correlation*

tuning of a lot of parameters. Having established the basic idea, the determination of the tentative transition probabilities from both directions poses mathematical problems, not to mention the implementation difficulties of these complicated path manipulation techniques. If the randomly selected mutation cannot generate an important tentative sample, then it will be rejected by the Metropolis sample, and the next random selection possibly finds another mutation type and sample. However, the badly selected and thus rejected perturbations also increase the computation time and increase the correlation.

2. Finding a good space for making mutations

The conclusions of the previous section immediately lead to the following question. Is it possible to adapt the mutation strategy itself, and when we are around the peak and thus anticipate that larger perturbations will be rejected, then the perturbation gets automatically smaller? At the first glance, the answer is negative since to construct the tentative transition probabilites explicite knowledge of the importance function would be needed, which is not available. However, if the domain and consequently the integrand is transformed in a way that the integrand is much smoother, the number of rejections can greatly be reduced (note that the rejection probability is proportional to the ratio of importances). Such transformation would obviously expand the domain where the original integrand is large and shrink it when the original integrand is small. If constant size perturbations are used in the transformed domain, they would correspond to larger steps where the original integrand is large and smaller steps where it is small.

Furthermore, such transformation is for free if the random paths are built using importance sampling, e.g. BRDF sampling, lightsource sampling and Russian roulette. Note that when importance sampling or Russian roulette are applied, we start with uniformly distributed pseudo-random numbers generated in the unit interval and transform them to obtain the directions, points, termination, etc. of the particular path. Those random numbers from which a complete path is generated can be considered as points in a high-dimensional unit cube. Let us call this cube as the *primary sample space*. Importance sampling means that transforming the points uniformly distributed in the cube we place more samples in path regions where the path are expected to have significant contribution. This is exactly what we need. Thus the perturbation strategy that works in the primary sample space will adapt to the properties of the integral and reduces the perturbation size where the integral is large.

The contribution to a pixel is an integral over the space of the possible light paths $\mathcal{P}$:

$$\Phi = \int_{\mathcal{P}} W(\mathbf{z}) \cdot L(\mathbf{z})\; d\mathbf{z}$$

In order to obtain the samples of $\mathbf{z}$ from random samples in the unit cube, the path space $\mathcal{P}$ is transformed to the unit cube $\mathcal{U}$ by a transformation $\mathbf{u} = T(\mathbf{z})$. Thus the pixel contribution is:

$$\Phi = \int_{\mathcal{U}} W(T^{-1}(\mathbf{u})) \cdot L(T^{-1}(\mathbf{u})) \cdot \left| \frac{dT^{-1}(\mathbf{u})}{d\mathbf{u}} \right| \; d\mathbf{u},$$

where

$$\left| \frac{dT^{-1}(\mathbf{u})}{d\mathbf{u}} \right| = \frac{1}{t(\mathbf{u})}$$

is the Jacobi determinant of the inverse mapping. Intuitively, the Jacobi determinant expresses the local expansion between two corresponding spaces. Considering this, the meaning of $t(\mathbf{u})$ is the density of the sample points in the neighborhood of $\mathbf{z} = T^{-1}(\mathbf{u})$. If $\mathbf{u}$ is a uniformly distributed random variable, then the probability density of $\mathbf{z} = T^{-1}(\mathbf{u})$ will be $t(\mathbf{u})$. If importance sampling is used, then the Jacobi determinant is approximately inversely proportional to the pixel independent part of the original integrand L, placing dense samples where L is large and also making the new integrand

$$L^*(\mathbf{u}) = L(T^{-1}(\mathbf{u})) \cdot \left| \frac{dT^{-1}(\mathbf{u})}{d\mathbf{u}} \right| = \frac{L(T^{-1}(\mathbf{u}))}{t(\mathbf{u})}$$

relatively flat. The importance function gets similarly flatter:

$$I^*(\mathbf{u}) = \frac{I(T^{-1}(\mathbf{u}))}{t(\mathbf{u})}.$$

Let us take an example to show how the pertubation in the primary sample space affects the path. The example can also be followed in figure 2.

Assume that we use bi-directional path tracing to find a light-path (the proposed algorithm can work with any random walk algorithm). Using two pseudo-random numbers u_1, u_2 a random point on the window is found and a ray is traced through this point, which finds surface point $\vec{x}_1$. At $\vec{x}_1$ we take u_3 to randomly select a BRDF of the elementary BRDFs composing the reflection function at $\vec{x}_1$ and to decide whether or not the walk has to be terminated according to Russian roulette. Assume that we decided to continue

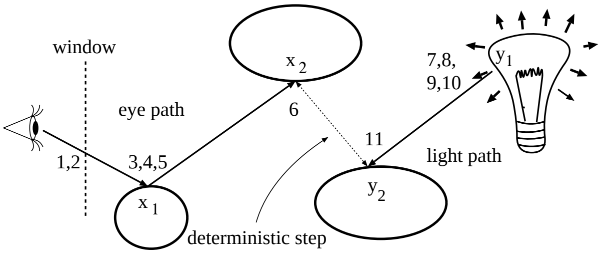


Figure 2: *A sample path*

the walk, thus we use another two random numbers u_4, u_5 to sample the direction with a density that is approximately proportional to the selected elementary BRDF. The obtained direction with starting point $\vec{x}_1$ define a new ray that is traced to find the new point $\vec{x}_2$. Here we again decide on the termination using u_6. Suppose that now, the random number u_6 and the albedo at $\vec{x}_2$ are such that the walk is terminated. In order to make the shooting part of the bi-directional path, a point and a direction is obtained on the surface of the light source. Sampling a $\vec{y}_1$ point can be done with numbers u_7, u_8, similarly, the direction sample is obtained with u_9, u_{10}. This again defines a ray which hits a new point $\vec{y}_2$. Here a random decision is made with pseudo-random number u_{11}. Suppose that the shooting part is terminated here, and finally the end points of the eye walk and the light walk are connected. If the end points are not occluded, we have established the following light path: $\mathbf{z} = (\vec{y}_1, \vec{y}_2, \vec{x}_2, \vec{x}_1, \vec{eye})$. Clearly, this path is unambigously defined by the vector $\mathbf{u} = (u_1, \ldots, u_{11})$. The mapping $\mathbf{z} = T^{-1}(\mathbf{u})$ is defined by the bi-directional ray-tracing scheme that involves BRDF sampling, Russian roulette and light source sampling.

Let us now perturb the elements of the vector $\mathbf{u} = (u_1, \ldots, u_{11})$ a little. Decision parameters u_3, u_6 and u_{11} are used to terminate the walks and to select from elementary BRDFs. If the perturbations of these values fit in the range allowed by the albedos, then the new values lead to the same decisions, thus the structure of the path is not altered (for example, the new path will also connect 5 points). However, when one of the new value steps over the albedo boundaries, then the remaining part of the sub-path is cut or the previously terminated walk now has to be continued. The other parameters responsible for changing the directions of steps of the walk.

3. Large steps

The next problem that needs special care comes from the regions of zero importance. The Markovian process used in Metropolis algorithm should be ergodic, i.e. all samples of non-zero importance should be generated with positive probability. In the global illumination setting it is very likely that light-paths of non-zero importance form islands in the path space. If the mutations are not big enough to jump from one island to the other, then the ergodicity condition cannot be met. In order to solve this problem, we include completely independent steps in the algorithm, which obtain the tentative sample without considering what the actual state is. These independent steps are called *large steps*. Large steps have three different merits. These large steps generate any point non-zero point with positive probability, thus the ergodicity, i.e. the problem of zero importance regions, is solved. If the large steps is accepted, then the Metropolis process start from a new random seed, which significantly reduces the start-up bias error. Finally, the probability density of the tentative samples obtained with large steps is known, which allows for their sophisticated secondary unitization according to the concept of multiple importance sampling.

Let us consider how the tentative sample worth generating in these large steps. In order to reduce correlation, the acceptance probability should be maximized. Note that the tentative probability depends now only on the target state, that is $T(\mathbf{z}_i \to \mathbf{z}_t) = T(\mathbf{z}_t)$. Considering this, the acceptance probability is:

$$a(\mathbf{z}_i \to \mathbf{z}_t) = \frac{\mathcal{I}(\mathbf{z}_t) \cdot T(\mathbf{z}_i)}{\mathcal{I}(\mathbf{z}_i) \cdot T(\mathbf{z}_t)}$$

This probability can be set to 1 if $T(\mathbf{z})$ is proportional to the importance. However, this would require the explicit knowledge of the importance function. If the large steps are examined in the primary sample space the acceptance probability has the following form:

$$a(\mathbf{u}_i \to \mathbf{u}_t) = \frac{\mathcal{I}^*(\mathbf{u}_t) \cdot T(\mathbf{u}_i)}{\mathcal{I}^*(\mathbf{u}_i) \cdot T(\mathbf{u}_t)}.$$

Due to importance sampling $\mathcal{I}^*(\mathbf{u}_t)$ is usually quite flat. More precisely, it is as flat as BRDF sampling, light source sampling and Russian roulette can describe the importance of a path. Using this approximation that the transformed importance is constant:

$$a(\mathbf{u}_i \to \mathbf{u}_t) \approx \frac{T(\mathbf{u}_i)}{T(\mathbf{u}_t)}.$$

The acceptance probability will be 1 if the mutation function of the large steps are realized with a uniform probability.

4. Designing mutation strategies in the infinite dimensional cube: lazy evaluation

So far, we neglected the fact that the primary sample space, i.e. our cube where the sample points are perturbed is infinite dimensional, where it is impossible to unambigously define a point with finite numbers. However, this is not a problem if the coordinates of each point are evaluated in a lazy way. Note that a point unambigously defines a bi-directional path according to BRDF sampling, Russian-roulette and light source sampling, however, only the first few coordinates are

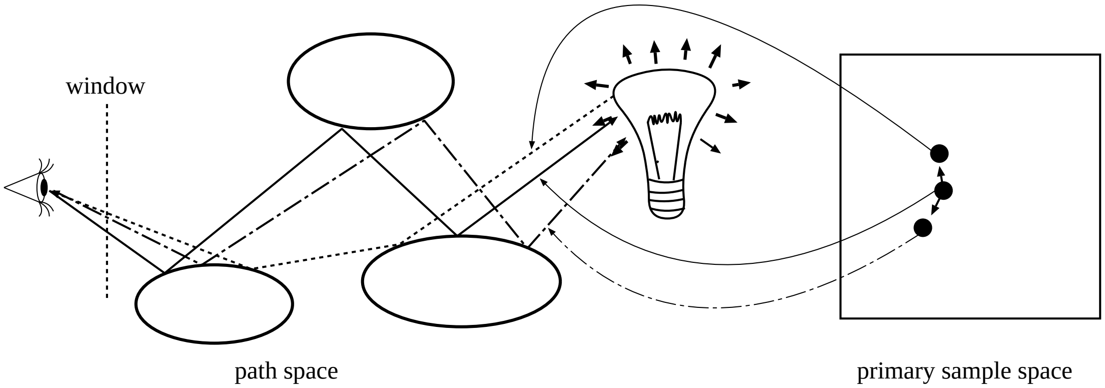


Figure 3: *The correspondance between the mutations in the primary sample space and in the path space*

used until the paths are terminated according to Russian-roulette. Let us evaluate, store and perturb only those coordinates of the current point, which has been needed by the longest path happened so far. If it turns out that the current path is longer than the longest path encountered so far and thus we need new coordinates, then we have repeat the life history just for these new coordinates. We have to go back to its last use or if it has not used before, to the last large step since the last step generated all coordinates randomly and independently of the former states (in this sense accepted large steps are those critical time instances beyond which we do not have to remember). If the point has never been used, then its initial value is obtained randomly as the large step would have obtained it. Then the history happened since the last use is played again, i.e. the point is perturbed by the times of the mutated perturbations happened since the last use. Note that the number of accepted mutations counts since rejected mutations do not affect the future.

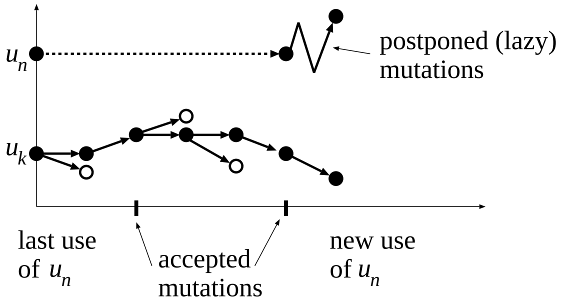


Figure 4: *Lazy evaluation of the coordinates*

The implementation of this scene is quite straigtforward. Let us define a counter for the global time of the process which counts the number of accepted mutations and call it *succ_mutation*. Each coordinate is associated with a local time-stamp called *last_modify* that stores the global time when it was modified. The time of the last accepted large step is also stored in variables *large_step_time*. When a new coordinate is needed, first it is checked whether or not this coordinate has been used before. If it has not been used, it is initialized as a random number and its time stamp is set to *large_step_time*. If it has already been used, the value of the coordinate is associated with its state at time *last_modify*. In both cases, the coordinate is perturbed by *succ_mutation* − *last_modify* times.

5. Utilization of rejected samples

The original Metropolis light transport algorithm makes use of rejected samples by replacing the random variable of the random acceptance by its mean, which is a common variance reduction technique and multiplies the original and tentative values by the rejection and acceptance probabilities. However, large steps allow us to do it even better. Merging large steps can also be viewed as a combination of two sampling techniques, also called multiple importance sampling [10]. Multiple importance sampling is capable to combine two methods in a way that the strengths of the methods are preserved.

According to the results of multiple importance sampling [10], if we use two sampling techniques, and the first sampling technique generates M_1 number of $\mathbf{u}$ samples with probability $p_1(\mathbf{u})$, while the second sampling technique M_2 samples with probability $p_2(\mathbf{u})$, then a quasi-optimal combination would weight the $F(\mathbf{u})/p_1(\mathbf{u})$ samples of the first method by

$$w_1(\mathbf{u}) = \frac{M_1 p_1(\mathbf{u})}{M_1 p_1(\mathbf{u}) + M_2 p_2(\mathbf{u})}$$

and would weight the $F(\mathbf{u})/p_2(\mathbf{u})$ samples of the second method by

$$w_2(\mathbf{u}) = \frac{M_2 p_2(\mathbf{u})}{M_1 p_1(\mathbf{u}) + M_2 p_2(\mathbf{u})}.$$

Metropolis generates a sequence of tentative samples, but sometimes rejects them and replaces them with the previous sample before using it in the quadrature. Applying mean value substitution, no matter whether or not the sample is rejected, both the original and the new values are substituted into the integral quadrature weighting their contribution with their acceptance probabilites. A secondary reuse of these samples can be found, if we consider their sequence coming from a different sampling technique and use multiple importance sampling to combine its results with that of the Metropolis sampling. The combination requires the knowledge of the generation probabilities of both methods, which is generally not easy to find. However, for the uncorrelated samples of the large steps this probability is trivial. Let us thus consider the sequence of all uncorrelated tentative samples as a separate sampling technique, and let us determine the probability of generating a given sample by the Metropolis method and by this alternative sequence of uncorrelated samples.

Metropolis sampling generates a sample with the probability that is proportional to the importance:

$$p_1(\mathbf{u}) = \frac{\mathcal{I}(\mathbf{u})}{b}.$$

For the large steps, the uniform point selection is used, thus:

$$p_2(\mathbf{u}) = 1.$$

If the probability of large steps is p_{large}, then the number of samples of the second method is expected to be p_{large} times the number of samples of the first method, thus $M_2/M_1 = p_{large}$.

Finally we should take into account that due to the mean value substitution the new samples of the Metropolis will be weighted by a, while the old samples are also used with weight $1-a$. Putting these altogether, the weight of a new Metropolis sample is:

$$w_1^{new}(\mathbf{u}) = \frac{a \cdot I(\mathbf{u})/b}{I(\mathbf{u})/b + p_{large}}.$$

The weight of the old sample is:

$$w_1^{old}(\mathbf{u}) = \frac{(1-a) \cdot I(\mathbf{u})/b}{I(\mathbf{u})/b + p_{large}}.$$

The weight of the uncorrelated samples is:

$$w_2(\mathbf{u}) = \frac{p_{large}}{I(\mathbf{u})/b + p_{large}}.$$

Thus when a large step is made, the weight of the new sample is derived from both sampling strategies:

$$w_1^{new}(\mathbf{u}) \cdot \frac{F(\mathbf{u})}{p_1(\mathbf{u})} + w_2(\mathbf{u}) \cdot \frac{F(\mathbf{u})}{p_1(\mathbf{u})} = \frac{a+1}{I(\mathbf{u})/b + p_{large}} \cdot F(\mathbf{u}).$$

Small steps can stem only from the Metropolis method, thus for small steps the weight is:

$$w_1^{new}(\mathbf{u}) \cdot \frac{F(\mathbf{u})}{p_1(\mathbf{u})} = \frac{a}{I(\mathbf{u})/b + p_{large}} \cdot F(\mathbf{u}).$$

Note that Metropolis method is good in generating bright image sections while poor at dark regions since the number of samples is proportional to the luminance. The perceptual error, on the other hand, depends on the relative error, which is thus small at bright and large at dark image areas. Conventional random walk uses the same number of samples in all pixels, thus the perceptual error will be roughly uniform. The proposed combination can thus improve dark image areas.

5.1. Theoretical analysis of the power of large steps to reduce the start-up bias

In order to theoretically evaluate the start-up bias, let us examine a simplified, 1-dimensional case when the importance is constant, thus the transition proposed by the tentative transition function is always accepted.

In this case, the probability density $p_n(x)$ in the equilibrium is constant. The question is how quickly the Metropolis method approaches to this constant density (figure 5).

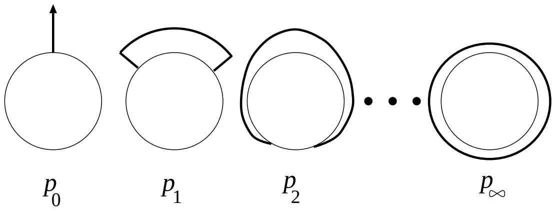


Figure 5: *Evaluation of the uniform distribution*

Metropolis method can generate samples following a given probability density in a closed interval. Since random mutations may result in points that are outside the closed interval, the boundaries should be handled in a special way.

If the variable of an integrand denotes “angle of direction”, then the interval can be assumed to be “circular”, that is, the external points close to one boundary are equivalent to the internal points of the other boundary. Using this assumption, let us suppose that the domain of the integration is $[-\pi, \pi]$ and the integrand is periodic with 2π.

In [9], the analysis of error of the distribution resulted in the following formula:

$$||p_n - p_\infty||_2 = \sqrt{\sum_{k=-\infty,k\neq 0}^{\infty} |P^*(k)|^{2n}} \qquad (5)$$

where $P^*(k)$ is the Fourier transform of the perturbation function $P(x)$. This perturbation function defines the probability of generating point x as the perturbation of point zero. If large steps are also implemented the perturbation function is a composition of the large steps and the perturbation function of the small steps:

$$P(x) = p_{large} \cdot \frac{1}{2\pi} + (1 - p_{large}) \cdot S(x)$$

where $S(x)$ is the perturbation function of the small steps. Substituting this into equation 5 we obtain:

$$||p_n - p_\infty||_2 = (1 - p_{large})^n \cdot \sqrt{\sum_{k=-\infty,k\neq 0}^{\infty} |S^*(k)|^{2n}}$$

Note that even with lower p_{large} values the start-up bias quickly disappears.

Let the perturbation be the selection of a point following uniform distribution from an interval of size Δ centered by the current point. Formally the transition probability is

$$S(x \to y) = \begin{cases} 1/\Delta & \text{if } |x-y| < \Delta, \\ 0 & \text{otherwise.} \end{cases} \qquad (6)$$

The Fourier transform of this function is

$$S^*(k) = \frac{\sin k\pi\Delta}{k\pi\Delta} \qquad (7)$$

which can be rather big even for large k values. This formula, together with equation (5) allows to generate the graph of the start-up errors for different sample numbers and for different perturbation size (figure 6). Note that the probability density is not accurate for many iterations if the perturbation size is small compared to the size of the domain. This situation gets just worse for higher dimensions.

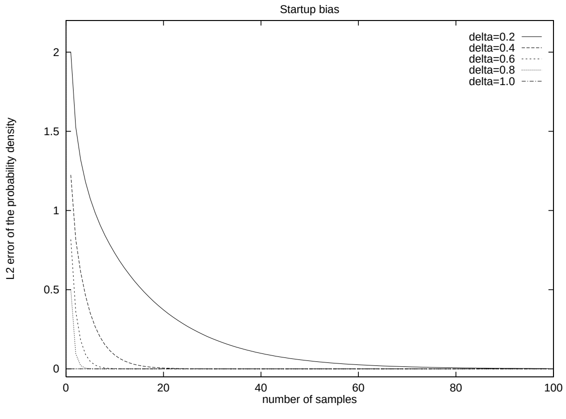


Figure 6: *Start-up error for different perturbation size* Δ

5.2. Numerical simulation

In order to demonstrate the proposed transformation, we took a simple integrand of figure 7. The function is piecewise linear its mimimum is 0.01 and its two peaks have 4 and 3 values, respectively. Such integrand is typical in rendering where the peaks represent importance light path sets and the minimum just the diminishing multiple reflections. Note that the also piece-wise linear importance function only mimics the original integrand, it is flatter. Again, this corresponds to the fact that in rendering the luminance is used as importance which averages the radiances on different wavelengths. The probability density is even less accurate, and consists of piece-wise constant section. Recall, that these probability densities are constructed in a way to allow importance sampling, just they should be analytically integrable. This requirement contradicts the accurate approximation of the importance function.

The right of figure 7 also shows the transformed integrand and importance function.

To simulate the fact that in rendering many integrals are evaluated simultaneously, let us suppose that the unit domain is decomposed into $N = 10$ intervals and we are interested in the integral values in each interval. The error measure will be the RMS error between the real and approximated integrals in all intervals. The functions in the original and transformed spaces are shown in figure 7.

5.3. Comparing the integration in the original and in the primary sample space

Note that in figure 7 the transformed function is much flatter at high importance regions, thus here the mutations are accepted with higher probability than in the original space. This change significantly increases and acceptance probability if the perturbations are large enough to step out from the support of the peaks. Indeed, the average acceptance probability of the mutations increased from 18% to 67% due to the transformation when we used $[-0.5, 0.5]$ perturbation interval. Higher acceptance probability, on the other hand, reduces the start-up bias and also the variance of the algorithm as shown in figure 8. However, when the perturbation size is small, the gain received from the transformation becomes negligible. For example, the average acceptance probability increased from 62% to 84% when the interval of perturbation was $[-0.05, 0.05]$. Thus we can conclude that large mutations can benefit from the transformation.

5.4. Performance evaluation of the large steps

In order the evaluate the efficiency of large steps, we took three different settings. This first one applied no large steps

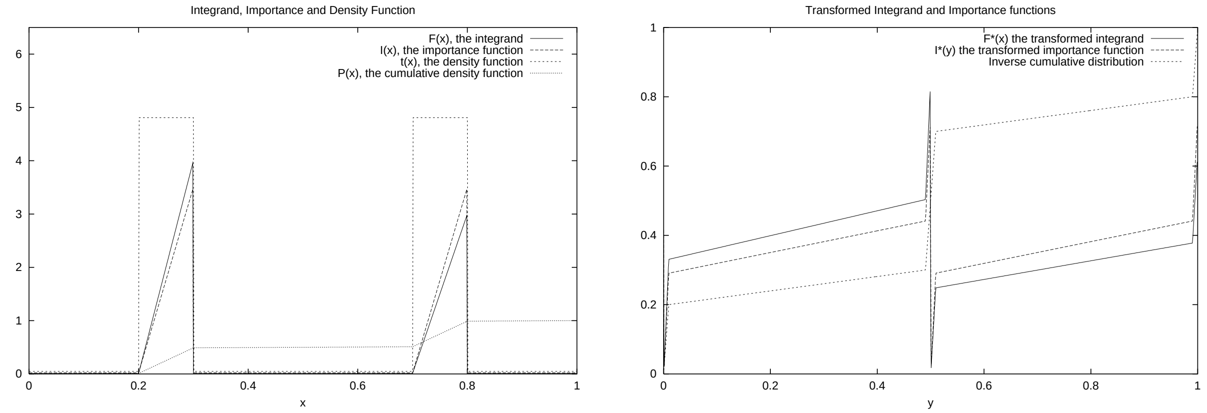


Figure 7: *Functions used for testing, before (left) and after (right) the proposed transformation*

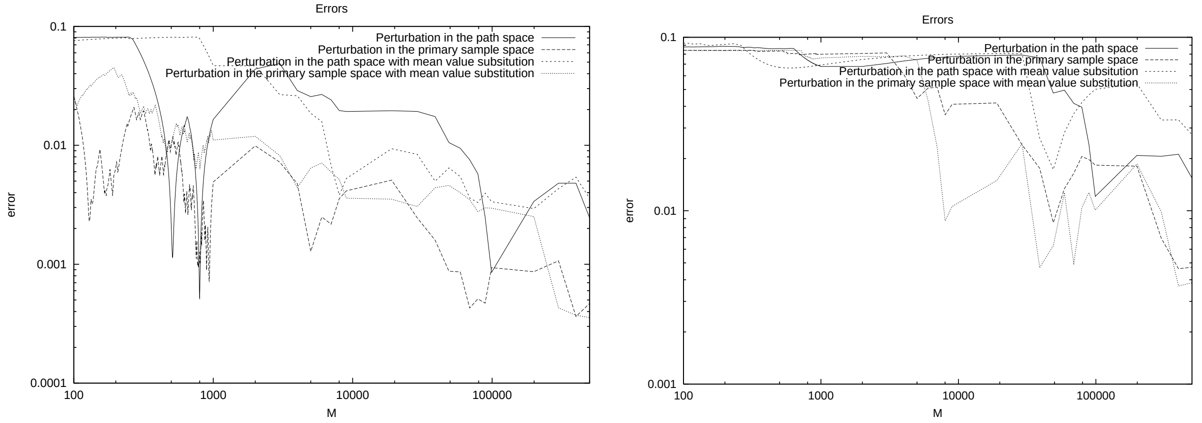


Figure 8: *RMS error curves of the Metropolis methods working in the original and the transformed spaces for large perturbations in* $[-0.5, 0.5]$ *(left), and small perturbations in* $[-0.05, 0.05]$ *(right)*

and both the original and the tentative samples were used in the quadrature. This first case corresponds to the original metropolis light transport. In the second example, we allowed large steps but still used the original combination of old and tentative samples. Finally, the large steps were taken into consideration with the proposed multiple importance sampling concept. The results are shown in the left of figure 9.

Note that the large steps improved the convergence but the benefits of the multiple importance sampling is not obvious yet. The reason is that we used the RMS error, which is not really good at expressing the image differences. For example, if a dark pixel has some σ^2 variance, then it would mean the same RMS error as if a bright pixel had it. However, this variance is much more noticable on the dark pixel.

To overcome this problem, a perceptual error measure will be used. For instance, according to Weber law, the human eye is sensitive to relative rather than absolute errors. Thus let us compute the relative error in each slot and sum up the errors. The right of figure 9 has been made with this metric, which clearly emphasizes the benefits of multiple importance sampling.

If we consider relative errors due to their superiority in perceptual sense, the complementary advantages of Metropolis sampling and normal random walks can be identified. Note that Metropolis will use more samples at high intensity regions and less at dim regions. For instance, according to equation 4, the relative error of a pixel will be proportional to $\mathcal{I}^{-1/2}$ which is good for bright and bad for dark pixels. Random walks, on the other hand, provide pixel values with approximately the same relative error if they use the same number of samples in each pixel. Thus random

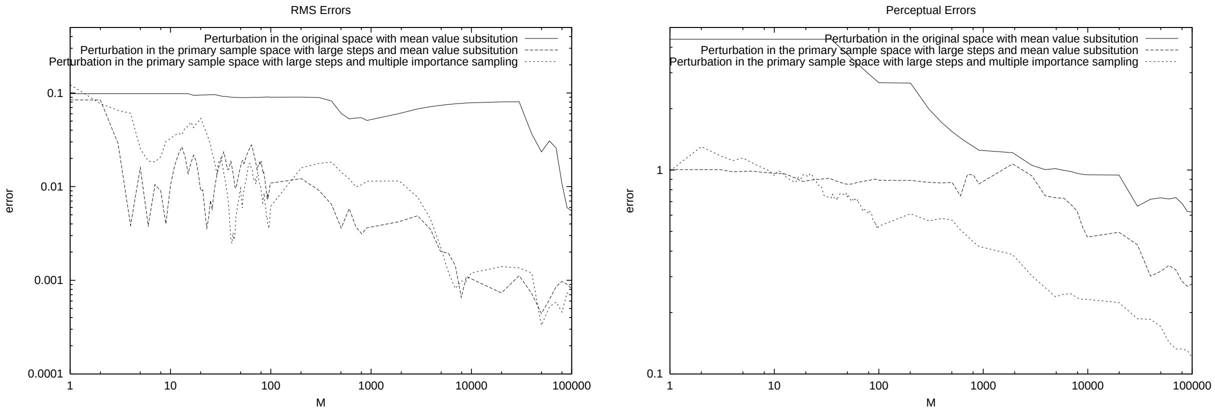


Figure 9: *Error curves of the Metropolis method in the original domain with and without large steps, supposing RMS measure (left), and using a perceptual error measuere (right)*

walks are better for darker regions. The large steps also have a drawback, that they may decrease the average acceptance ratio. However, working in the primary sample space solves this problem as we concluded in the previous section. From the combination of the two techniques, we expect a method that is as good as Metropolis sampling in bright sections, but can also handle dark regions as well as random walks.

The relative importance of the two techniques can be controlled by the probability of the large steps (p_{large}). In order to set this probability, several contradicting criteria should be taken into consideration. For example, if we increase the large step probability, then the start up bias will quickly vanish and the dark regions will be better. However, large probability values also increase the correlation of samples. For instance, the average acceptance probability is decreased from 87% to 50% if the large step probability increased from 0.1 to 0.9.

We took different large step probabilities and compared the error curves in figure 10. Note that the given integrand the optimal probability is about 0.7. This high value is due to the fact that the integrand is relatively flat in the primary sampling space. The question of the optimal setting of the large step probability is not solved yet for the general case. According to our experiences, it is worth increasing until the average acceptance probability does not drop significantly.

6. Performance evaluations in the global illumination setting

7. Conclusions

This paper presented a new mutation method for Metropolis light transport algorithm. The mutations are computed in the primary sample space where usually the pseudo-random numbers are obtained. If importance sampling is used, this strategy smoothes the integrand, and thus increases the average acceptance probability and thus reduces the correlation. The other main advantage of this approach is that it does not require sophisticated techniques and dubious tricks to set parameters. In this way, any random walk algorithm can be equippped with Metropolis sampler, just the random number generation and the averaging scheme should be modified. We also proposed the application of large steps to include independent samples in the sequence. These large, independent steps have threefold benefits. They reduce the start-up bias, guarantee the ergodicity and allow a more sophisticated reuse of tentative samples in the integral quadrature, based on the concepts of multiple importance sampling.

8. Acknowledgements

This work has been supported by the National Scientific Research Fund (OTKA ref. No.: T029135), the Eötvös Foundation and the IKTA-00101/2000 project.

9. Conclusions

References

1. M. Ashikhmin, S. Premoze, P. Shirley, and B. Smits. A variance analysis of the metropolis light transport algorithm. *Computer and Graphics*, 2001.

2. I. Deák. *Random Number Generators and Simulation*. Akadémia Kiadó, Budapest, 1989.

3. Ph. Dutre, E. Lafortune, and Y. D. Willems. Monte Carlo light tracing with direct computation of pixel intensities. In *Compugraphics '93*, pages 128–137, Alvor, 1993.

4. E. Lafortune and Y. D. Willems. Bi-directional path-tracing. In *Compugraphics '93*, pages 145–153, Alvor, 1993.

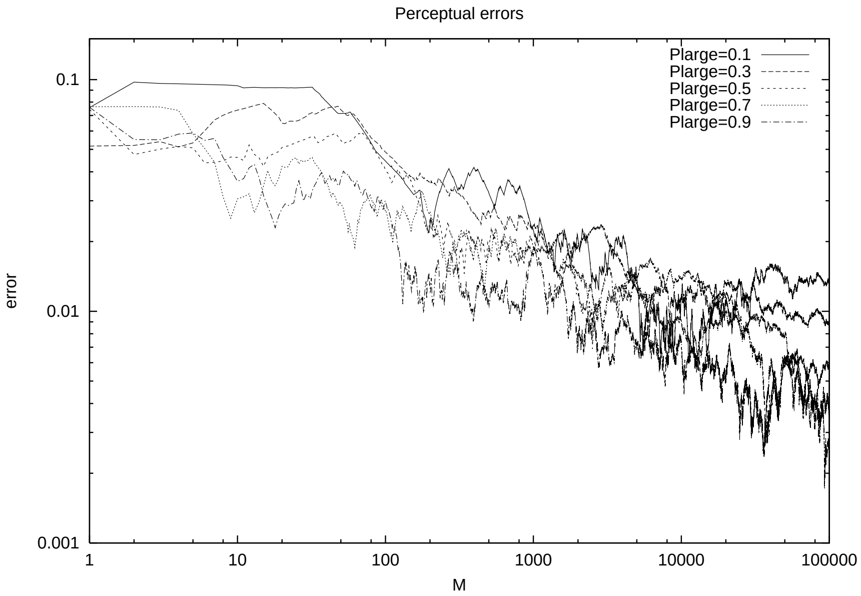


Figure 10: *Perceptual error curves using different p_{large} large step probabilities*

5. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091, 1953.

6. Alfréd Rényi. *Wahrscheinlichkeitsrechnung*. VEB Deutscher Verlag der Wissenschaften, Berlin, 1962.

7. P. Shirley, C. Wang, and K. Zimmerman. Monte Carlo techniques for direct lighting calculations. *ACM Transactions on Graphics*, 15(1):1–36, 1996.

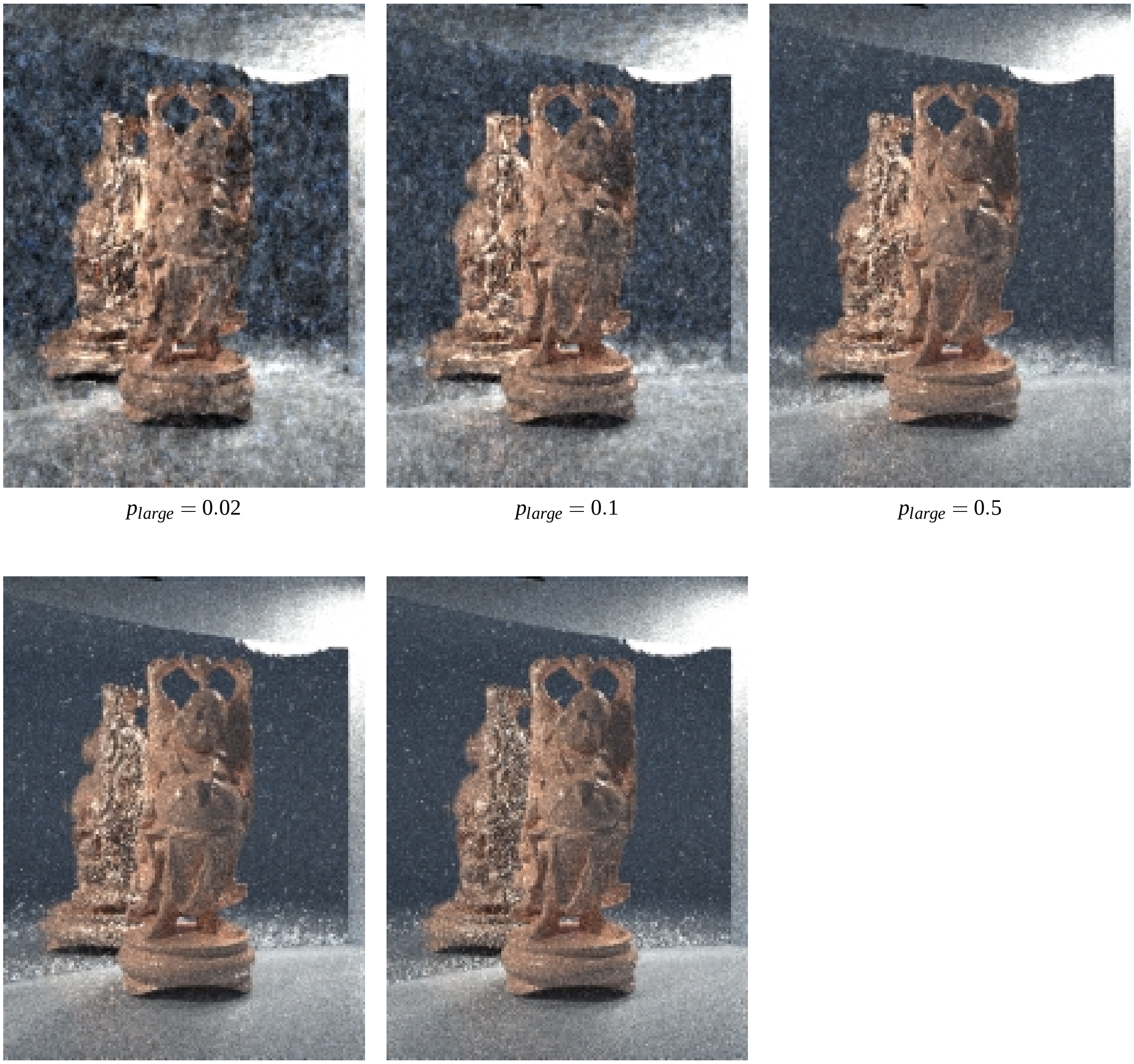
8. I. Sobol. *Die Monte-Carlo Methode*. Deutscher Verlag der Wissenschaften, 1991.

9. L. Szirmay-Kalos, P. Dornbach, and W. Purgathofer. On the start-up bias problem of metropolis sampling. In *Winter School of Computer Graphics '99*, pages 273–280, Plzen, Czech Republic, 1999.

10. E. Veach and L. Guibas. Optimally combining sampling techniques for Monte Carlo rendering. In *Rendering Techniques '94*, pages 147–162, 1994.

11. E. Veach and L. Guibas. Metropolis light transport. *Computer Graphics (SIGGRAPH '97 Proceedings)*, pages 65–76, 1997.

12. G. J. Ward. The RADIANCE lighting simulation and rendering system. *Computer Graphics*, 28(4):459–472, 1994.



$p_{large} = 0.02$

$p_{large} = 0.1$

$p_{large} = 0.5$

$p_{large} = 0.9$

$p_{large} = 0.98$

Figure 11: *Images rendered by the proposed algorithm using multiple importance sampling with different large step probabilities. We used 30 mutations per pixel.*

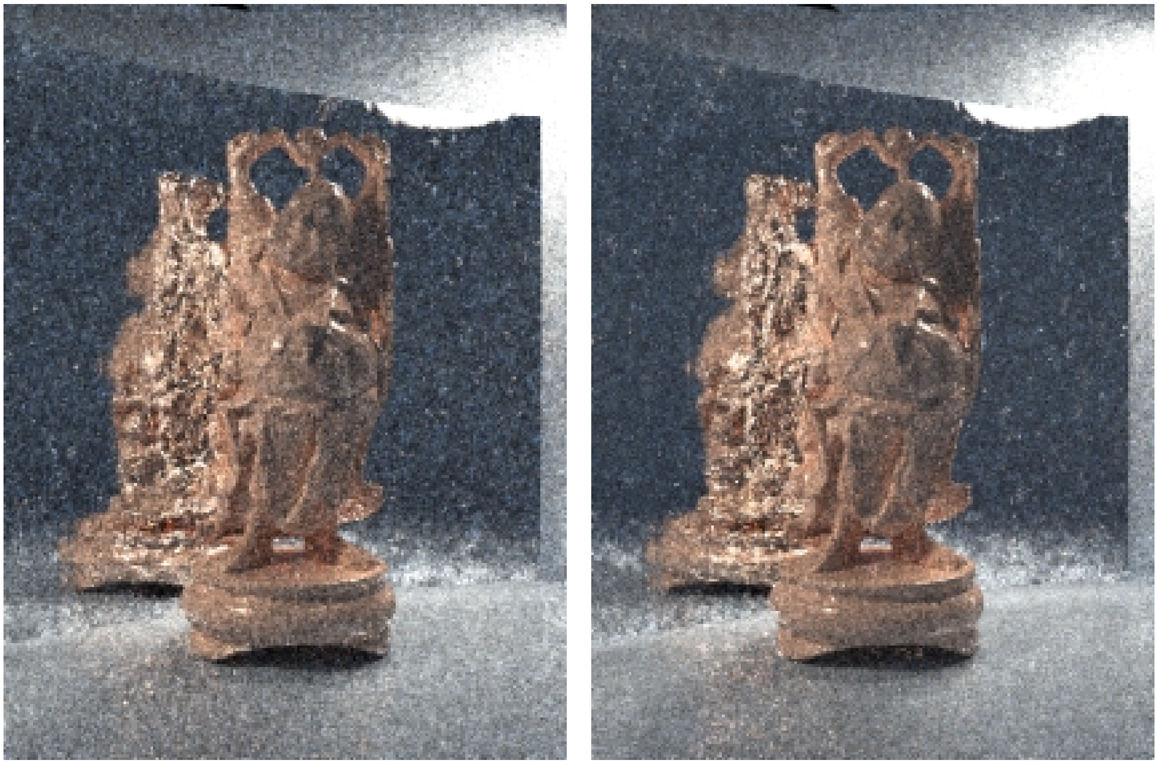


Figure 12: *Images rendered without (left) and with (right) multiple importance sampling. We used 30 mutations per pixel.*