mathematics

MDPI

Article

Efficiency Investigation of Langevin Monte Carlo Ray Tracing

Sergey Ershov [1], Vladimir Frolov [1,2], Alexander Nikolaev [2], Vladimir Galaktionov [1] and Alexey Voloboy [1,*]

[1] Department of Computer Graphics and Computational Optics, Keldysh Institute of Applied Mathematics RAS, 125047 Moscow, Russia; ersh@gin.keldysh.ru (S.E.); vladimir.frolov@graphics.cs.msu.ru (V.F.); vlgal@gin.keldysh.ru (V.G.)

[2] Laboratory of Computer Graphics and Multimedia, Faculty of Computational Mathematics and Cybernetics, Moscow State University, 119991 Moscow, Russia; alexander.nikolaev@graphics.cs.msu.ru

* Correspondence: voloboy@gin.keldysh.ru

Abstract: The main computationally expensive task of realistic computer graphics is the calculation of global illumination. Currently, most of the lighting simulation methods are based on various types of Monte Carlo ray tracing. One of them, the Langevin Monte Carlo ray tracing, generates samples using the time series of a system of the Langevin dynamics. The method seems to be very promising for calculating the global illumination. However, it remains poorly studied, while its analysis could significantly speed up the calculations without losing the quality of the result. In our work, we analyzed the most computationally expensive operations of this method and also conducted the computational experiments demonstrating the contribution of a particular operation to the convergence speed. One of our main conclusions is that the computationally expensive drift term can be dropped because it does not improve convergence. Another important conclution is that the preconditioning matrix makes the greatest contribution to the improvement of convergence. At the same time, calculation of this matrix is not so expensive, because it does not require calculating the gradient of the potential. The results of our study allow to significantly speed up the method.

Keywords: global illumination; monte carloray tracing; Markov chain; Langevin dynamics; preconditioning matrix

MSC: 65D18; 65C05; 65C40

check for updates

Citation: Ershov, S.; Frolov, V.; Nikolaev, A.; Galaktionov, V.; Voloboy, A. Efficiency Investigation of Langevin Monte Carlo Ray Tracing. *Mathematics* **2024**, *12*, 3437. https://doi.org/10.3390/math12213437

Academic Editors: Xiaoqiang Zhu and Lihua You

Received: 8 October 2024
Revised: 29 October 2024
Accepted: 1 November 2024
Published: 3 November 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Modern realistic computer graphics is based on the physically correct simulation of the propagation of light in various environments. Calculation of the global illumination, i.e., the distribution of light in a virtual scene, is one of the main computationally expensive tasks when generating a realistic image. It has to take into account the multiple scattering of light, all kinds of interaction with objects in the scene, and the propagation of light through various media. The calculation is based on the rendering equation [1], which includes the integral over the angular distribution of light coming from all possible directions. There are several approaches to the solution of this problem, but currently, most of them are based on various types of the Monte Carlo ray tracing (MCRT). In realistic rendering problems, the result of the work of MCRT algorithms is the accumulated average of the function w_p over all ray paths running through the given pixel p. This average the estimates pixel luminance L_p and is taken over the probability distribution proportional to the importance function I [2].

$$L_p \sim \langle w_p \rangle_I. \tag{1}$$

The review [3] considers several variants of Monte Carlo integration applied to the problem of lighting simulation and concludes that the Markov Chain Monte Carlo (MCMC)

methods have advantages. These methods can be regarded as an extension of the ordinary Monte Carlo method. Contrary to the ordinary Monte Carlo where the samples are independent, in MCMC, they are correlated. This feature of MCMC allows to effectively utilize information about areas of high importance. One of the widely used variants of MCMC is the Metropolis–Hastings algorithm [4]. The main goal of all MCMC algorithms is to generate samples distributed with the density proportional to an arbitrary importance function. However, different types of MCMC achieve this in different ways [5].

The hybrid Monte Carlo methods (HMCs) [6] generate samples using time series of a dynamical system: Hamiltonian mechanics or the Brownian motion in viscous medium based on the Langevin equation. These methods look prospective for lighting simulation problems because they are universal and they have better convergence when the dimension of the integration space grows ($d^{5/4}$ for the HMC vs. d^2 for the MCMC, where d is the dimension [4]). The reason is making the proposal of the anisotropic transition based on the derivatives of the importance function. The HMC outperforms the MCMC because the isotropic transition proposal in the MCMC has a lower chance of remaining in the region of the high importance function. However, HMC methods are new, and many of their aspects remain uninvestigated.

The MALA (Metropolis-Adjusted Langevin Algorithm) method, based on the Langevin dynamics, had been proposed in [7,8]. At the moment, the work of [8] seems to be the last one in the application of hybrid Monte Carlo to the problem of illumination calculation. The MALA algorithm with a preconditioning matrix is promising, but it has not been fully studied in the mentioned works. We can formulate the following questions. First, the issue of dealing with obstacles which create "potential barriers" (discontinuities) of the importance function has not been investigated. In terms of lighting, these relate to the shadowing and masking by various scene objects. The gradient of potential cannot help to detect these barriers because the potential changes smoothly and slowly until it reaches some point where the potential suddenly jumps, while the derivatives predict only the smooth part of the change. An efficient preconditioning matrix must adapt to the barriers. Second, in [7], the Langevin equation is written in the form different from the original equation by dropping the divergence term. The effect of the divergence term on the MALA convergence under different conditions has remained unknown. Third, the diagonal form of the preconditioning matrix and the method for its calculation is proposed in [7]. However, the efficiency of this method drops as the dimension of the phase space increases, because it is impossible to make a good transition proposal in any direction. For example, a diagonal matrix cannot represent rotation.

Thus, there are three computationally expensive parts the MALA algorithm.

1. Calculation of the preconditioning matrix is expensive because this takes $O(d^3)$ operations and $O(d^2)$ of memory, where d is the dimension of phase space.
2. Calculation of the divergence term is a very expensive operation. But since the Hastings' rule provides exactly the desired stationary distribution for an arbitrary Markov chain, possibly this term can be dropped. However, this may worsen the convergence to a stationary distribution. For example, the probability of accepting a transition will decrease so that the generated sequence will contain many repeating points.
3. Similarly, we can try to drop the very term with the gradient of potential.

We investigate the influence of these three terms (matrix, divergence term and gradient term) in order to accelerate the MALA method applied to the illumination calculation problem.

Any implementation of MALA in practice will face a number of unobvious difficulties. First, the phase space must be chosen in some way. Almost all further calculations depend on this. Second, it is necessary to choose the time step. Third, one needs to calculate the probability densities of the forward and backward transitions for the Hastings rule. Finally, the preconditioning matrix needs to be calculated. These are the key points which influence the effeciency of MALA. We also consider them in detail in our study.

2. Related Works

Traditionally [2,9], a ray path is described by a set of its nodes (points of intersection of the ray with the surfaces of the scene) $\boldsymbol{X} = \{\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_m\}$. This representation is sometimes called the world path space. Another popular parameterization is through the set of directions of its segments $\boldsymbol{X} = \{\boldsymbol{v}_0, \boldsymbol{v}_1, ..., \boldsymbol{v}_m\}$ [10]. In this case, the coordinates of the nodes are not directly specified, but they are calculated. Other representations are also possible, for example, a primary path space, in which any point of a multidimensional unit cube corresponds to some ray path although maybe with a zero importance function. This group of methods includes, for example, Kelemen Metropolis Light Transport (MLT) [11] and Multiplexed Metropolis Light Transport (MMLT) [12].

When changing the representation to $\boldsymbol{Y} = \{\boldsymbol{y}_0, \boldsymbol{y}_1, ..., \boldsymbol{y}_m\}$, the density of the path distribution in the new variables will change from I to $I\left|\det \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{Y}}\right|$. It is natural to assume that this is the importance function in the new variables. Then, the image luminance can still be calculated as the average over a random sample of $\boldsymbol{Y}$ (1), i.e., paths, but now they must be generated with the density proportional to the new importance function $I\left|\det \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{Y}}\right|$. With a good choice of parameterization $\boldsymbol{Y}$, generating random paths with a new distribution may be easier and more efficient.

One of the most straightforward ways to choose parameterization is to use algorithms in the primary path space [12]. The second way is to work in the world path space formed by concatenating all coordinates of the path vertices in the world space (for example, the Veach Metropolis Light Transport [9] and Manifold Exploration Metropolis Light Transport [13,14] methods). The third way is to combine the first two. In the works [15–17], an invertible mapping between the world space and the primary path space is constructed, which allows to combine the advantages of both approaches. Many other kinds of the phase space can be devised. Practical aspects of the MCMC-based lighting calculation methods are well described in [5]. This work can be considered as a starting point for understanding them at a deeper level for the purpose of practical implementation.

Thus, most of the existing works applying MCMC to the problem of lighting simulation are based on the Metropolis–Hastings algorithm or even on the simple Metropolis algorithm with equiprobable forward and backward transitions [5]. On the other hand, the development of MCMC methods themselves has significantly advanced. The application of the Langevin dynamics is theoretically possible to any parameterization of the importance function. The work in which this has been accomplished so far [7] is based on MMLT. In our work, we use the parameterization of a path rather similar to that in the Half Space Light Transport method [10].

Hybrid Monte Carlo is a broad subset of MCMC [18–23]. These methods generate samples through trajectories of a dynamical system, such as Hamiltonian mechanics (Hamiltonian Monte Carlo) or Brownian motion in a viscous medium based on the Langevin equation (Langevin Monte Carlo). The basic idea of these methods is that the dynamical system can generate successful transition proposals and traverse the space more efficiently because it can move a point in the "allowed" directions of the phase space.

The Hamiltonian dynamic system is used in the Hessian–Hamiltonian Monte Carlo (HHMC) method [24,25]. A distinctive feature of the method is that the expensive simulation of the Hamiltonian dynamics is performed not for the potential function itself but on its quadratic approximation by the Taylor series. This is an important point, since the Hamiltonian equations for a quadratic function admit an analytical solution. Further, Hamiltonian Monte Carlo requires calculating the first (gradient) and second (Hessian) derivatives obtained using the automatic differentiation library [26]. The basis of the HHMC method is calculating the Hessian, which is used in the algorithm twice. The first time, it is used to calculate the approximation of the potential through the Taylor series. The second time, it is used to calculate the inverse mass matrix (an analog of the preconditioning matrix). In this case, the transition proposal itself is made in the form of a multidimensional elongated

Gaussian. Calculating the Hessian is expensive, and this is the main limiting factor of the method.

In [27], a relatively simple way to reduce the cost of the HHMC was proposed. At the beginning of each step, a normal transition proposal is made like in the Metropolis algorithm. If it is accepted, the algorithm continues its work as a normal Metropolis step. If the transition proposal is rejected, a new attempt is made using Hamiltonian Monte Carlo. The main difficulty in this approach is to correctly calculate the transition proposal probabilities.

To reduce the cost of the MALA algorithm step, several ideas are proposed: (1) to use a diagonal preconditioning matrix; (2) to accumulate the gradient and matrix over several steps of the algorithm like in the Adam optimization algorithm [28]; and (3) to use supervised or adaptive MCMC to correctly calculate the transition probabilities [29]. In the paper [8], two more methods are proposed: storing the gradient in memory and a hybrid accumulation and storing method to obtain a more accurate estimate of the averaged gradients. Thus, a kind of "integrated gradient" is proposed as the only source of information for forming a good transition proposal.

3. Langevin Monte Carlo Ray Tracing

3.1. Application of the Langevin Dynamics to MCRT

Langevin equations historically describe the Brownian motion of a small particle in a viscous medium under the action of a potential field (i.e., external force). In the limit of high viscosity, when inertia can be neglected, we arrive at the "overdumped Langevin dynamics". The original Brownian motion is just the 2nd Newton law under stochastic force $\boldsymbol{F}$:

$$\frac{d^2\boldsymbol{X}}{dt^2} = -\gamma\frac{d\boldsymbol{X}}{dt} + \frac{\boldsymbol{F}(\boldsymbol{X})}{m},$$

where t is the time, $\boldsymbol{X}$ is the position of the particle, m is its mass and γ is the viscosity. In the limit of high viscosity, one can derive an approximation using the well-known technique of the "singular perturbations" [30]. It is already a first-order ODE. Now, this singular perturbation is the inertia force $m\frac{d^2\boldsymbol{X}}{dt^2}$. This derivation can be seen in [31] which ends with a stochastic differential equation (1.45) sometimes called the "Smoluchowski equation" and sometimes called the "overdamped Langevin equation". For our purposes, it is convenient to write it in the "explicit form".

$$d\boldsymbol{X} = -\nabla U dt + \boldsymbol{\xi}(t)\sqrt{2dt}, \tag{2}$$

where $\boldsymbol{X}$ is the position of the particle, t is the time, U is the potential, $\boldsymbol{\xi}(t)$ is the Gaussian random process with zero mean and covariation $\langle\xi_i(t)\xi_j(t')\rangle = \delta_{i,j}\delta(t-t')$. While the original meaning of $\boldsymbol{X}$ is the 3D position of the particle, Equation (2) can be applied in a more general case. Then, $\boldsymbol{X}$ is a point in a multidimensional phase space.

The probability distribution associated with this equation obeys the Fokker–Planck equation (see, e.g., ODE) [30]. Now, this singular perturbation is the inertia force $m\frac{d^2\boldsymbol{X}}{dt^2}$. This derivation can be seen in [31] and exponentially converges to the stationary distribution $\rho(\boldsymbol{X}) = Ze^{-U(\boldsymbol{X})}$, where Z is the normalization factor. As a result, one can construct the dynamics for any (to be precise, under some rather weak limitations) probability density ρ by using the potential $U = -\log\rho$. This made this kind of dynamics popular in Monte Carlo methods as a generator of random samples with desired density [7].

One of the areas where Monte Carlo methods are widely used is ray tracing. Here, the state vector' $\boldsymbol{X}$ is the ray path, and the desired density of samples is the importance function. To be precise, the density is proportional to the importance function. They cannot be completely identical, since $\int \rho d\boldsymbol{X} = 1$, and the importance function is in general not normalized to 1.

This equation can be integrated numerically, for example, using the explicit Euler–Maruyama difference scheme. However, if the relief of the potential is complex, e.g., contains long narrow ravines, then a very small time step is required to reproduce the desired distribution with good accuracy. As a result, many steps have to be taken, which means a long computation time. When the potential is discontinuous across some boundaries, especially when the potential becomes infinite past them, an accurate numerical integration is yet more difficult.

The situation can be considerably improved by using the generalized Langevin equation [18,32]:

$$\begin{aligned} d\boldsymbol{X} &= \boldsymbol{a}(\boldsymbol{X})dt + \sqrt{2dt}\hat{T}\boldsymbol{\xi}(t), \\ a_i(\boldsymbol{X}) &= -\left(\hat{T}\hat{T}^*\nabla U\right)_i + \sum_j \frac{\partial(\hat{T}\hat{T}^*)_{i,j}}{\partial X_j}, \end{aligned} \tag{3}$$

where the "drift term" $\boldsymbol{a}(\boldsymbol{X})$ has the meaning of a "sliding force" acting on our particle. The expression $\sum_j \frac{\partial(\hat{T}\hat{T}^*)_{i,j}}{\partial X_j}$ is the divergence of the i-th row of $\hat{T}\hat{T}^*$, so we call it the divergence term; $\hat{T}^*$ is the transposed matrix. The matrix $\hat{T}(\boldsymbol{X})$ performs a local coordinate transformation (usually to remove strong anisotropy in the original phase space). It is often called the preconditioning matrix. Historically, it comes from optimization descent methods [33,34].

The idea behind this matrix is quite simple. For example, if the relief of potential has a narrow ravine, e.g., if the space is shrunk in some direction, then the isotropic Brownian term $\boldsymbol{\xi}(t)$ would nearly always kick the point out of the allowed region. So, the space must be stretched back along those directions to make the resulting potential more isotropic. In such a situation, the preconditioning matrix must orient (rotate) the space along the ravine and stretch it across the ravine. Without such a matrix, almost any transition proposal jumps out of the ravine of the potential and thus will be rejected by the Hastings procedure.

The simplest way of simulation of the Langevin dynamics is numerical integration using the explicit Euler–Maruyama difference scheme, where the final displacement of the point $\delta\boldsymbol{X}$ is calculated using the same formula as the differential:

$$\delta\boldsymbol{X} = \boldsymbol{a}(\boldsymbol{X})\delta t + \sqrt{2\delta t}\hat{T}\boldsymbol{\xi}(t), \tag{4}$$

where δt is the final time step.

Calculation of the new position of $\hat{\boldsymbol{X}}$ at $t + \delta t$ given the displacement $\delta\boldsymbol{X}$ is not easy; typically, it will not be $\hat{\boldsymbol{X}} = \boldsymbol{X} + \delta\boldsymbol{X}$. The problem is that in most cases, the trajectory is constrained to a manifold of significantly lower dimension. For example, if the ray path is described by a set of directions of its N segments, then this manifold is the direct product of N surfaces of unit spheres because the end of each direction vector is constrained to the unit sphere. For the world space, the nodes of the ray path lie on the surfaces of the scene and cannot leave them in the volume. Meanwhile, $\boldsymbol{X} + \delta\boldsymbol{X}$ belongs to the tangent hyperplane which generally deviates from the "curved" manifold. For example, for the direction of the ray segment, $\boldsymbol{v} + \delta\boldsymbol{v}$ will no longer be a unit vector. Therefore, instead of the trivial addition $\hat{\boldsymbol{X}} = \boldsymbol{X} + \delta\boldsymbol{X}$, it is also necessary to somehow return the point to the resolved manifold:

$$\hat{\boldsymbol{X}} = \mathcal{P}(\boldsymbol{X} + \delta\boldsymbol{X}). \tag{5}$$

This can be accomplished in different ways: for example, projecting the moved point onto the manifold. In case of ray direction, that means its normalization, so the new direction of the segment is $\frac{\boldsymbol{v}+\delta\boldsymbol{v}}{|\boldsymbol{v}+\delta\boldsymbol{v}|}$.

However, in the case where the potential has steep slopes, such numerical integration requires an unacceptably small time step. Moreover, for any finite δt, the stationary distribution of the trajectory does not exactly coincide with the desired one. To cure this, MALA after each time step (4)–(5) applies the Hastings rule treating $\hat{\boldsymbol{X}}$ not as the new

position but only as the conditional proposal that can be accepted (and then $\boldsymbol{X}(t+\delta t) = \hat{\boldsymbol{X}}$) or rejected (and then $\boldsymbol{X}(t+\delta t) = \boldsymbol{X}$). The Hastings rule is a powerful tool which transforms the sequence generated by an arbitrary Markov chain so that it has exactly the desired probability distribution. The probability of acceptance is computed as

$$\Pr = \min\left(1, \frac{p(\boldsymbol{X}|\hat{\boldsymbol{X}})\rho(\hat{\boldsymbol{X}})}{p(\hat{\boldsymbol{X}}|\boldsymbol{X})\rho(\boldsymbol{X})}\right), \tag{6}$$

where ρ is the density of the stationary distribution $\rho(\boldsymbol{X}) = Ze^{-U(\boldsymbol{X})}$, $p(\hat{\boldsymbol{X}}|\boldsymbol{X})$ is the probability density of the proposed transition $\boldsymbol{X} \mapsto \hat{\boldsymbol{X}}$, and $p(\boldsymbol{X}|\hat{\boldsymbol{X}})$ is the probability density of the backward transition proposal. This is the same function with the arguments swapped. The transition $\boldsymbol{X} \mapsto \hat{\boldsymbol{X}}$ is a composition of (4) and (5). As a result, MALA allows working even with a discontinuous potential with a finite (even arbitrarily large) time integration step.

It is important to understand that the Markov chain obtained as a result of combining the difference scheme and the Hastings rule is a completely different object than the original Langevin equation. It has exactly the same stationary distribution, but that is all. The other statistical properties (e.g., the time correlation function) can be completely different from that in the original Langevin dynamics. However, this is insignificant for its use as a generator of random points in phase space because we only need the desired distribution from it.

3.2. Influence of the Divergent Term

The drift term includes the divergence of $\sum_j \frac{\partial(\hat{T}\hat{T}^*)_{i,j}}{\partial x_j}$, which requires fairly expensive calculations. To accomplish this, it is not enough to calculate the matrix $\hat{T}$ itself at a given point, but it is also necessary to find how it depends on this point. In the case of a stochastic differential equation (SDE) (3), this term is fundamentally important and cannot be dropped or replaced with an approximation because then the stationary distribution will be distorted. It is difficult to predict how significantly. Therefore, it is necessary to calculate this expression as accurately as possible.

The only way to remove this term would be to use a constant transformation matrix $\hat{T}$ independent of the point in the phase space. However, this can lead to another difficulty: now, the Brownian component is not adapted to the relief of the potential, and thus it is necessary to use a very small time step to integrate the SDE with sufficient accuracy. But since now the calculations are simplified, each step is calculated faster. So sometimes (alas, rarely) even some overall acceleration is possible. However, in most cases, this leads to a fatal slowdown.

Fortunately, if the MALA algorithm is used instead of the true SDE dynamics, the resulting stationary distribution will be exact regardless of which terms are present in the SDE. One can even drop the whole drift term (both the gradient of the potential and the divergence term), reducing the dynamics to Metropolis with a Brownian mutation

$$d\boldsymbol{X} = \sqrt{2dt}\hat{T}\boldsymbol{\xi}(t), \tag{7}$$

and it will still work. The rate of convergence may depend on which components of the drift term are kept and which are dropped. However, in the example we used for testing, even this did not happen. The rate of convergence did not suffer from dropping the divergence term, as it can be seen from the results below.

4. Choosing the Integration Step

The time step is the most important parameter in the numerical integration of ODEs. Its correct and adaptive (depending on the point) choice determines the accuracy and speed of calculations. It is even more important in the case of stochastic differential equations. As it was said, all that we need from the Langevin dynamics is that it provides the target stationary distribution. Thanks to the Hastings rule, MALA provides an exact stationary

distribution for any step δt provided that the trajectory goes around the entire region $\rho > 0$. Regrettably, this condition may be violated for a sufficiently large step, as we will see below. We obtain exactly the desired distribution when this condition is satisfied. But the convergence rate may strongly depend on the time step. Indeed, a very large number of steps $O(1/\delta t)$ will be required for a very small δt, so the calculations will be very slow. On the other hand, the Hastings procedure often rejects the transition proposal for large δt repeating the current position of the point. Therefore, the trajectory turns out to be highly correlated, and this is bad for the Monte Carlo integration. Therefore, both too large and too small time steps work poorly, so there must be some value in between that produces the most efficient series of samples (points along the trajectory).

For a sufficiently large time step δt, the trajectory can become stuck at some point almost forever. Indeed, the displacement of the point under the influence of the Brownian term $\sqrt{2\delta t}\hat{T}\boldsymbol{\xi}$ cannot compete with the much larger displacement under the influence of the drift term $\boldsymbol{a}(\boldsymbol{X})\delta t$ because the latter grows much faster with δt. Therefore, although formally, the proposed new position of the point $\hat{\boldsymbol{X}} = \boldsymbol{X} + \delta\boldsymbol{X}$ is random, it cannot deviate significantly from $\hat{\boldsymbol{X}}' \equiv \boldsymbol{X} + \boldsymbol{a}(\boldsymbol{X})\delta t$. The Hastings rule (6) allows a transition to $\hat{\boldsymbol{X}}$ only if the trajectory can return from there, i.e., if $\hat{\boldsymbol{X}} + \boldsymbol{a}(\hat{\boldsymbol{X}})\delta t' + \sqrt{2\delta t'}\hat{T}(\hat{\boldsymbol{X}})\boldsymbol{\eta} = \boldsymbol{X}$, where $\boldsymbol{\eta}$ is some new random vector (instead of ξ), and $\delta t'$ is the time step at the point $\hat{\boldsymbol{X}}$. Since for large $\delta t'$ the drift dominates over the Brownian term, then $\hat{\boldsymbol{X}} + \boldsymbol{a}(\hat{\boldsymbol{X}})\delta t' \approx \boldsymbol{X}$ and so

$$\boldsymbol{X} \approx \hat{\boldsymbol{X}} + \boldsymbol{a}(\hat{\boldsymbol{X}})\delta t' \approx \boldsymbol{X} + \boldsymbol{a}(\boldsymbol{X})\delta t + \boldsymbol{a}(\hat{\boldsymbol{X}})\delta t',$$

This is possible only if $\boldsymbol{a}(\hat{\boldsymbol{X}})$ turns in the opposite direction, which is not the general case. Otherwise, the reverse transition is impossible or its probability is very small, and the Hastings rule rejects the proposed transition of $\boldsymbol{X} \mapsto \hat{\boldsymbol{X}}$.

This situation is shown in the left half of Figure 1, and its right half represents the opposite case when the time step is small and the Brownian displacement may exceed the drift.

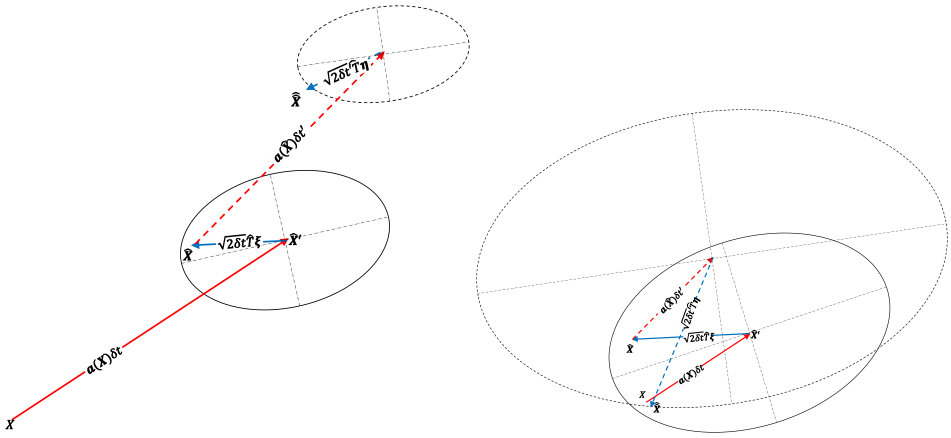


Figure 1. The left half shows the case of a sufficiently large δt when the Brownian term is small compared to the drift. The right half shows the case for small δt when the Brownian term can exceed the drift. The red arrow shows the shift due to the drift term, the blue arrow shows the shift due to the Brownian term and the ellipse encloses the region where the point will be with high probability after the combination of both shifts. The dashed arrows and the ellipse refer to the imaginary next step $\hat{\boldsymbol{X}} \mapsto \hat{\hat{\boldsymbol{X}}}$. It can return the trajectory back to $\boldsymbol{X}$ for the case shown on the right, while it cannot for the left case.

Therefore, the time step must be small enough so that the Brownian fluctuations can overcome the displacement due to the drift term. For this, it is sufficient that the size of the ellipse along the direction of the drift term is not less than the length of the red arrow, i.e., $|\boldsymbol{a}(\boldsymbol{X})\delta t|$. As a result, the time step depends on the point $\boldsymbol{X}$.

Since the Brownian displacement $\sqrt{2\delta t}\hat{T}\boldsymbol{\xi}$ is a Gaussian random vector with covariance matrix $2\delta t\hat{T}^*\hat{T}$, then the size of the ellipse along the direction $\frac{\boldsymbol{a}}{|\boldsymbol{a}|}$ is

$$3\sqrt{2\delta t\left(\frac{\boldsymbol{a}}{|\boldsymbol{a}|},\hat{T}^*\hat{T}\frac{\boldsymbol{a}}{|\boldsymbol{a}|}\right)} = 3\sqrt{2\delta t}\frac{|\hat{T}\boldsymbol{a}|}{|\boldsymbol{a}|}.$$

The factor of 3 arises due to the famous "3σ rule"; i.e., the random value almost never exceeds three r.m.s. According to our requirement, this value should be no less than $|\boldsymbol{a}\delta t|$:

$$\sqrt{2\delta t}\frac{|\hat{T}\boldsymbol{a}|}{|\boldsymbol{a}|} \geq |\boldsymbol{a}\delta t|, \tag{8}$$

and so

$$\delta t \leq 2\frac{|\hat{T}\boldsymbol{a}|^2}{|\boldsymbol{a}|^4}. \tag{9}$$

It is better to limit the time step so that the integration of the Langevin equation without the Hastings procedure yields a stationary distribution that does not deviate too much from the desired one. The stationary distribution ρ is the stationary solution of the master equation for the Markov chain corresponding to the difference scheme of integration of the SDE. For a sufficiently smooth distribution, its evolution in discrete time involves the convolution of the difference of the distribution at the initial and final points of the step $\rho(\boldsymbol{X}(t+\delta t)) - \rho(\boldsymbol{X}(t))$ with the distribution of the Brownian term. When this difference can be replaced by a Taylor series expansion of the distribution up to the second derivatives, then we obtain the parabolic PDE describing how the initial density evolves with time and converges to the stationary distribution. For the Taylor expansion to be applicable, it is at least necessary that

$$|\rho(\boldsymbol{X}+\delta\boldsymbol{X}) - \rho(\boldsymbol{X})| \leq \varepsilon\rho(\boldsymbol{X})$$

for a reasonably small ε, say, $\varepsilon = 0.1$. Estimating the left side using the gradient, we arrive at the criterion

$$(\delta\boldsymbol{X}, \nabla\log\rho)| \leq \varepsilon,$$

or, since the stationary distribution is related to the potential as $\rho(\boldsymbol{X}) = Ze^{-U(X)}$,

$$|(\delta\boldsymbol{X}, \nabla U)| \leq \varepsilon.$$

If the time step satisfies (9), then for the approximate criterion, we can take $\delta\boldsymbol{X} \approx \boldsymbol{a}(\boldsymbol{X})\delta t$ since the Brownian and the drift components are of the same order of magnitude. This gives

$$|(\boldsymbol{a}, \nabla U)|\delta t \leq \varepsilon. \tag{10}$$

Combining (10) with (9), we come to the expression of the necessary condition:

$$\delta t \leq \min\left(2\frac{|\hat{T}\boldsymbol{a}|^2}{|\boldsymbol{a}|^4}, \frac{\varepsilon}{|(\boldsymbol{a}, \nabla U)|}\right), \tag{11}$$

where ε is some reasonably small number. The exact value is not so important, because this is still an approximate derivation, in which one part is excessively sufficient and the second is minimally necessary. Therefore, we can adopt $\varepsilon = 0.1$, treat (11) as the upper bound and use 10 times smaller δt in the simulation of the dynamics. Or we take the minimum between (11) and some empirical limitation like $\delta t \leq 0.01$.

5. Phase Space and Probability Densities of Transition Proposals

The choice of ray path representation, i.e., the phase space, has a crucial role for all aspects of working with the Langevin equation as the ray path generator. In particular, the choice of phase space determines the specific form of constraints on the ray path in a given scene, and thus, on the preconditioning matrix $\hat{T}$ and the time step. The main criteria for

choosing a ray path representation is what form the constraints take for it and how much this simplifies (or complicates) the construction of the preconditioning matrix.

For a ray path of $N+1$ segments, the last one does not need to be stored because it is uniquely determined by the previous N segments (Figure 2a). If the ray path is represented by the directions of segments, then the importance function is equal to the product of the BDF values at the path nodes. The BDF (Bidirectional scattering Distribution Function) describes the optical properties of the scene surface. If there are almost specular BDFs, then for the importance function to be distinct from zero, the direction of the segment must be close to the direction of the ideal mirror reflection of the previous segment $\boldsymbol{s}_i$:

$$\begin{aligned} \boldsymbol{v}_{i+1} &\approx \boldsymbol{s}_i, \\ \boldsymbol{s}_i &= \boldsymbol{v}_i - 2(\boldsymbol{v}_i, \boldsymbol{n}_i)\boldsymbol{n}_i, \end{aligned} \tag{12}$$

where $\boldsymbol{v}_i$ is the direction of the i-th segment and $\boldsymbol{n}_i$ is the surface normal at the end point of the i-th segment. The condition $\boldsymbol{v}_{i+1} \approx \boldsymbol{s}_i$ is the constraint imposed by a glossy BDF.

Therefore, it is convenient to choose a phase space where the deviations from the direction of the specular reflection are independent variables, i.e., are included in the vector representing the ray path. These deviations can be described in different ways: for example, through the vector $\boldsymbol{\omega}_i \perp \boldsymbol{s}_i$, defined as shown in Figure 2b. For the first ray segment, there is no previous ray and its specular reflection, so the first segment must be specified explicitly, for example, through its direction $\boldsymbol{v}_1$.

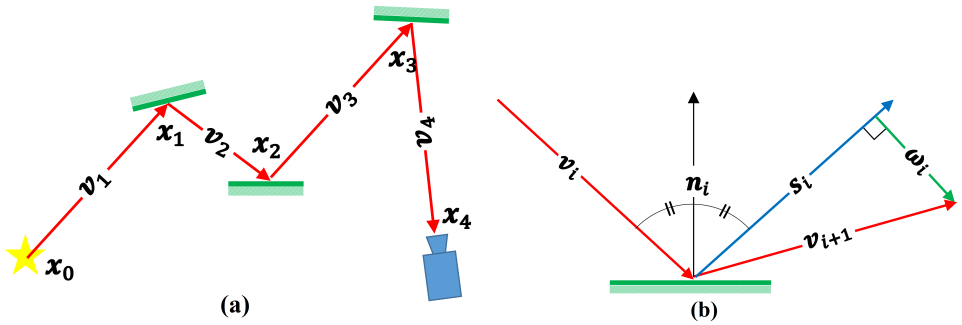


Figure 2. (**a**) The last segment ending at the camera is a function of the previous $\{v_1; v_2; ...; v_N\}$ because its end is the camera position and thus is fixed. The case $N = 4$ is shown. (**b**) The direction of a segment can be specified through its deviation from the specular scattering of the previous segment. Red arrows are the incident and reflected ray segments, the blue arrow is the specular reflection of the incident segment, the green arrow is the vector of deviation from the specular direction, and the black one is the normal to the surface.

By construction, the deviation vector $\boldsymbol{\omega}_i$ is orthogonal to the specular direction $\boldsymbol{s}_i$ and

$$\begin{aligned} \boldsymbol{v}_{i+1} &= \sqrt{1 - |\boldsymbol{\omega}_i|^2}\boldsymbol{s}_i + \boldsymbol{\omega}_i, \\ \boldsymbol{\omega}_i &= \boldsymbol{v}_{i+1} - (\boldsymbol{v}_{i+1}, \boldsymbol{s}_i)\boldsymbol{s}_i. \end{aligned} \tag{13}$$

If we represent the ray path by the vector $\{v_1; \boldsymbol{\omega}_1; \boldsymbol{\omega}_2; \boldsymbol{\omega}_3; ...; \boldsymbol{\omega}_{N-1}\}$, the Jacobian of the transition from the set of directions $\{v_1; v_2; ...; v_N\}$ is unity, so the importance function in this representation is the same, i.e., the product of BDFs (in intensity units). The $\boldsymbol{v}_i$ values included in it are calculated from $\{v_1; \boldsymbol{\omega}_1; \boldsymbol{\omega}_2; \boldsymbol{\omega}_3; ...; \boldsymbol{\omega}_{i-1}\}$.

Since $\boldsymbol{\omega}_i$ is orthogonal to the specular direction $\boldsymbol{s}_i$, then this is actually a two-dimensional vector $\boldsymbol{\Omega}_i = \{\Omega_{i,p}, \Omega_{i,q}\}$ containing projections onto some two directions $\boldsymbol{p}'$ and $\boldsymbol{q}'$ which are perpendicular to $\boldsymbol{s}_i$.

Thus, the representation of a ray path (a point in phase space) with N segments is now

$$\boldsymbol{X} = \{v_1; \boldsymbol{\Omega}_1; \boldsymbol{\Omega}_2; \boldsymbol{\Omega}_3; ...; \boldsymbol{\Omega}_{N-1}\}.$$

Variation $\delta\boldsymbol{X}$ is a vector in the tangent hyperplane. The corresponding variation of the unit vector $\boldsymbol{v}_1$ is a two-dimensional vector $\delta\boldsymbol{V}_1 = \{\delta V_p; \delta V_q\}$, containing two projections of $\delta\boldsymbol{v}_1$ onto the directions $\boldsymbol{p}_1$ and $\boldsymbol{q}_1$ perpendicular to $\boldsymbol{v}_1$. Thus,

$$\delta\boldsymbol{X} = \{\delta\boldsymbol{V}_1; \delta\boldsymbol{\Omega}_1; ...; \delta\boldsymbol{\Omega}_N\}.$$

For such a representation, it is impossible to calculate a new point simply as $\hat{\boldsymbol{X}} = \boldsymbol{X} + \delta\boldsymbol{X}$ because then the first element $\boldsymbol{v}_1 + \delta\boldsymbol{v}_1$ will no longer be a unit vector. The simplest way is to normalize the result, which is a special case of the projection onto the allowed manifold:

$$\begin{aligned}\hat{\boldsymbol{X}} &= \left\{\frac{\boldsymbol{v}_1 + \delta\boldsymbol{v}_1}{|\boldsymbol{v}_1 + \delta\boldsymbol{v}_1|}; \quad \boldsymbol{\Omega}_1 + \delta\boldsymbol{\Omega}_1; \quad \boldsymbol{\Omega}_2 + \delta\boldsymbol{\Omega}_2; ...; \quad \boldsymbol{\Omega}_{N-1} + \delta\boldsymbol{\Omega}_{N-1}\right\} \\ &= \left\{\frac{\boldsymbol{v}_1 + \boldsymbol{p}_1\delta V_p + \boldsymbol{q}_1\delta V_q}{\sqrt{1 + \delta V_p^2 + \delta V_q^2}}; \quad \boldsymbol{\Omega}_1 + \delta\boldsymbol{\Omega}_1; \right. \\ &\quad \left. \boldsymbol{\Omega}_2 + \delta\boldsymbol{\Omega}_2; ...; \quad \boldsymbol{\Omega}_{N-1} + \delta\boldsymbol{\Omega}_{N-1}\right\}. \end{aligned} \tag{14}$$

Other ways are also possible. For example, we can take a new direction as $\hat{\boldsymbol{v}}_1 = \sqrt{1 - |\delta\boldsymbol{v}_1|^2}\boldsymbol{v}_1 + \delta\boldsymbol{v}_1$. The importance function in this space (as well as in the space where the path is described by the directions of all segments) turns out to be simply the product of the BDF values at the path nodes.

After having chosen the phase space, we must implement the Langevin Monte Carlo ray tracing in it. This requires the probability density of the transition proposal in this phase space, i.e., the probability density with which $\hat{\boldsymbol{X}}$ is distributed for a fixed $\boldsymbol{X}$ and a random $\boldsymbol{\xi}$. This mapping is defined by two formulas: we find $\delta\boldsymbol{X} = \{\delta\boldsymbol{V}_1; \delta\boldsymbol{\Omega}_1; ...; \delta\boldsymbol{\Omega}_N\}$ with Formula (4) and $\hat{\boldsymbol{X}}$ with Formula (14).

Thus, the conditional probability density function $p(\hat{\boldsymbol{X}}|\boldsymbol{X})$ refers to the case when $\boldsymbol{X}$ is fixed, and the randomness of $\hat{\boldsymbol{X}}$ is due to the random vector $\boldsymbol{\xi}$. And since the random variable $\hat{\boldsymbol{X}}$ is a function of the random variable $\boldsymbol{\xi}$ distributed according to Gaussian density G, then the probability density is calculated as

$$p(\hat{\boldsymbol{X}}|\boldsymbol{X}) = \frac{1}{\left|\det \frac{\partial\hat{\boldsymbol{X}}}{\partial\boldsymbol{\xi}}\right|} G(\boldsymbol{\xi}), \tag{15}$$

where $\boldsymbol{\xi}$ should be calculated from $\hat{\boldsymbol{X}}$ and $\boldsymbol{X}$. First, we calculate $\delta\boldsymbol{X} = \{\delta\boldsymbol{V}_1; \delta\boldsymbol{\Omega}_1; ...; \delta\boldsymbol{\Omega}_N\}$ from Formula (14):

$$\delta\boldsymbol{v}_1 = \frac{\hat{\boldsymbol{v}}_1}{(\hat{\boldsymbol{v}}_1, \boldsymbol{v}_1)} - \boldsymbol{v}_1, \tag{16}$$

$$\delta\boldsymbol{V}_1 = \{(\boldsymbol{p}_1, \delta\boldsymbol{v}_1); (\boldsymbol{q}_1, \delta\boldsymbol{v}_1)\}, \tag{17}$$

$$\delta\boldsymbol{\Omega}_i = \hat{\boldsymbol{\Omega}}_i - \boldsymbol{\Omega}_i \tag{18}$$

and then calculate $\boldsymbol{\xi}$:

$$\boldsymbol{\xi} = \frac{1}{\sqrt{2\delta t}} \hat{T}^{-1}(\boldsymbol{X})(\delta\boldsymbol{X} - \boldsymbol{a}(\boldsymbol{X})\delta t). \tag{19}$$

The Jacobian $\left|\det \frac{\partial\hat{\boldsymbol{X}}}{\partial\boldsymbol{\xi}}\right|$ of the combined transformation is the product of its two "halves": $\left|\det \frac{\partial\hat{\boldsymbol{X}}}{\partial\boldsymbol{\xi}}\right| = \left|\det \frac{\partial(\delta\boldsymbol{X})}{\partial\boldsymbol{\xi}} \det \frac{\partial\hat{\boldsymbol{X}}}{\partial(\delta\boldsymbol{X})}\right|$. The first term is found from Formula (4): $\det \frac{\partial\hat{\boldsymbol{X}}}{\partial(\delta\boldsymbol{X})} = (\sqrt{2\delta t})^N \left|\det \hat{T}\right|$ and the second term—from the transformation (14):

$$\det\frac{\partial(\hat{\boldsymbol{v}}_1;\hat{\boldsymbol{\Omega}}_1;\ldots;\delta\hat{\boldsymbol{\Omega}}_N)}{\partial(\delta\boldsymbol{V}_1;\delta\boldsymbol{\Omega}_1;\ldots;\delta\boldsymbol{\Omega}_N)}=\det\begin{pmatrix}\frac{\partial(\hat{\boldsymbol{v}}_1)}{\partial(\delta\boldsymbol{V}_1)} & 0 & 0 & \cdots & 0\\ 0 & 1 & 0 & \cdots & 0\\ 0 & 0 & 1 & \cdots & 0\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ 0 & 0 & 0 & \cdots & 1\end{pmatrix}=\det\frac{\partial(\hat{\boldsymbol{v}}_1)}{\partial(\delta\boldsymbol{V}_1)},$$

where $\hat{\boldsymbol{v}}_1=\frac{\boldsymbol{v}+\boldsymbol{p}\delta V_p+\boldsymbol{q}\delta V_q}{\sqrt{1+\delta V_p^2+\delta V_q^2}}$ according to (14).

Since $\hat{\boldsymbol{v}}_1$ is a unit vector, its derivative is orthogonal to it and so effectively has just two (not three) components along the two directions $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{q}}$ orthogonal to $\hat{\boldsymbol{v}}_1$. They are the analogues of $\boldsymbol{p}$ and $\boldsymbol{q}$. The vector $\delta\boldsymbol{V}_1$ (not only its differential) also has just two components $\{\delta V_p;\delta V_q\}$. Therefore, the Jacobi matrix $\frac{\partial(\hat{\boldsymbol{v}}_1)}{\partial(\delta\boldsymbol{V}_1)}$ is 2×2:

$$\frac{\partial(\hat{\boldsymbol{v}}_1)}{\partial(\delta\boldsymbol{V}_1)}=\begin{pmatrix}\left(\hat{\boldsymbol{p}},\frac{\partial\hat{\boldsymbol{v}}_1}{\partial(\delta V_p)}\right) & \left(\hat{\boldsymbol{p}},\frac{\partial\hat{\boldsymbol{v}}_1}{\partial(\delta V_q)}\right)\\ \left(\hat{\boldsymbol{q}},\frac{\partial\hat{\boldsymbol{v}}_1}{\partial(\delta V_p)}\right) & \left(\hat{\boldsymbol{q}},\frac{\partial\hat{\boldsymbol{v}}_1}{\partial(\delta V_q)}\right)\end{pmatrix}$$

Substituting $\frac{\partial\hat{\boldsymbol{v}}_1}{\partial(\delta V_p)}=\frac{\boldsymbol{p}}{\sqrt{1+\delta V_p^2+\delta V_q^2}}-\hat{\boldsymbol{v}}_1\frac{\delta V_p}{1+\delta V_p^2+\delta V_q^2}$ and $\frac{\partial\hat{\boldsymbol{v}}_1}{\partial(\delta V_q)}=\frac{\boldsymbol{q}}{\sqrt{1+\delta V_p^2+\delta V_q^2}}-\hat{\boldsymbol{v}}_1\frac{\delta V_q}{1+\delta V_p^2+\delta V_q^2}$ and taking into account that $(\hat{\boldsymbol{p}},\hat{\boldsymbol{v}}_1)=(\hat{\boldsymbol{q}},\hat{\boldsymbol{v}}_1)=0$, one calculates the elements of this matrix:

$$\begin{aligned}\frac{\partial(\hat{\boldsymbol{v}}_1)}{\partial(\delta\boldsymbol{V}_1)} &= \frac{1}{\sqrt{1+\delta V_p^2+\delta V_q^2}}\begin{pmatrix}(\hat{\boldsymbol{p}},\boldsymbol{p}) & (\hat{\boldsymbol{p}},\boldsymbol{q})\\ (\hat{\boldsymbol{q}},\boldsymbol{p}) & (\hat{\boldsymbol{q}},\boldsymbol{q})\end{pmatrix}\\ &= (\hat{\boldsymbol{v}}_1,\boldsymbol{v}_1)\begin{pmatrix}(\hat{\boldsymbol{p}},\boldsymbol{p}) & (\hat{\boldsymbol{p}},\boldsymbol{q})\\ (\hat{\boldsymbol{q}},\boldsymbol{p}) & (\hat{\boldsymbol{q}},\boldsymbol{q})\end{pmatrix}\end{aligned}$$

so

$$\det\frac{\partial(\hat{\boldsymbol{v}}_1)}{\partial(\delta\boldsymbol{V}_1)}=(\hat{\boldsymbol{v}}_1,\boldsymbol{v}_1)^2|(\boldsymbol{p},\hat{\boldsymbol{p}})(\boldsymbol{q},\hat{\boldsymbol{q}})-(\boldsymbol{q},\hat{\boldsymbol{p}})(\boldsymbol{p},\hat{\boldsymbol{q}})|$$

As a result,

$$\left|\det\frac{\partial\hat{\boldsymbol{X}}}{\partial\boldsymbol{\xi}}\right|=(\sqrt{2\delta t})^N|\det\hat{T}|(\hat{\boldsymbol{v}}_1,\boldsymbol{v}_1)^2|(\boldsymbol{p}_1,\hat{\boldsymbol{p}}_1)(\boldsymbol{q}_1,\hat{\boldsymbol{q}}_1)-(\boldsymbol{q}_1,\hat{\boldsymbol{p}}_1)(\boldsymbol{p}_1,\hat{\boldsymbol{q}}_1)|$$

and

$$p(\hat{\boldsymbol{X}}|\boldsymbol{X})=\frac{G(\boldsymbol{\xi})}{(\sqrt{2\delta t})^N|\det\hat{T}(\boldsymbol{X})|(\boldsymbol{v}_1,\hat{\boldsymbol{v}}_1)^2|(\hat{\boldsymbol{p}}_1,\boldsymbol{p}_1)(\hat{\boldsymbol{q}}_1,\boldsymbol{q}_1)-(\hat{\boldsymbol{p}}_1,\boldsymbol{q}_1)(\hat{\boldsymbol{q}}_1,\boldsymbol{p}_1)|},\tag{20}$$

where $\boldsymbol{\xi}$ is defined by Formula (19).

The probability density of the transition in the opposite direction is, of course, the same function but with the swapped arguments:

$$p(\hat{\boldsymbol{X}}|\boldsymbol{X})=\frac{G(\boldsymbol{\eta})}{(\sqrt{2\delta t'})^N|\det\hat{T}(\boldsymbol{X})|(\boldsymbol{v}_1,\hat{\boldsymbol{v}}_1)^2|(\hat{\boldsymbol{p}}_1,\boldsymbol{p}_1)(\hat{\boldsymbol{q}}_1,\boldsymbol{q}_1)-(\hat{\boldsymbol{p}}_1,\boldsymbol{q}_1)(\hat{\boldsymbol{q}}_1,\boldsymbol{p}_1)|},$$

where $\delta t'$ is the time step calculated for the case when the initial point is $\hat{\boldsymbol{X}}$ and

$$\boldsymbol{\eta}=\frac{1}{\sqrt{2\delta t'}}\hat{T}^{-1}(\hat{\boldsymbol{X}})\left(\delta\boldsymbol{X}'-\boldsymbol{a}(\hat{\boldsymbol{X}})\delta t\right)$$

and eventually $\delta\boldsymbol{X}'=\{\delta\boldsymbol{V}_1';\delta\boldsymbol{\Omega}_1';\ldots;\delta\boldsymbol{\Omega}_N'\}$ is the displacement of the point in the tangent hyperplane during an imaginary backward transition from $\hat{\boldsymbol{X}}$ to $\boldsymbol{X}$. Therefore, it is calculated exactly as for the direct transition by swapping $\boldsymbol{X}\leftrightarrow\hat{\boldsymbol{X}}$ in Formulas (16)–(18).

Note that the term $(\boldsymbol{v}_1,\hat{\boldsymbol{v}}_1)^2|(\hat{\boldsymbol{p}}_1,\boldsymbol{p}_1)(\hat{\boldsymbol{q}}_1,\boldsymbol{q}_1)-(\hat{\boldsymbol{p}}_1,\boldsymbol{q}_1)(\hat{\boldsymbol{q}}_1,\boldsymbol{p}_1)|$ in the denominator of the expressions for $p(\hat{\boldsymbol{X}}|\boldsymbol{X})$ and $p(\boldsymbol{X}|\hat{\boldsymbol{X}})$ is the same due to its symmetry against the

permutation $\boldsymbol{X} \leftrightarrow \hat{\boldsymbol{X}}$. So, it cancels out when calculating the probability of acceptance in the Hastings rule (6).

6. Constraints on the Variation of Ray Path

Often, the relief of the importance function in the phase space contains narrow ravines, and the trajectory must make its way inside them. If the ravine has smooth slopes, then the gradient of the potential pulls the trajectory toward the midline of the ravine. For a narrow ravine, the gradient is large. Therefore, it is necessary to strongly limit δt so that it does not jump out of the ravine. But the movement along the ravine for small δt is too slow, so a very large number of steps are needed to traverse it entirely (maybe several times to accumulate due statistics).

The situation is similar to gradient descent methods, as there, the situation is significantly improved by a local transformation of the variables by the "preconditioning matrix" in hybrid Monte Carlo algorithms. This matrix makes the projection of $\delta \boldsymbol{X}$ small on the direction across the ravine and large on the direction along it. We can say that it adapts the displacement proposal $\delta \boldsymbol{X}$ to the relief.

In ray tracing, such narrow ravines with smooth slopes arise mainly due to the presence of glossy surfaces whose BDF quickly vanishes when the ray deviates from the direction of specular reflection or refraction. Another case is a ravine with vertical slopes, i.e., a narrow potential well or at least a significant finite discontinuity of the potential. They can arise, for example, when a ray is scattered by a small scene object or passes through a small hole (Figure 3).

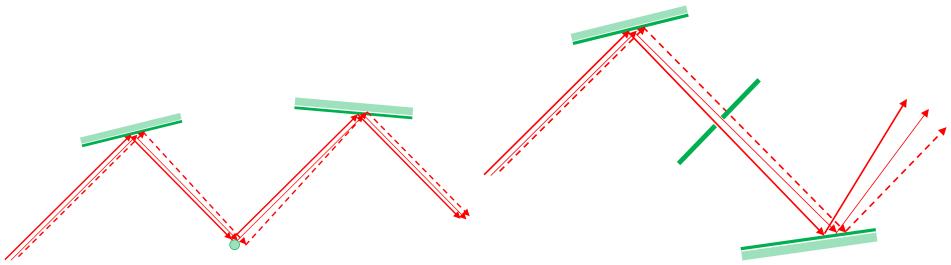


Figure 3. Two cases of a narrow potential well. Left: a ray reflects from a small scene object. Right: a ray passes through a small hole. Even a small variation in the path can render it invalid. The thick red arrows show segments of the original ray path, the thin arrows show possible mutations and the dotted arrows show forbidden mutations (impossible paths).

The case of passing through a small hole is the most intuitive. Let the nodes of the ray path before and after the hole be $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k+1}$ (these are the beginning and end of one segment) and the hole is in the plane $(\boldsymbol{x}, \boldsymbol{n}) = c$. Then, the segment intersects this plane at the point

$$\boldsymbol{x} = \boldsymbol{x}_k + \frac{c - (\boldsymbol{x}_k, \boldsymbol{n})}{((\boldsymbol{x}_{k+1}, \boldsymbol{n}) - (\boldsymbol{x}_k, \boldsymbol{n}))}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)$$

and an infinitesimal displacement of the nodes $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k+1}$ leads to the following variation of the intersection point of the plane:

$$\begin{aligned} d\boldsymbol{x} \quad = \quad & -\frac{c - (\boldsymbol{x}_{k+1}, \boldsymbol{n})}{((\boldsymbol{x}_{k+1}, \boldsymbol{n}) - (\boldsymbol{x}_k, \boldsymbol{n}))} d\boldsymbol{x}_k + \frac{c - (\boldsymbol{x}_{k+1}, \boldsymbol{n})}{((\boldsymbol{x}_{k+1}, \boldsymbol{n}) - (\boldsymbol{x}_k, \boldsymbol{n}))^2}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)(d\boldsymbol{x}_k, \boldsymbol{n}) \\ & + \frac{c - (\boldsymbol{x}_k, \boldsymbol{n})}{((\boldsymbol{x}_{k+1}, \boldsymbol{n}) - (\boldsymbol{x}_k, \boldsymbol{n}))} d\boldsymbol{x}_{k+1} - \frac{c - (\boldsymbol{x}_k, \boldsymbol{n})}{((\boldsymbol{x}_{k+1}, \boldsymbol{n}) - (\boldsymbol{x}_k, \boldsymbol{n}))^2}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)(d\boldsymbol{x}_{k+1}, \boldsymbol{n}), \end{aligned}$$

which can be rewritten in a matrix form

$$d\boldsymbol{x} = \hat{A}(\boldsymbol{x}_k; \boldsymbol{x}_{k+1}) d\boldsymbol{x}_k + \hat{A}(\boldsymbol{x}_{k+1}; \boldsymbol{x}_k) d\boldsymbol{x}_{k+1}.$$

where $\hat{A}(\boldsymbol{x};\boldsymbol{y})$ is a 3×3 matrix calculated from two vertices $\boldsymbol{x}, \boldsymbol{y}$.

Obviously, in this case, in the tangent subspace $(d\boldsymbol{x}_k; d\boldsymbol{x}_{k+1})$, there are directions for which the displacement along them does not change $\boldsymbol{x}$ at all. So, even a large variation of the path in this direction leaves it in the allowed region of the phase space. There are directions in which the displacement along them moves the point where the ray runs through the hole along the long side of the hole. So, in this direction, the path can be varied rather strongly until it leaves the allowed region. And, finally, there are directions that move the point where the ray runs through the hole along its short side. Here, the permissible step is the smallest.

Let $\{\boldsymbol{e}_i\}$ be the set of orthonormal vectors adapted to the orientation of the hole and l_i be the size of the hole along $\boldsymbol{e}_i$. Then, the constraint on the displacement of the intersection point of the ray with the plane of the hole is

$$(d\boldsymbol{x}, \boldsymbol{e}_i)^2 \leq l_i^2, \qquad i = 1, ..., N,$$

that is

$$(\hat{A}(\boldsymbol{x}_k; \boldsymbol{x}_{k+1}) d\boldsymbol{x}_k + \hat{A}(\boldsymbol{x}_{k+1}; \boldsymbol{x}_k) d\boldsymbol{x}_{k+1}, \boldsymbol{e}_i)^2 \leq l_i^2, \qquad i = 1, ..., N. \tag{21}$$

The case when the ray hits a small scene object is considered similarly. Let the hit point be the k-th node $\boldsymbol{x}_k$ of the path. Then, the allowed mutations are those for which $\boldsymbol{x}_k + d\boldsymbol{x}_k$ is still on the surface of this body. Approximately, this means that the displacement $d\boldsymbol{x}_k$ should be limited differently in different directions. Its general form is as follows:

$$(d\boldsymbol{x}_k, \hat{B}(\boldsymbol{x}_k) d\boldsymbol{x}_k) \leq 1. \tag{22}$$

Let us consider the phase space we have chosen. The deviations from the specular direction are independent variables, and they are the components of the ray path representation for this space. Therefore, the restrictions on them are as follows:

$$|\boldsymbol{\Omega}_i| \leq \max \vartheta_i, \tag{23}$$

where $\max \vartheta_i$ is the BDF angle beyond which this BDF practically vanishes. Note that the phase space itself does not impose restrictions on this variable, and any vector $\boldsymbol{\Omega}_i$ is admissible. Another thing is that the BDF values for it will be extremely small.

We limit the variation of the original path so that

$$(\delta\boldsymbol{\Omega}_i, \delta\boldsymbol{\Omega}_i) \leq (\max \vartheta_i)^2, \tag{24}$$

then after the variation, the worst case will be $|\boldsymbol{\Omega}_i| \leq 2 \max \vartheta_i$.

However, the value of $\boldsymbol{\Omega}_N$ in the final segment ending at the light source is not an independent variable and is not even included in the state vector. Similarly, $d\boldsymbol{\Omega}_N$ is not included in $d\boldsymbol{X}$ but is calculated from it, that is, from $(d\boldsymbol{v}_1; d\boldsymbol{\Omega}_1; d\boldsymbol{\Omega}_2; d\boldsymbol{\Omega}_3; ...; d\boldsymbol{\Omega}_{N-1})$. The general form of this dependence is obviously as follows:

$$d\boldsymbol{\Omega}_N = \hat{B} d\boldsymbol{X} = \hat{B}_0 d\boldsymbol{V}_1 + \hat{B}_1 d\boldsymbol{\Omega}_1 + \hat{B}_2 d\boldsymbol{\Omega}_2 + \cdots + \hat{B}_{N-1} d\boldsymbol{\Omega}_{N-1}. \tag{25}$$

This relation is approximately satisfied for finite variations as well. Therefore, the restriction $|\delta\boldsymbol{\omega}_N| \leq \max \vartheta_N$ leads to

$$|\hat{B}_0 \delta\boldsymbol{V}_1 + \hat{B}_1 \delta\boldsymbol{\Omega}_1 + \hat{B}_2 \delta\boldsymbol{\Omega}_2 + \cdots + \hat{B}_{N-1} \delta\boldsymbol{\Omega}_{N-1}| \leq \max \vartheta_N,$$

which can be written as a quadratic form

$$(\delta\boldsymbol{X}, \hat{Q}_1 \delta\boldsymbol{X}) \leq (\max \vartheta_N)^2. \tag{26}$$

This expresses the angular constraints in the scene.

Now, let us come to the spatial constraints. There are two that are the most common. The variations of the ray path nodes should, first, be constrained so that the normal at

these points does not change too much and so that the point does not move away from the surface. Second, the variation of the intersection point of the ray with the plane of the hole must remain inside the hole.

The first constraint applies to each node separately and can be written as a quadratic form

$$(\delta \boldsymbol{x}_k, \hat{D}(\boldsymbol{x}_k)\delta \boldsymbol{x}_k) \le d_k^2. \tag{27}$$

For our phase space, the coordinates of the path nodes are not independent variables. They must be calculated based on the state vector $\boldsymbol{X}$. Therefore,

$$d\boldsymbol{x}_k = \hat{\mathcal{V}}_k d\boldsymbol{X}, \tag{28}$$

Generally, $\hat{\mathcal{V}}_k$ is not a square matrix. The relation (28) is approximately satisfied for finite variations as well, and substituting it into (27), we have

$$\begin{aligned} \hat{Q}_k &\equiv \hat{\mathcal{V}}_k^* \hat{D}(\boldsymbol{x}_k)\hat{\mathcal{V}}_k, \\ (\delta \boldsymbol{X}, \hat{Q}_k \delta \boldsymbol{X}) &\le d_k^2. \end{aligned} \tag{29}$$

The second constraint is rather the relation between the variations of two nodes: they must be moved so that the point where the ray runs through the hole does not leave that hole. Then, we obtain an expression of the form (21). It is true that if the displacements of the path nodes are small, this is so due to the first constraints. Expressing $d\boldsymbol{x}_k$ from $d\boldsymbol{X}$ by the Formula (28) and applying the result to the final variations, we have

$$(\delta \boldsymbol{X}, \boldsymbol{w}_{i,k})^2 \le l_i^2, \qquad i = 1, ..., N,$$

where

$$\begin{aligned} \hat{W}_k &\equiv \hat{A}(\boldsymbol{x}_k; \boldsymbol{x}_{k+1})\hat{\mathcal{V}}_k + \hat{A}(\boldsymbol{x}_{k+1}; \boldsymbol{x}_k)\hat{\mathcal{V}}_{k+1}, \\ \boldsymbol{w}_{i,k} &\equiv \hat{W}_k^* \boldsymbol{e}_i^{(k)}. \end{aligned}$$

The quadratic form $(\delta \boldsymbol{X}, \boldsymbol{w}_{i,k})^2$ can also be written in a matrix form:

$$\begin{aligned} (\delta \boldsymbol{X}, \hat{Q}_{i,k}\delta \boldsymbol{X}) &\le l_i^2, \qquad i = 1, ..., N, \\ \hat{Q}_{i,k} &\equiv |\boldsymbol{w}_{i,k}\rangle\langle \boldsymbol{w}_{i,k}|. \end{aligned} \tag{30}$$

In general, all constraints (24), (26), (29) and (30) can be written as

$$(\delta \boldsymbol{X}, \hat{Q}_k(\boldsymbol{X})\delta \boldsymbol{X}) \le q_k^2,$$

where $\hat{Q}_k(\boldsymbol{X})$ is a symmetric non-negative matrix.

Of course, some of these constraints may be absent for the given scene, say, if all BDFs are smooth enough or if all holes are large, etc. Formally, we can keep all constraints but set a very large or simply infinite q_k for the absent constrains.

The allowed variations of the ray path must satisfy all the constraints. The simplest way to combine them into one is to write all the corresponding inequalities in the form $(\delta \boldsymbol{X}, q_k^{-2}\hat{Q}_k(\boldsymbol{X})\delta \boldsymbol{X}) \le 1$ and sum them up. This leads to

$$\begin{aligned} (\delta \boldsymbol{X}, \hat{Q}(\boldsymbol{X})\delta \boldsymbol{X}) &\le 1, \\ \hat{Q}(\boldsymbol{X}) &\equiv \sum_{k=1}^{M} q_k^{-2}\hat{Q}_k(\boldsymbol{X}). \end{aligned} \tag{31}$$

Of course, this is an excessively strong constraint; i.e., in reality, the phase space point may remain in the allowed region even for $\delta \boldsymbol{X}$ significantly exceeding the bounds derived

from (31). As a compromise, it would be possible to make the matrix in the combined constraint slightly different, for example,

$$\hat{Q}(\boldsymbol{X}) = \frac{1}{M}\sum_k q_k^{-2}\hat{Q}_k(\boldsymbol{X}),$$

where M is the number of the initial, individual constraints. If they all are identical, such a matrix would provide a perfectly precise constraint in contrast to the excessive requirement from (31).

However, in case of different initial constraints, such a matrix may lead to the proposed $\delta\boldsymbol{X}$ being $\sqrt{M}$ times larger than allowed. Such a transition proposal will surely be rejected. However, many proposals will still be accepted and will give a larger trajectory variation. It is not clear in advance what is better: all accepted small steps or large steps, but some of them are rejected. Therefore, it is possible to propose to form the final matrix as

$$\hat{Q}(\boldsymbol{X}) = \frac{1}{\sqrt{M}}\sum_k q_k^{-2}\hat{Q}_k(\boldsymbol{X}).$$

In any way, we still arrive at an inequality for a quadratic form similar to (31). The difference is only in the matrix $\hat{Q}(\boldsymbol{X})$ that can be calculated slightly differently.

7. Calculation of the Preconditioning Matrix from the Constraints

For any kind of dynamics, be it (2), (3), or (7), a transition proposal can move the point to a place where the importance function vanishes (or nearly vanishes) and such a mutation will be rejected by the Hastings rule. The acceptable transitions form a region around the current point (i.e., the ray path). It is therefore reasonable to arrange that all proposed transitions $\delta\boldsymbol{X}$ are automatically restricted to this region. This means that we impose some limitations on the vector $\delta\boldsymbol{X}$ or constraints among its components. The most typical restrictions arise from, first, the presence of glossy surfaces and, second, from situations where the ray hits a small object in the scene or passes through a small hole.

Although different scenes have a different number and different kinds of constraints on the ray path, they admit a simple uniform formulation (31). In case the time step obeys (9), the full displacement $\delta\boldsymbol{X}$ and its Brownian component are about the same (see Section 4). Therefore, it is possible to apply the constraint to the Brownian component $\sqrt{2\delta t}\hat{T}\boldsymbol{\xi}$ only, which yields

$$2\delta t(\boldsymbol{\xi}, \hat{T}^*\hat{Q}(\boldsymbol{X})\hat{T}\boldsymbol{\xi}) \le 1. \tag{32}$$

Since $\hat{Q}$ is a positively defined symmetric matrix, then $\hat{Q} = \left(\sqrt{\hat{Q}}\right)^*\sqrt{\hat{Q}}$, and (32) takes the form

$$(\boldsymbol{\xi}, \left(\sqrt{\hat{Q}}\hat{T}\right)^*\sqrt{\hat{Q}}\hat{T}\boldsymbol{\xi}) \le \frac{1}{2\delta t}.$$

A natural possible solution is $\hat{T} = c\left(\sqrt{\hat{Q}}\right)^{-1}$, where c is a constant. Then, our condition becomes $(\boldsymbol{\xi}, \boldsymbol{\xi})c^2 \le \frac{1}{2\delta t}$, which must be satisfied for nearly all random vectors $\boldsymbol{\xi}$.

Since $\boldsymbol{\xi}$ is normally distributed with a unit covariance matrix, it obeys the "3σ rule": its length nearly always is below the r.m.s, i.e., $(\boldsymbol{\xi}, \boldsymbol{\xi}) \le 3^2$. Therefore, it is possible to take $c \le \frac{1}{3\sqrt{2\delta t}}$, and then all the constraints will be satisfied for most of the transition proposals.

On the other hand, (11) is yet another condition binding $\hat{T}$ and δt, and our solution $\hat{T} = c\left(\sqrt{\hat{Q}}\right)^{-1}$ must satisfy it. The drift term $\boldsymbol{a}$ in (11) is

$$\begin{aligned} a_i &\equiv -\left(\hat{T}\hat{T}^*\nabla U\right)_i + \frac{\partial(\hat{T}\hat{T}^*)_{i,j}}{\partial X_j} = c^2 a'_i, \\ a'_i &\equiv -\left(\hat{Q}^{-1}\nabla U\right)_i + \frac{\partial(\hat{Q}^{-1})_{i,j}}{\partial X_j} \end{aligned}$$

so (11) becomes

$$\delta t \le c^{-2} \min\left(2\frac{|\left(\sqrt{\hat{Q}}\right)^{-1} \boldsymbol{a'}|^2}{|\boldsymbol{a'}|^4}, \frac{\varepsilon}{|(\boldsymbol{a'}, \nabla U)|} \right).$$

Meanwhile, our new limitation $c \le \frac{1}{3\sqrt{2\delta t}}$ can be written as $\delta t \le c^{-2}/18$. Combining these two limitations, we arrive to

$$\delta t \le c^{-2} \min\left(\frac{1}{18}, \quad 2\frac{|\left(\sqrt{\hat{Q}}\right)^{-1} \boldsymbol{a'}|^2}{|\boldsymbol{a'}|^4}, \frac{\varepsilon}{|(\boldsymbol{a'}, \nabla U)|} \right).$$

As a result, one can take any c and then the time step from the above condition. Historically, we adopted $c = 1$, i.e.,

$$\begin{aligned} a &= a' \\ \hat{T} &= \left(\sqrt{\hat{Q}}\right)^{-1}. \end{aligned}$$

Finally, the limitation (11) on the time step takes the form

$$\delta t \le \min\left(\frac{1}{18}, \quad 2\frac{|\hat{T}\boldsymbol{a}|^2}{|\boldsymbol{a}|^4}, \quad \frac{\varepsilon}{|(\boldsymbol{a}, \nabla U)|} \right). \tag{33}$$

8. Results

Here, we present and analyze the results of computational experiments that confirm the conclusions of our research.

8.1. The Benchmark Scene

In most cases, a specular path cannot be varied because there are no other specular paths in its vicinity. Then, the importance function is distinct from zero only in a very small region, which is nearly a point. Such a simple case is unsuitable for studying the work of Langevin Monte Carlo ray tracing with limitations imposed by BDF (optical properties). Therefore, we chose a scene where two elliptical fuzzy mirrors redirect all rays from the point light source to the virtual detector that registers the angular distribution of light coming from all directions. The mirrors are not the closed ellipsoids but some parts of them; the rest is clipped off so that the light can enter or leave them (Figure 4). Another typical source of limitations comes from light passing through a small hole. So, we included a diaphragm with a small round hole in our scene; light can reach the detector only through it. This element resembles the use of a field diaphragm in telescopes.

The benchmark scene is shown in Figure 4. A ray emitted from the light source can hit the detector directly going along the axis from left to right, thus forming a path of one segment. However, it can also hit the detector along a three-segment path (the first segment: from the light source to the surface of mirror 1, the second: through the diaphragm to the surface of mirror 2, the third segment: from the mirror 2 to the detector). If there is a diffuse component at the mirror surfaces, then the ray can even experience multiple reflections and reach the detector through, say, seven segments. However, a three-segment path is always present both when the surfaces are perfectly specular and when they are completely diffuse. We used the MALA [8] algorithm for the case of three-segment paths.

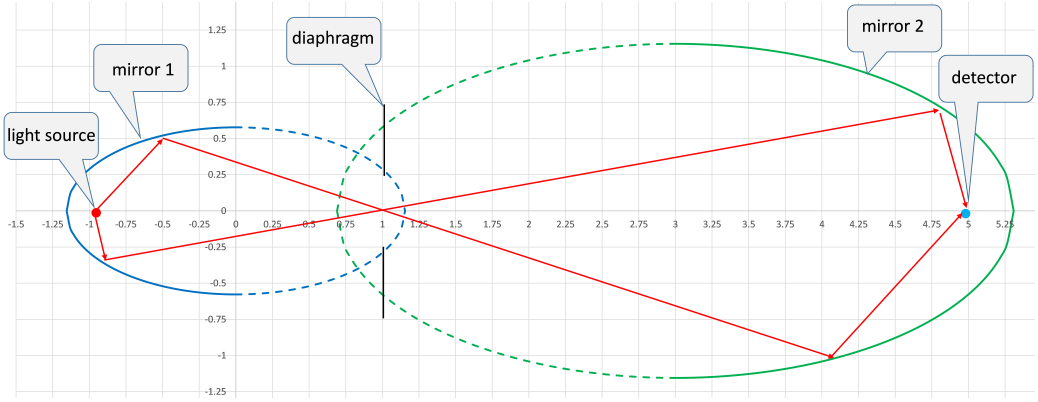


Figure 4. The benchmark scene shown as a projection onto the Oxz plane with the z-axis directed to the right. Elliptical mirrors 1 and 2 are shown in blue and green, respectively. The clipped parts of ellipsoids (absent in the scene) are drawn with the blue and green dotted lines. The right focus of the left ellipsoid and the left focus of the right ellipsoid coincide. The thick black line is the diaphragm with a hole in its center; its plane passes through the coinciding focuses. The light source is located at the left focus of ellipsoid 1 (red dot), and the point detector is at the right focus of ellipsoid 2 (blue dot). Red arrows show possible three-segment light paths.

The optical properties of the surfaces of the mirrors are described by the Phong-like BDF:

$$f = C\frac{(\cos\vartheta)^{g_s}}{\sqrt{\cos\sigma\cos\gamma}},$$

where σ is the angle of incidence, γ is the observation angle between the scattering direction and the normal, ϑ is the angle between the scattering direction and the specular reflection of the incident ray, g_s is the glossiness coefficient, and C is the scaling factor. The C does not actually affect the distribution of the paths because it cancels in the normalized importance function. A somewhat unexpected denominator appears in the formula so that for a narrow lobe (large g_s), the integral of the BDF is almost independent of the angle of incidence. One could take simply $\frac{1}{\cos\sigma}$, but then the BDF would not satisfy the Helmholtz reciprocity. Therefore, the symmetric expression $\frac{1}{\sqrt{\cos\sigma\cos\gamma}}$ is taken for a narrow lobe $\gamma \approx \sigma$ so that $\frac{1}{\sqrt{\cos\sigma\cos\gamma}} \approx \frac{1}{\cos\sigma}$ and the reciprocity is obeyed. As g_s decreases, the dependence of the BDF integral on the angle of incidence becomes stronger.

For the sake of simplicity, the surface properties of both ellipsoids are the same $g_{s,1} = g_{s,2} = g_s$. The larger the g_s value, the more glossy the surface is, becoming a mirror in the limit $g_s \to \infty$. For large g_s values, the BDF approximately looks like this:

$$f \approx C\frac{(1-\frac{\vartheta^2}{2})^{g_s}}{\sqrt{\cos\sigma\cos\gamma}} \approx C\frac{e^{-\frac{g_s\vartheta^2}{2}}}{\sqrt{\cos\sigma\cos\gamma}},$$

so that the BDF almost vanishes for $\vartheta > 3\sqrt{1/g_s}$ (as usual, the factor 3 arises due to the famous "3σ rule"). For safety, we allow even twice as large deviations of the ray from the specular direction setting for the constraints (24) and (26):

$$\max\vartheta = 6\sqrt{1/g_s}.$$

The fully absorbing diaphragm with the hole of radius r_a is placed in the plane passing through the coinciding focuses of the ellipsoids and perpendicular to their axes of rotation. For very small r_a values, only rays passing through the focuses of the ellipsoids are possible, i.e., the same as for mirror surfaces without a diaphragm.

Thus, the control parameters in our scene are the BDF width specified by g_s and the radius of the diaphragm r_a. By changing them, we can manipulate the constraints: for large

g_s values, the major role is played by the angular constraints due to the BDF shape, and for $r_a \to 0$, the spatial constraints are dominating. In both limiting cases, the allowed paths are the same as for pure mirrors, and the rays run through the focuses. Therefore, the light distribution converges to the same limit both for $r_a \to 0$ and for $g_s \to \infty$.

8.2. Description of Experiments

An isotropic source was used in the computational experiments. The virtual detector registers the density of the angular distribution of the rays hitting it (i.e., it is sensitive to the direction of the last, third, segment of the ray path). The density is proportional to the number of rays per solid angle element. This distribution is axially symmetric and depends only on the angle between the third segment and the Oz axis. Calculations were made for 2 $\times$ 2 combinations of scene parameters: $g_s = 100$ (fuzzy surface), g_s =10,000 (glossy suface), $r_a = 0.25$ (big hole) and $r_a = 0.025$ (small hole).

The importance function for our three-segment path is the product of two BDFs and cosines at the two reflection points. It is therefore determined by the directions of the segments and the surface normals. So, to calculate the gradient of the potential, we need the derivatives of the segment directions and the normals with respect to the state vector $\boldsymbol{X} = \{\boldsymbol{v}_1, \boldsymbol{\Omega}_1\}$ in our phase space. We used numerical (finite-difference) differentiation with respect to the four elements of $\boldsymbol{X}$ (two for the displacement of $\boldsymbol{v}_1$ in two perpendicular directions $\boldsymbol{p}_1, \boldsymbol{q}_1$ and two for the displacement of $\boldsymbol{\Omega}_1$ in two directions $\boldsymbol{p}'_1, \boldsymbol{q}'_1$ perpendicular to the specular direction $\boldsymbol{s}_1$). When calculating the constraint matrices, we need the derivatives of the angles ϑ_1, ϑ_2 (the angles between the scattered ray and the direction of mirror reflections for the 1st and the 2nd ellispoids) and the point $\boldsymbol{x}_a$ where the second segment intersects the plane of the diaphragm (Formulas (25) and (28)). They were also found by numerical differentiation with respect to the same four variables. Finally, the divergence term includes the derivatives of $\hat{T}\hat{T}^*$ also calculated by numerical differentiation.

For each set of scene parameters, we constructed the starting ray path of three segments by MCRT. This path was then used as the starting point of the Markov chain (numerical integration of the Langevin dynamics assisted by the Hastings rule). The time series were computed for a very large number of time steps (usually 10^{10}), and the histogram of the angular distribution of the direction of the last ray segment was accumulated. It estimates the angular density of radiation registered by the detector. Due to the axial symmetry of the scene, the histogram depends only on the angle between the segment and the scene axis in the range from 0° to 180° (zero direction is toward the light source).

Every 10,000 steps, the distribution density was estimated from the accumulated histogram by dividing the number of the rays in the cell by the cell's solid angle. Then, the distribution density was compared with the reference density calculated by a classic Monte Carlo method (the "reference" method below). The time series of the relative deviation of this density from the reference distribution gives the achieved accuracy, and we thus can estimate the rate of convergence.

Experiments were carried out for several algorithm modifications. Most of them (variants 1–6 in the list below) are modifications of Langevin Monte Carlo ray tracing (MALA). The variant 7 is the Metropolis–Hastings [9] algorithm (Formula (7)); essentially, it is MALA without the drift term. The variants 8 and 9 are for numerical integration of the Langevin dynamics without the Hastings rule. Thus, the following algorithm modifications were used in the experiments:

1. **full**: The full version of the Langevin Equation (2) with the gradient of potential, the divergent term and the preconditioning matrix. An adaptive time step is chosen based on (33).
2. **full, fix dt**: The full version of the Langevin Equation (2) with the gradient of potential, the divergent term and the preconditioning matrix. Fixed time step $\delta t = 0.01$.
3. **no div**: Truncated version of the Langevin equation without divergence term. The adaptive time step is chosen based on (33).

4. **no grad**: Truncated version of the Langevin equation without the gradient of potential. The adaptive time step is chosen based on (33).
5. **metropolis**: Truncated version of the Langevin equation without both the gradient of potential and the divergence term (in fact, this is the Metropolis algorithm with a Brownian mutation (Equation (7)). The time step is fixed.
6. **var dt, T = 1**: Full version of the Langevin Equation (2) without the preconditioning matrix ($\hat{T} = 1$). The adaptive time step is chosen based on (33).
7. **metropolis, T = 1**: Metropolis algorithm with Brownian mutation (7) without the preconditioning matrix ($\hat{T} = 1$). The time step is fixed.
8. **no MALA, T = 1**: Difference scheme (4) without the Hastings rule and without the preconditioning matrix ($\hat{T} = 1$). If the path is impossible after the shift by $\delta \boldsymbol{X}$, then the previous position of the point is retained. This is not the Hastings rule in full volume, because accepting the step is a deterministic decision, and the quantitative comparison of densities is not performed.
9. **no MALA** : Difference scheme (4) without the Hastings rule. If the path is impossible after a shift by $\delta \boldsymbol{X}$, then the previous position of the point is retained.

The program was written in C++ under Visual Studio 2017. All calculations were performed on processor Intel Core i7-11800H @ 2.30GHz with 64 Gb RAM. The calculation time of the one time step is the following:

- For the **«full»** with preconditioning matrix and adaptive δt, it took about 87 ms;
- For the **«full, fix dt»** with a fixed δt, it took 85 ms;
- For the **«var dt, T = 1»** with adaptive δt but without preconditioning matrix, it took 29 ms;
- For the pure Brownian motion with the matrix **«metropolis »** time step, it took 19 ms;
- For the pure Brownian motion without the matrix **«metropolis, T = 1 »**, it took 7 ms.

The computational complexity of the construction of the preconditioning matrix is seen from this timing.

8.3. Results of Experiments

We analyzed the angular distribution density $\rho(\alpha)$ of the directions of the last segment of the ray path on the detector where α is the angle between the last ray segment and Oz, so $\alpha = 0$ corresponds to the direction from the light source to the detector. We registered only three-segment paths. As shown in Figure 4, the ellipsoids in the scene are truncated, so in some neighborhood of the zero angle, such three-segment paths do not exist, and here, we obtain zero density values. Figure 5 shows the reference angular distribution density and the density calculated by the algorithm modifications 1 through 7. It is evident that for these variants, the angular distribution density $\rho(\alpha)$ almost coincides with the reference one with the exception of small fluctuations caused by stochastic noise.

Figure 6 shows the angular distribution density of the direction of the last segment for the difference schemes without the Hastings rule (modifications 8 and 9) in comparison with the reference. As can be seen from the graphs, the difference scheme without the Hastings rule gives an acceptable approximation in the case of $g_s = 100$ and $r_a = 0.25$, i.e., wide and smooth ravines (left upper graph in Figure 6). This case corresponds to low glossiness and wide aperture. In other cases, the modifications 8 and 9 give a large deviation from the reference.

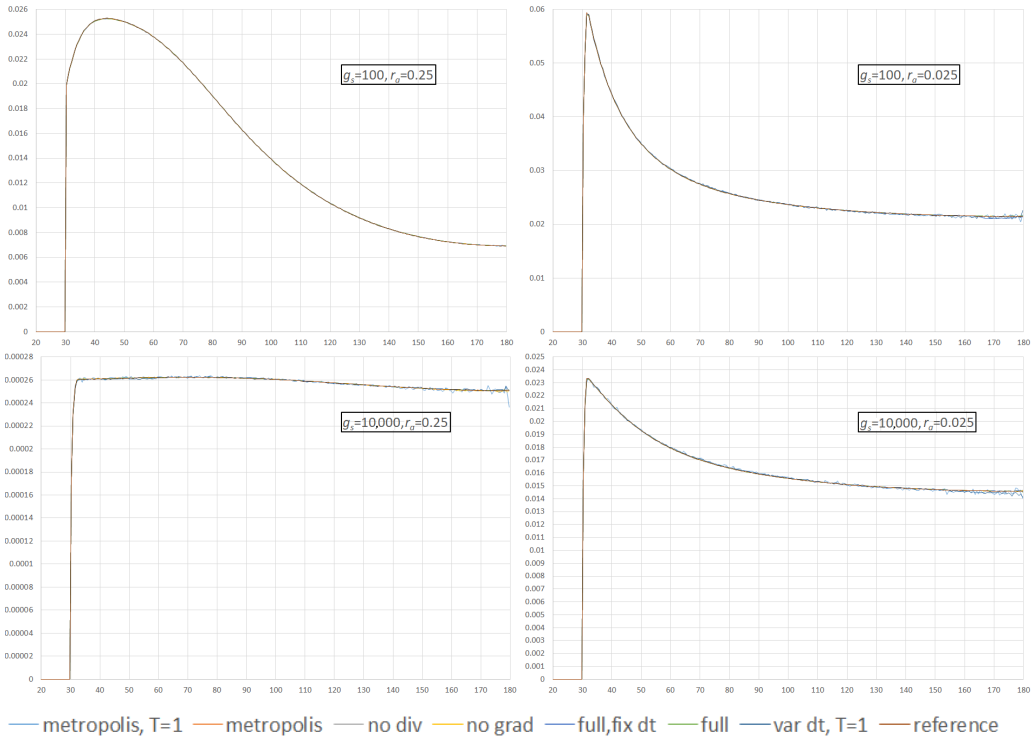


Figure 5. The density of the angular distribution of directions of the last segment for modifications 1–7 from 10^{10} time steps in comparison with the reference distribution.

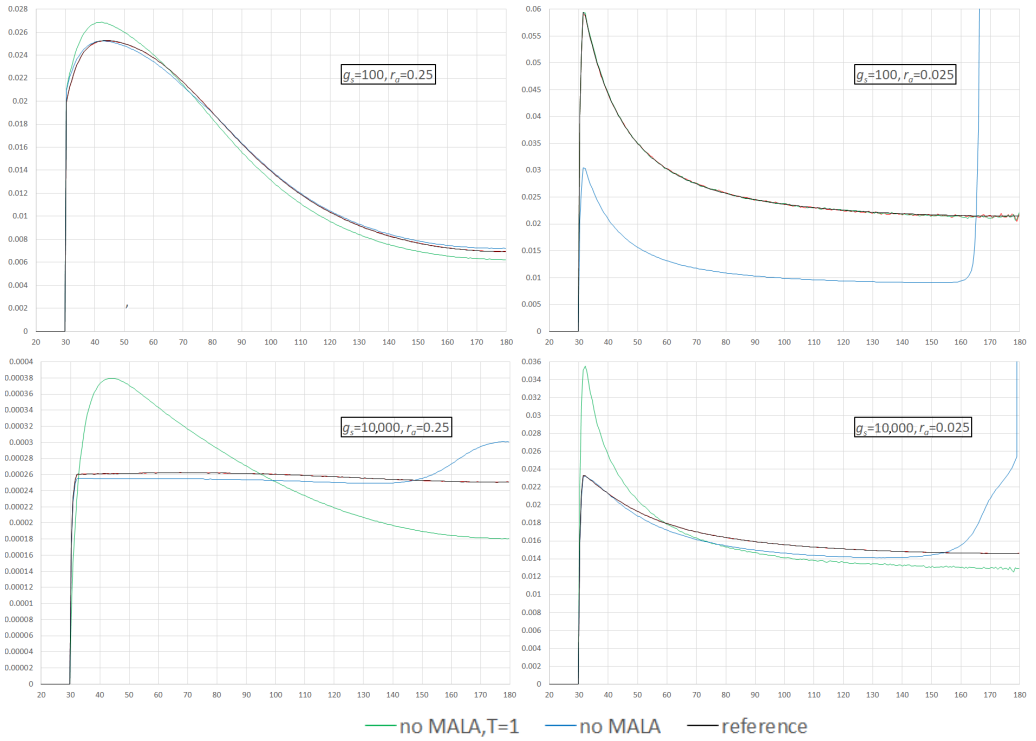


Figure 6. The density of the angular distribution of directions of the last segment for the modifications 8 and 9 from 10^{10} time steps in comparison with the reference distribution.

Figure 7 shows how the difference between the accumulated and the reference histograms decreases as the number of time steps increases. It is plotted against the number of time steps and against the processor time spent. For some algorithm modifications, we

need more time steps while each is calculated faster, so the convergence may be faster. This is so for the modifications 1–7, i.e., except the last two difference schemes without the Hastings rule. Note that the reference distribution was obtained numerically and therefore has some inaccuracy: about 0.0003 (so this is the low limit of the deviation). The deviation from the reference solution decreases over time for most variants. An exception is the variant of the full Langevin equation with a fixed time step «**full, fix dt**»; its error saturates and stops decreasing for the parameters $g_s = 100, r_a = 0.025$ (Figure 7, top row).

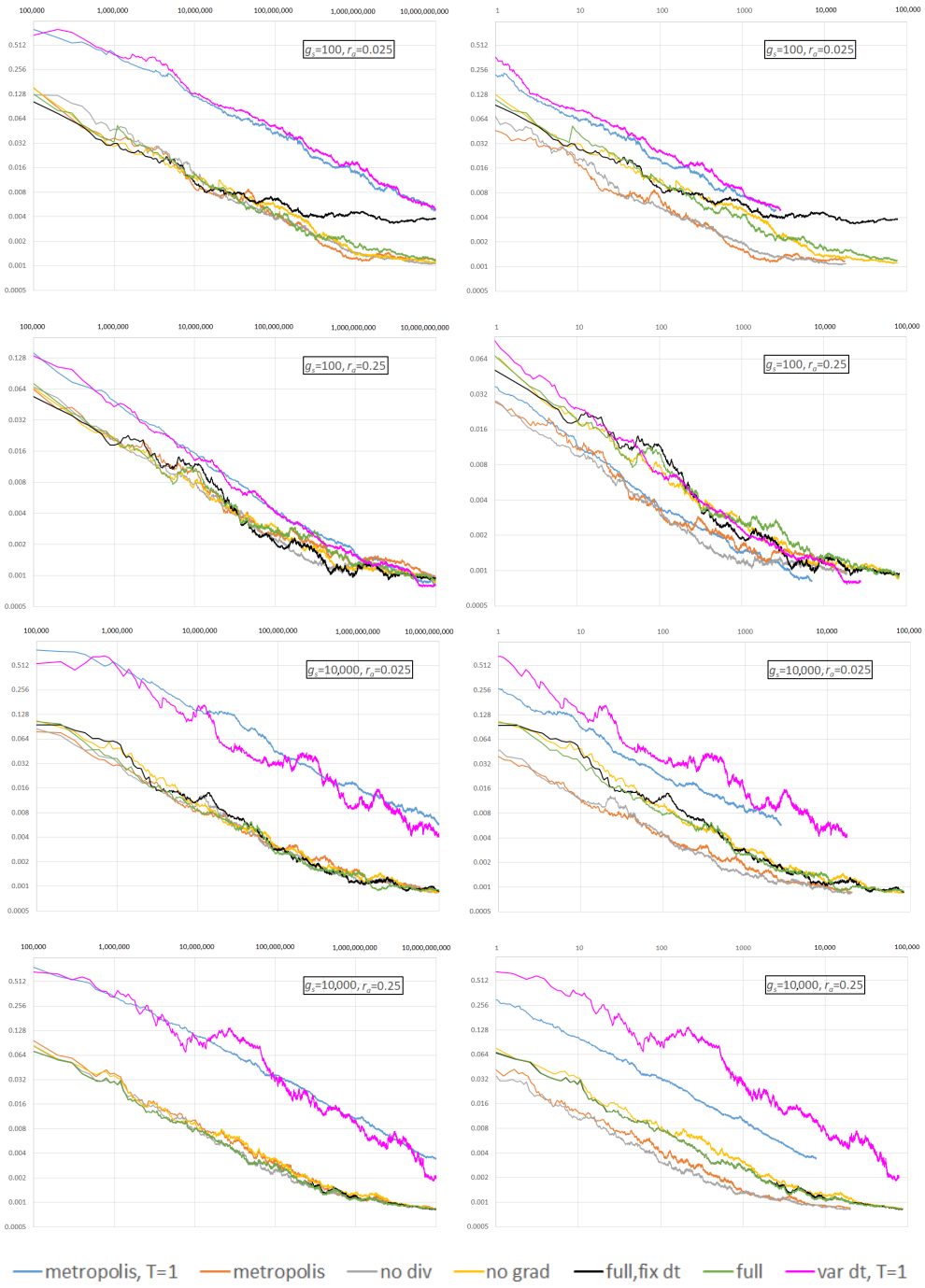


Figure 7. Dependence of the deviation of the accumulated distribution from the reference one on the number of steps (**left**) and on the spent processor time in seconds (**right**) for modifications 1–7.

But it is possible to calculate with a fixed time step discarding the divergent term. The «**metropolis** » modification does not choose δt adaptively because it does not have a drift

term (which enters the time step estimation). The deviation of the calculated distribution from the reference one gradually disappears. This is due to the fact that it is the divergent term that can reach very large values (sometimes $O(10^4)$) while the gradient is of the order of unity. Comparing the behavior of the graphs for the modifications «**no div** » and «**metropolis**» with «**full**», we see that discarding the divergence term (either only it or the drift term at all) improves convergence: approximately the same accuracy is achieved in significantly less computation time due to simplification of calculations. Comparing the «**no grad** » with «**full** » graphs, we can conclude that taking the gradient into account does not give a noticeable gain in accuracy for a given time.

Meanwhile, discarding the transformation matrix (the case $\hat{T} = 1$) significantly slows down the convergence, as can be seen from comparison of «**var dt, T = 1**» with «**full**» and «**metropolis, T = 1**» with «**metropolis**» in Figure 7.

When comparing the left (dependence on the number of steps) and the right (dependence on the computation time) graphs in Figure 7, we can see the computational complexity of the algorithm modifications. One modification can be executed faster, while it has the same rate of convergence in the number of steps. This is clearly seen, for example, for the modifications «**var dt, T = 1** » and «**metropolis, T = 1** » in the last row of graphs for $r_a = 0.25$ and g_s =10,000.

Figure 8 shows the dependence of the difference between the accumulative and the reference distributions on the number of time steps for the difference scheme without the Hastings rule (variants 8 and 9). The convergence graph of the Metropolis–Hastings method is also added for comparison. As expected, the accuracy is significantly worse without the Hastings rule («**metropolis, T = 1**»). The deviation quickly reaches the limiting value and stops decreasing. However, the error is still greater than for the methods with the Hastings rule (Figure 7) even at small calculation times.

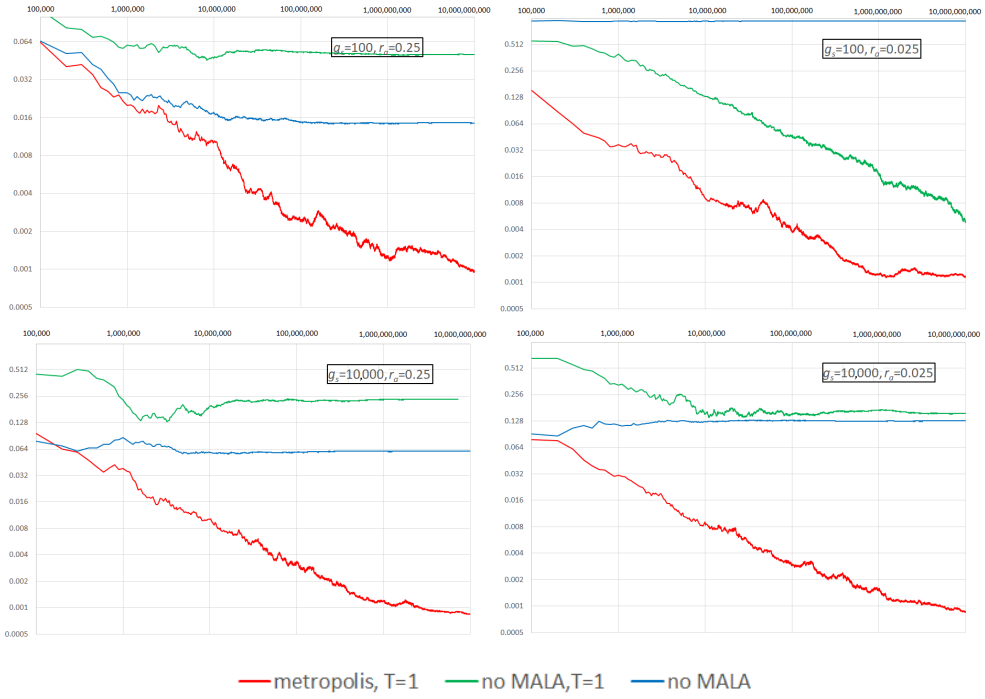


Figure 8. Dependence of the difference between the accumulated and reference distributions on the number of steps for the difference scheme without the Hastings rule.

Since variants 8 and 9 have the same computational complexity, their graphs of dependence on the calculation time are close to the graphs of dependence on the number of steps and are therefore not presented here.

9. Discussion and Conclusions

The Langevin Monte Carlo ray tracing (MALA) seems to be a very promising method for calculation of the global illumination of a scene. However, this method remains poorly studied, while its analysis can result in its optimization. This could significantly speed up the calculations without losing the quality of the result.

In this paper, the most computationally expensive operations of this method were analyzed. We conducted a series of computational experiments with the scene that allows us to demonstrate the advantages of Langevin Monte Carlo ray tracing to generate random light paths. Seven modifications of the MALA algorithm were tested for the accuracy of the result and the convergence rate. In each modification, we disabled one or another component of the method. This allowed us to reveal the contribution of each of them to the convergence rate. The results also demonstrated that all modifications of the MALA algorithm provide acceptable accuracy.

In addition, experiments were conducted with two modifications of difference schemes without the Hastings rule. Both modifications showed unsatisfactory accuracy and convergence. Therefore, they cannot be used for lighting simulation.

Computational experiments allowed us to draw the following conclusions:

1. The transformation matrix (preconditioning matrix) is mandatory. The convergence rate drops noticeably without it.
2. The transformation matrices calculated by the constraints on the path in the scene but without using the potential (i.e., the importance function) are quite sufficient and provide a good convergence rate.
3. The drift term can be excluded while the convergence of the angular distribution as a function of the number of steps remains approximately the same. Since in this case, the calculations are simplified and the step is made noticeably faster, then the convergence as a function of the computation time improves.
4. The adaptive choice of the time step δt is very important and is simply mandatory in the presence of the divergent term because in our scene, it can reach values of more than 10,000, while the other terms of the equation are all of the order of 1 and less.

Our main conclusion is that the greatest contribution to improving convergence is made by the preconditioning matrix that does not require calculating the gradient of the potential, unlike the other terms of the equation. Possibly, this allows us to significantly speed up the method.

Author Contributions: Conceptualization, V.F. and S.E.; methodology, A.V.; software, V.F. and A.N.; validation, A.V.; formal analysis, S.E.; investigation, S.E., A.N. and V.F.; data curation, S.E.; writing—original draft preparation, S.E., V.F. and A.V.; writing—review and editing, A.V. and V.G.; visualization, A.N.; supervision, A.V.; project administration, V.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The program code adapted for Linux is available in the repository https://github.com//FROL256/langevenmc, accessed on 1 November 2024. The data presented in this study are available within this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kajiya, J.T. The rendering equation. In Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH 86), Dallas, TX, USA, 18–22 August 1986; pp. 143–150. [CrossRef]
2. Veach, E. Robust Monte Carlo Methods for Light Transport Simulation. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 1998.
3. Frolov, V.A.; Voloboy, A.G.; Ershov, S.V.; Galaktionov, V.A. Light transport in realistic rendering: State-of-the-art simulation methods. *Program. Comput. Softw.* **2021**, *47*, 298–326. [CrossRef]
4. Brooks, S.; Gelman, A.; Jones, G.; Meng, X. *Handbook of Markov Chain Monte Carlo*; CRC Press: Boca Raton, FL, USA, 2011.

5. Sik, M.; Krivanek, J. Survey of Markov chain Monte Carlo methods in light transport simulation. *IEEE Trans. Vis. Comput. Graph.* **2018**, *26*, 1821–1840. [CrossRef] [PubMed]
6. Duane, S.; Kennedy, A.D.; Pendleton, B.J.; Roweth, D. Hybrid Monte Carlo. *Phys. Lett.* **1987**, *195*, 216–222. [CrossRef]
7. Luan, F.; Zhao, S.; Bala, K.; Gkioulekas, I. Langevin Monte Carlo rendering with gradient-based adaptation. *ACM Trans. Graph. (TOG)* **2020**, *39*, 140:1–140:16. [CrossRef]
8. Luan, F. Forward and Inverse Rendering with Gradient Based Optimizations. Ph.D. Thesis, Cornell University, Ithaca, NY, USA, 2021.
9. Veach, E.; Guibas, L.J. Metropolis Light Transport. In Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques SIGGRAPH'97, Los Angeles, CA, USA, 3–8 August 1997; pp. 65–76. [CrossRef]
10. Kaplanyan, A.S.; Hanika, J.; Dachsbacher, C. The natural-constraint representation of the path space for efficient light transport simulation. *ACM Trans. Graph. (TOG)* **2014**, *33*, 1–13. [CrossRef]
11. Kelemen, C.; Szirmay-Kalos, L.; Antal, G.; Csonka, F. A simple and robust mutation strategy for the Metropolis light transport algorithm. *Comput. Graph. Forum* **2002**, *21*, 531–540. [CrossRef]
12. Hachisuka, T.; Kaplanyan, A.S.; Dachsbache, C. Multiplexed Metropolis light transport. *ACM Trans. Graph. (TOG)* **2014**, *33*, 1–10. [CrossRef]
13. Wenzel, J. Light Transport on Path-Space Manifolds. Ph.D. Thesis, Cornell University, Ithaca, NY, USA, 2013.
14. Hanika, J.; Droske, M.; Fascione, L. Manifold next event estimation. *Comput. Graph. Forum* **2015**, *34*, 87–97. [CrossRef]
15. Bitterli, B.; Wenzel, J.; Novak, J.; Jarosz, W. Reversible jump Metropolis light transport using inverse mappings. *ACM Trans. Graph. (TOG)* **2017**, *37*, 1–12. [CrossRef]
16. Otsu, H.; Kaplanyan, A.S.; Hanika, J.; Dachsbacher, C.; Hachisuka, T. Fusing state spaces for Markov chain Monte Carlo rendering. *ACM Trans. Graph. (TOG)* **2017**, *36*, 1–10. [CrossRef]
17. Pantaleoni, J. Charted Metropolis light transport. *ACM Trans. Graph. (TOG)* **2017**, *36*, 1–14. [CrossRef]
18. Xifara, T.; Sherlock, C.; Livingstone, S.; Byrne, S.; Girolami, M. Langevin diffusions and the Metropolis-adjusted Langevin algorithm. *Stat. Probab. Lett.* **2014**, *91*, 14–19. [CrossRef]
19. Schreck, A.; Fort, G.; Moulines, E. Adaptive equi-tnergy sampler: Convergence and illustration. *ACM Trans. Model. Comput. Simul. (TOMACS)* **2013**, *23*, 1–27. [CrossRef]
20. Braak, C.J. A Markov chain Monte Carlo version of the genetic algorithm differential evolution: Easy Bayesian computing for real parameter spaces. *Stat. Comput.* **2006**, *16*, 239–249. [CrossRef]
21. Betancourt, M. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv* **2017**, arXiv:1701.02434.
22. Homan, M.D.; Gelman, A. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.* **2014**, *15*, 1593–1623.
23. Girolami, M.; Calderhead, B.; Chin, S.A. Riemannian manifold Hamiltonian Monte Carlo. *arXiv* **2009**, arXiv:0907.1100.
24. Li, T.-M.; Lehtinen, J.; Ramamoorthi, R.; Wenzel, J.; Durand, F. Anisotropic Gaussian mutations for Metropolis light transport through Hessian-Hamiltonian dynamics. *ACM Trans. Graph. (TOG)* **2015**, *34*, 1–13. [CrossRef]
25. Li, T.-M. Differentiable Visual Computing. Ph.D. Thesis, MIT, Cambridge, MA, USA, 2019.
26. Hogan, R.J. Fast reverse-mode automatic differentiation using expression templates in C++. *ACM Trans. Math. Softw. (TOMS)* **2014**, *40*, 1–16. [CrossRef]
27. Rioux-Lavoie, D.; Litalien, J.; Gruson, A.; Hachisuka, T.; Nowrouzezahrai, D. Delayed rejection Metropolis light transport. *ACM Trans. Graph. (TOG)* **2020**, *39*, 1–14. [CrossRef]
28. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
29. Andrieu, C.; Thoms, J. A tutorial on adaptive MCMC. *Stat. Comput.* **2008**, *18*, 343–373. [CrossRef]
30. Vasil'eva, A.B.; Butuzov, V.F. *Asymptotic Methods in the Singular Perturbation Theory*; High School Press: Moscow, Russia, 1990. (In Russian)
31. Enzo Orlandini Lectures. Available online: https://userswww.pd.infn.it/~orlandin/fisica_sis_comp/fokker_planck.pdf (accessed on 24 October 2024).
32. Zwanzig, R. Nonlinear generalized Langevin equations. *J. Stat. Phys.* **1973**, *9*, 215–220. [CrossRef]
33. Cornell University. Lecture 8: Accelerating SGD with Preconditioning and Adaptive Learning Rates. CS4787—Principles of Large-Scale Machine Learning Systems. Available online: https://www.cs.cornell.edu/courses/cs4787/2019sp/notes/lecture8.pdf (accessed on 3 October 2024).
34. Amari, S.; Ba, J.; Grosse, R.; Li, X.; Nitanda, A.; Suzuki, T.; Wu, D.; Xu, J. When does preconditioning help or hurt generalization? *arXiv* **2020**, arXiv:2006.10732.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.