Utrecht University

Gradient-Domain Volume Rendering

Guowei Lu
ICA-6374395

Supervisors

Jerry Guo, MSc
Prof. Dr. Alexandru C. Telea
Prof. Dr. Elmar Eisemann
Dr. ing. Jacco Bikker

August 21, 2020

Abstract

In this thesis, we present a Gradient-Domain method for volume rendering using the MLT algorithm. One contribution is that we compute the gradient of the light path in the medium while taking into account the medium properties, including transmittance, phase function, and the geometry factor. With this gradient method, when we mutate a new path from the existing path, we estimate its radiance at negligible cost. The second contribution is the gradient strategy that can be seamlessly integrated into the Multiplexed Metropolis Light Transport (MMLT) to exploit correlated paths in the medium. A progressive version of this algorithm handles a wide variety of situations by reducing the exploration scope in a medium gradually. We demonstrate that our method yields a superior result compared to the MMLT alone.

Keywords: gradient, Volume Rendering, Metropolis Light Transport, Ratio Tracking

1 Introduction

The world is filled with media that attenuate and scatters light when it travels from the light source, hits surfaces, and finally reaches the camera. Simulating this transport is necessary for rendering many natural phenomena and special effects, such as clouds, smoke, and fog. Stochastic evaluation of the rendering equation, with many random paths between the camera and the lights is commonly used to estimate real-world light transport in a virtual scene. However, when considering the influence of media, volume rendering still provides its own set of unique challenges. Compared to surface-only scenes, vertices of the random paths can be anywhere in a medium. So, the collection of possible light paths in media is larger and higher-dimensional, and light path construction is accordingly more expensive.

A lot of work has focused on filling the gap between surface and volume transport simulation. Unfortunately, it is still a significant challenge to quickly find all important light paths in scenes that include participating media. On the other side, medium density typically is either constant or changes slowly; the density at each position in the medium has a strong correlation with its neighbors. Intuitively, once we get the radiance at a given position in a medium, we can estimate the radiance distribution in the direct vicinity. Our method is based on this observation: if the existing path is important, we generate neighbor paths by perturbing the path randomly, these new paths are likely to be important; then, we estimate the radiance of these new paths by exploring the correlation among neighbor paths. We can do so with limited computation, as we avoid the expensive construction and radiance evaluation of entirely new paths.

For rendering purposes, a medium is defined by a small set of properties: absorption and scattering coefficients, a phase function and emission. The Volume Rendering Equation (VRE) [PKK00] describes the behavior of light in a medium that absorbs, emits, and scatters radiation. Once we use the VRE to express the radiance of a light path mathematically, it tells us how the radiance along this path distributes at each path vertex.

We compute the spatial gradient of this equation with respect to one vertex. The gradient measures the change of the radiance caused by perturbing this vertex locally. Our work is an extension of the work by Jarosz et al. [JDZJ08], who derive the gradients for paths of inscattered light within a medium. Our work is also based on Metropolis Light Transport (MLT) [Vea97], a Markov Chain Monte Carlo method (MCMC) which constructs correlated paths iteratively by perturbing existing paths. We efficiently estimate the radiance of the perturbed paths using the path gradient.

The goal of this thesis is thus to explore new paths at low cost. We present a method to compute the gradient of light paths based on the VRE. Then, we design a gradient mutation strategy for MLT to apply this gradient algorithm to estimate the contribution of perturbed paths. Concretely, our contributions are:

- *Gradient Algorithm.* Based on the VRE, we provide a method to calculate the gradient of the light path with respect to the vertex as long as the vertex is in the medium. We consider all related variables in the equation, including transmittance, phase function, and geometry factor. The gradient allows us to efficiently estimate the radiance of the new path.
- *Gradient Mutation Strategy.* We present an MLT mutation strategy based on the gradient algorithm, which operates in *path space*: the set of all possible light paths of finite length [Vea97]. Currently,

many strategies are based on Primary Sample Space (PSS) [KSKAC02]. Path space is good at local exploration, while PSS is good at global exploration. We propose a method to combine them and improve convergence by applying the gradient algorithm.

- *Radius reduction scheme.* The gradient algorithm is biased, where the bias is proportional to the radius used for the perturbations. Inspired by progressive photon mapping [HOJ08], we progressively reduce the radius in our gradient mutation strategy to make our algorithm consistent.

Mathematically, our approach is an implementation of Taylor-series expansion to the VRE mathematically. We compute the gradient of the light path by applying the gradient algorithm, generate correlated paths by applying the gradient mutation strategy, and estimate the contributions of these new paths. Meanwhile, we provide the radius-reduction scheme to reach the optimal balance between bias and variance.

2 Related Work

Our work builds upon several existing algorithms. We briefly discuss these, as well as Gradient-Domain Rendering and the difference between this approach and the use of gradients in our MLT mutations.

Volume Rendering Equation. Media consist of small particles. If the particles in the medium are evenly distributed, the density is constant. This is referred to as a *homogeneous medium*; otherwise, the medium is *inhomogeneous*. Three events affect the propagation of radiance in the medium: *absorption*, *emission*, and *scattering*. *Transmittance* gives the fraction of radiance that is transmitted along a ray, because radiance is reduced due to absorption and out-scattering. Meanwhile, a scattering event scatters radiance into a different direction. Media that are *isotropic* have an equal probability of scattering incoming light in any direction; otherwise, the medium is *anisotropic*. A *Phase function* describes this angular distribution of scattered radiance at a given point.

We can add up all constituent terms to form the *Radiative Transfer Equation* (RTE) [Cha13], which describes the spatial variation due to absorption, emission, and scattering. The VRE is related to the RTE and can be interpreted as a generalization of the Rendering Equation (RE) [Kaj86], taking media into account. Moreover, we can extend from the RE to the VRE by getting the integral representation of RTE.

Transmittance estimator. For homogeneous media, the density is constant, and transmittance is only influenced by the length of the ray in the medium. For inhomogeneous media, *Ray Marching* is one standard method for computing transmittance. This method requires many fine steps in high-resolution data and results in unpredictable, systematic bias. Another method named *Ratio Tracking* provided by Novak et al. [NSJ14] is an unbiased transmittance estimator, which however yields a higher variance than Ray Marching. The gradient of transmittance for the Ray Marching method is provided by Jarosz et al. [JDZJ08].

MLT for participating media. Since the introduction of the original MLT [LKL$^+$13], much work has been done to provide specialized mutation strategies. One technique based on *Primary Sample Space* (PSSMLT) [KSKAC02] significantly simplifies the MCMC process compared to the original MLT methods. Based on PSSMLT, another technique called *Multiplexed MLT* (MMLT) [HKD14] bridges multiple importance sampling (MIS) [Vea97] and MCMC by providing an extra state dimension to choose a single strategy, instead of invoking all connection strategies. So, MMLT spends more computation on effective strategies. Another interesting algorithm, named *Energy redistribution path tracing* [CTE05], is a hybrid algorithm that uses MLT strategies in a standard MC method.

MLT is extended to incorporate volume rendering by Pauly et al. [PKK00]. One strategy, named *Propagation Perturbation*, is provided for volume rendering. This strategy aims to move the last vertex (assuming it is in the medium) along the incoming ray, with the distance and direction randomly generated, re-connects it with the camera, and builds a new path, which is a path space mutation strategy.

Radius reduction. Density estimation in Photon mapping [Jen96] is biased. However, it builds a connection between the number (N) of photons and the radius (r) of the search region. When $N \to \infty$, $r \to 0$, and the radiance estimate converges to the correct solution, which makes density estimation consistent.

The radius of density estimation in progressive photon mapping [HOJ08] is reduced in each iteration. A similar scheme is provided for the Vertex Connection and Merging algorithm by Georgiev et al. [GKDS12]. This scheme is used to reach a balance between variance and bias.

Relevance to Gradient-Domain Rendering. Gradient-Domain Rendering is an algorithm that directly computes image gradients and reconstructs the final image from the gradients. This algorithm can be applied in MLT [LKL+13] and standard MC path tracing [KMA+15]. There are two significant differences between them. Our gradient is in path space, which is 3-dimensional, while the Gradient-Domain Rendering is in image space, which is 2-dimensional. Another difference is that our algorithm uses the gradient to estimate radiance. However, in Gradient-Domain Rendering, the radiance of pairs of paths is computed to measure the gradient.

Shift mapping is a path space mapping function that generates a correlated path from a provided input path [HGP+19], which reduces variance by using path correlation. In our gradient mutation strategy, we use correlation as a sampling method to construct new path.

The structure of the remainder of this thesis is as follows. We introduce the basic concepts needed in Section 3. In Section 4, we provide our gradient algorithm to compute the gradient of the light path, including transmittance and phase function. We apply the gradient algorithm in MMLT and implement a path space mutation with a radius reduction scheme in Section 5. In Section 6 and Section 7, we present and discuss our results and propose further research.

3 Background

While our Gradient-Domain Volume Rendering method is conceptually simple, a deep understanding of the VRE and light transport is the premise to apply this algorithm. The gradient algorithm is derived from the formulation of the VRE, and we implement it in different light transport algorithms such as *standard Path Tracing*, *Bidirectional Path Tracing* (BDPT), and MLT.

In Table 1, we provide an overview of the notation used in this thesis.

Symbol	**Meaning**
x_i	a point
$\bar{x}_i$	a light path
σ_a, σ_s,σ_t	absorption, scattering and extinction coefficient
$T_r(x_i \leftrightarrow x_{i+1})$	transmittance
$f_p(x, \omega, \omega')$	phase function
$G(x_i \leftrightarrow x_{i+1})$	geometry factor
$L_e(x, \omega)$, $L_s(x, \omega)$	emitted and in-scattering radiance

Table 1: *Notation.*

3.1 Volume Light Transport

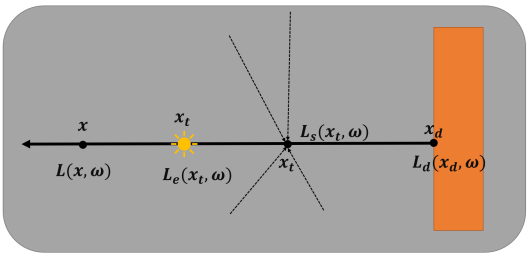


Figure 1: *The Volume Rendering Equation (VRE) visualized.*

Figure 1 visualizes the propagation of radiance from x_d to x in the medium. The radiance emitted or reflected at x_d is $L_d(x_d, \omega)$. The transmittance $T_r(x_d \leftrightarrow x)$ is the reduction of radiance $L_d(x_d, \omega)$ between x_d and x due to absorption and out-scattering in the medium:

$$L_{incident}(x, \omega) = T_r(x_d \leftrightarrow x) L_d(x_d, \omega), \tag{1}$$

Meanwhile, along this ray, at any position x_t, in-scattering $L_s(x_t, \omega)$ is the contribution due to scattering towards x from other directions, with the proportionality factor being the scattering coefficient σ_s. In-scattering may happen at any position between x_d and x. We need to consider the reduction by the transmittance up to x using the integral:

$$L_{in\_scattering}(x, \omega) = \int_{x_d}^{x} T_r(x_t \leftrightarrow x) \sigma_s(x_t) L_s(x_t, \omega) dx_t, \tag{2}$$

Finally, if there are luminous particles along this ray, at any position x_t, the emission factor $L_e(x_t, \omega)$ is the contribution due to emission. Like the radiance at x_d, emitted radiance is reduced while travelling though the medium.:

$$L_{emission}(x, \omega) = \int_{x_d}^{x} T_r(x_t \leftrightarrow x) L_e(x_t, \omega) dx_t, \tag{3}$$

We obtain the Volume Rendering Equation (VRE) by assembling Eq. 1, Eq. 2, and Eq. 3:

$$\begin{aligned} L(x, \omega) &= L_{incident}(x, \omega) + L_{in\_scattering}(x, \omega) + L_{emission}(x, \omega) \\ &= \int_{x_d}^{x} T_r(x_t \leftrightarrow x) \left[L_e(x_t, \omega) + \sigma_s(x_t) L_s(x_t, \omega) \right] dx_t + T_r(x_d \leftrightarrow x) L_d(x_d, \omega) \end{aligned} \tag{4}$$

In this thesis, we ignore the contribution from emission. To solve this path integral numerically, we construct paths by sampling path vertices sequentially. So, we get the measurement contribution function of Eq. 4 to describe the light path in the medium:

$$\begin{aligned} f(\bar{x}) &= L_e(x_0 \rightarrow x_1) \left(\prod_{i=0}^{k-1} f(\bar{x}_i) \right) W_e(x_{k-1} \rightarrow x_k) \\ f(\bar{x}_i) &= \begin{cases} G(x_i \leftrightarrow x_{i+1}) T_r(x_i \leftrightarrow x_{i+1}), & i = k-1 \\ G(x_i \leftrightarrow x_{i+1}) T_r(x_i \leftrightarrow x_{i+1}) \sigma_s(x_{i+1}) f_p(x_i \leftrightarrow x_{i+1} \leftrightarrow x_{i+2}), & otherwise \end{cases} \end{aligned}, \tag{5}$$

where:

$L_e(x_0 \rightarrow x_1)$ is the emittance of a light source

$W_e(x_{k-1} \rightarrow x_k)$ is the sensor sensitivity function.

Based on Eq. 5, there are three major factors that influence the result: the transmittance $T_r(x_i \leftrightarrow x_{i+1})$, the phase function $f_p(x_i \leftrightarrow x_{i+1} \leftrightarrow x_{i+2})$, and the geometry factor $G(x_i \leftrightarrow x_{i+1})$.

Transmittance. In homogeneous media, the extinction coefficient σ_t in the medium is constant, and the equation of transmittance along a path from $\bar{x}_i$ to $\bar{x}_{i+1}$ is:

$$T_r(x_i \leftrightarrow x_{i+1}) = e^{-\sigma_t \| x_{i+1} - x_i \|}, \tag{6}$$

where:

σ_t is the extinction coefficient of the medium, which is the sum of absorption coefficient σ_a and scattering coefficient σ_s.

We can apply Ratio Tracking to compute the transmittance when the medium is inhomogeneous. In this case, the transmittance is:

$$T_r(x_i \leftrightarrow x_{i+1}) = \prod_{j=0}^{k} \left(1 - \frac{\sigma_t(x_j)}{\bar{\sigma}}\right), \tag{7}$$

where:

x_j is a sampling point along this ray, which is generated randomly

$\sigma_t(x_j)$ is the extinction coefficient at x_j

$\bar{\sigma}$ is the maximum value of σ_t in the medium.

Phase function. The phase function of isotropic media is constant over the unit sphere:

$$f_p(x_i \leftrightarrow x_{i+1} \leftrightarrow x_{i+2}) = \frac{1}{4\pi}, \tag{8}$$

A widely used phase function for *anisotropic* media is the Henyey-Greenstein phase function [Wit77]:

$$f_p(x_i \leftrightarrow x_{i+1} \leftrightarrow x_{i+2}) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 - 2g(\cos\theta))^{3/2}}, \tag{9}$$

where:

θ is $\angle x_i x_{i+1} x_{i+2}$

g controls the distribution of scattered light, in the range $(-1, 1)$.

Geometry factor. The geometry factor in media is:

$$G(x_i \leftrightarrow x_{i+1}) = \frac{1}{\|x_{i+1} - x_i\|^2} = \frac{1}{A}, \tag{10}$$

where:

A is the squared distance between x_{i+1} and x_i.

3.2 Bidirectional Path Tracer

Bidirectional path tracing (BDPT) is an algorithm in which partial paths are generated from the eye and from the camera. The final set of paths $\bar{x}_i$ is then generated by connecting the vertices of these partial paths. MIS weights are used for these connections. This is illustrated in Figure 2. The blue line is the connection ray. We have four techniques to build a path with three vertices: (1) an implicit path, where a ray generated from a point on the surface happens to intersect a light source; (2) an explicit path, with a connection ray generated from a point on the surface to the light source; (3) path from the light source with a connection ray generated from a point on the surface to the camera; (4) path from the light source where a ray generated from a point on the surface happens to intersect the camera. We apply MIS to combine the contributions from all these four paths with different weights. This algorithm is also available for volume rendering [LW96].

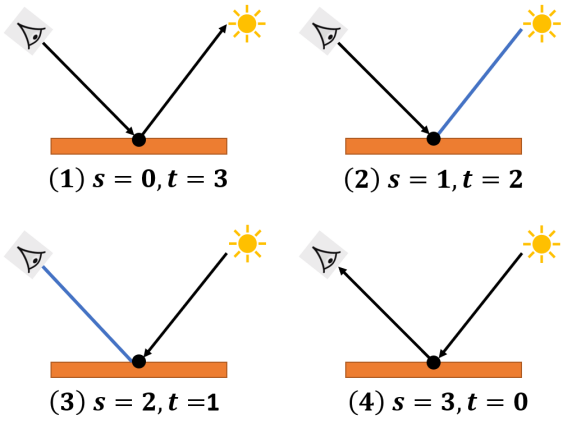


Figure 2: *Four different techniques in which BDPT can build a path consisting of three vertices.*

The equation to evaluate the contribution in BDPT is:

$$L \approx \sum_{i=1}^{N} \omega(\bar{x}_i) \frac{f(\bar{x}_i)}{p(\bar{x}_i)}, \tag{11}$$

where:

$f(\bar{x}_i)$ is the radiance of the path $\bar{x}_i$ including eye path and light path

$p(\bar{x}_i)$ is the probability density of the path $\bar{x}_i$

$\omega(\bar{x}_i)$ is the weight of the path $\bar{x}_i$.

3.3 Metropolis Light Transport

Instead of generating paths independently, MLT generates a sequence of correlated paths. Each sample $\bar{x}_i$ is obtained by a random change to $\bar{x}_{i-1}$. In this random walk, $\bar{x}_i$ only depends on $\bar{x}_{i-1}$, which is a *Markov Chain*. In order to generate the correct distribution, a *tentative function* T is provided. $T(\bar{x}_i|\bar{x}_{i-1})$ gives the probability density of a transition to $\bar{x}_i$, when the current state is $\bar{x}_{i-1}$. Then, an *acceptance probability* is defined, which gives the probability of accepting the new state $\bar{x}_i$, or not, to ensure the distribution of states approaches the target distribution in the limit. If the distribution is in equilibrium, the transition density between any two states must be equal:

$$f(\bar{x}_{i-1})T(\bar{x}_i|\bar{x}_{i-1})\alpha(\bar{x}_i|\bar{x}_{i-1}) = f(\bar{x}_i)T(\bar{x}_{i-1}|\bar{x}_i)\alpha(\bar{x}_{i-1}|\bar{x}_i), \tag{12}$$

where:

$f(\bar{x}_{i-1})$ and $f(\bar{x}_i)$ are the scalar contribution of the light path $\bar{x}_{i-1}$ and $\bar{x}_i$

T is the tentative function

α is the acceptance probability.

Eq. 12 is sufficient to maintain equilibrium distribution of the Markov Chain. This property is called *detailed balance*, which is an important requirement to follow. *Metropolis-Hastings sampling* is an algorithm from

the family of MCMC algorithms. The pseudocode of Metropolis-Hastings sampling is:

Algorithm 1: Metropolis-Hastings()

```
1  x̄ = generateRandomPath();
2  for i = 1 to n do
3  |   x̄' = mutate(x̄);
4  |   α = accept(x̄,x̄');
5  |   record(x̄,1-α);
6  |   record(x̄',α);
7  |   if random() < α then
8  |   |   x̄ = x̄';
9  |   end
10 end
```

First, we generate a path randomly (line 1). We then start to mutate n new paths in the loop. In each iteration, we apply the mutation strategy to generate a new path $\bar{x}'$ from the current path $\bar{x}$ (line 3). We get the acceptance probability value of $\bar{x}$ and $\bar{x}'$ (line 4); then, we add the contributions in both locations $\bar{x}$ and $\bar{x}'$ (line 5,6); finally, we get a random number to accept or reject the new path $\bar{x}'$. If we accept the new path $\bar{x}'$ (line 7), we set this new path $\bar{x}'$ as the latest state instead of $\bar{x}$ (line 8).

We apply the expected value technique in MLT. This optimization replaces a random variable by its expected value, which reduces variance by doing part of the integration analytically [KW86].

$$h\left(\bar{x}_{i+1}\right) \;=\; f\left(\bar{x}_{i+1}\right)\alpha\left(\bar{x}_{i+1}\,|\bar{x}_i\right) \;+\; f\left(\bar{x}_i\right)\left[1-\alpha\left(\bar{x}_{i+1}\,|\bar{x}_i\right)\right] \;\;, \tag{13}$$

where:

α is the acceptance probability function, returns the value of the acceptance portion

$f(\bar{x}_i)$ is the contribution of the path $\bar{x}_i$

$h(\bar{x}_{i+1})$ is the contributions we add in the $i+1$ iteration.

Primary Sample Space. The original mutation strategies in MLT are based on path space. A path space strategy contains the structure of the light path and recreates the new paths. We can use different strategies in different scenarios, but it is difficult to evaluate the tentative function $T(\bar{x}_i|\bar{x}_{i-1})$, which guarantees a correct distribution. PSSMLT explores paths indirectly, as shown in Figure 3. It generates a vector $\bar{u}$ of random numbers, which is used for sampling in path space and constructing a path $\bar{x}$. We define this vector as a unit hypercube $\bar{\mu}$. Each path is defined by this vector to get a simplified path integral over $\bar{\mu}$. This hypercube set μ is called Primary Sample Space (PSS).

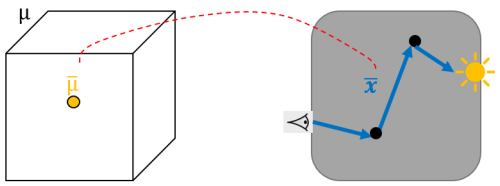


Figure 3: *Primary Sample Space MLT.*

4 Gradient Algorithm

The basis of our method is the gradient algorithm. We need to compute the gradient of each term in the VRE, including the transmittance, the phase function, and the geometry factor. In this section, we provide

(1) the physical meaning of the gradient in the VRE; (2) the solution to compute the gradient of each term; (3) a comprehensive method to compute the gradient of the VRE as a whole.

4.1 Overview

We illustrate our method using a $2D$ medium, as shown in Figure 4. There are many particles in this medium, and we want to measure the density of particles at any given position (x_t, y_t). We denote $f(x_i, y_j) = p$, where p is the number of particles in grid (x_i, y_j).

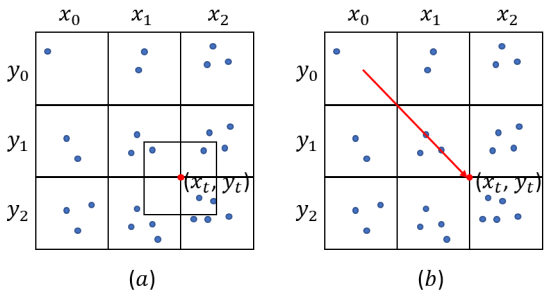


Figure 4: *A medium in 2D.*

There are two distinct methods to solve this problem, as shown in Figure 4. We can build a unit grid cell with the centre at (x_t, y_t), then, we count the number of particles in the cells that overlap this cell with their weights respectively:

$$f(x_t, y_t) = \frac{1}{4} f(x_1, y_1) + \frac{1}{4} f(x_1, y_2) + \frac{1}{4} f(x_2, y_1) + \frac{1}{4} f(x_2, y_2) = 4 \tag{14}$$

Alternatively, we measure the gradient between neighbor grid cells in $x-$ and y direction. The unit gradient value is a vector $v_{grad} = (1, 1)$. The length of the red line is $v_{dis} = (1.5, 1.5)$. Suppose $f(x_0, y_0)$ is known, then the density at (x_t, y_t) is:

$$f(x_t, y_t) = f(x_0, y_0) + \langle v_{grad}, v_{dis} \rangle = 4 \tag{15}$$

For any given position (x_t, y_t), using the first method, we need to measure the values of 4 grids and their weights. Using the second method,, we only measure the value of $f(x_0, y_0)$ and its gradient v_{grad} once, then, we obtain the result at any position (x_t, y_t) by calculating the vector $v_{dis} = (x_t, y_t) - (x_0, y_0)$. The gradient represents the change among neighbor grids including direction and magnitude, which is easy to implement and requires less computation, at the cost of precision loss. The general expression of this method is the first-order Taylor expansion:

$$f(x_t) = f(x_0) + \langle f'(x_0), x_t - x_0 \rangle , \tag{16}$$

where:

x_0 and x_t are the positions in the medium, which are vectors

$f(x)$ is a function we need to apply

$f'(x_0)$ is the gradient of $f(x)$ at position x_0

$x_t - x_0$ is the vector from x_0 to x_t.

4.2 Gradients in the VRE

Now, we list the gradients for the individual parts in the VRE with their physical meaning.

Gradient of the Transmittance. The transmittance in the medium is the reduction factor from absorption and out-scattering of a ray. This is illustrated in Figure 5. We measure the densities in the medium at different sampling points. Based on Eq. 7, we sum these densities to compute the transmittance of this ray.

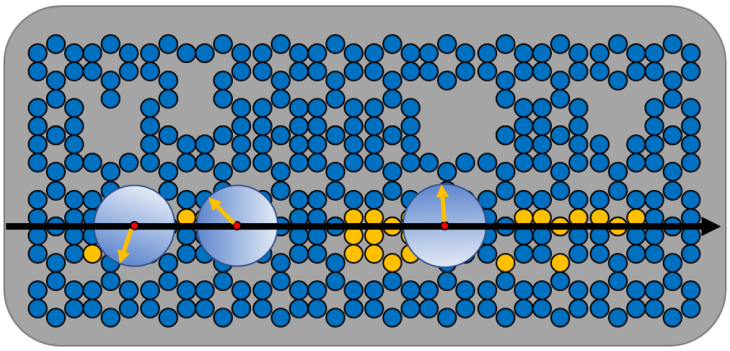
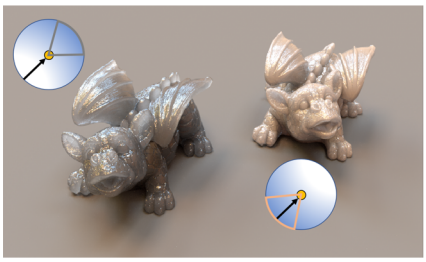


Figure 5: *Transmittance in a medium.*

Meanwhile, at each sampling point, the gradient of the density (the gradient circle in Figure 5) tells us the direction in which the density rises most quickly. The magnitude of the gradient determines how fast the density rises in that direction. Based on Eq. 7, we sum the gradient of density at all these sampling points as the transmittance gradient of the ray, which shows how the transmittance changes when we perturb this ray.

Gradients of the phase function and geometry factor. Shown in Figure $6(a)$, the light is from the same direction, the one on the left is strong backward scattering and the one on the right is strong forward scattering. The phase function describes the fraction of radiance scattered from the input direction $\overrightarrow{x_i x_{i+1}}$ to the output direction $\overrightarrow{x_{i+1} x_{i+2}}$, which is illustrated in Figure $6(b)$.

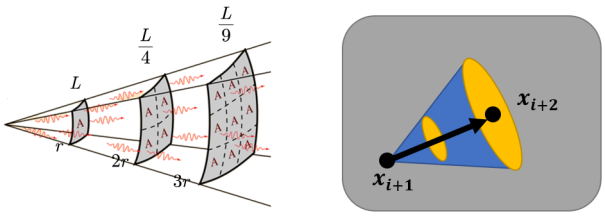


(a) phase functions in different media (b) phase function

Figure 6: *The phase function of a medium. Image courtesy of Pharr et al.*

Figure 7 shows the geometry factor, which is the derivative of the projected solid angle at vertex x_{i+1} with respect to area at vertex x_{i+2}.

When we perturb vertex x_{i+2}, the gradient of the phase function shows how the fraction of radiance changes due to the change of the angle $(x_i \leftrightarrow x_{i+1} \leftrightarrow x_{i+2})$. The gradient of the geometry factor measures the change of area the ray will cover. We see that the offset perpendicular to the vector $\overrightarrow{x_{i+1} x_{i+2}}$ causes the change to the angle, while the offset along the vector $\overrightarrow{x_{i+1} x_{i+2}}$ causes the change of the area. The gradient of these two terms tells us how the change of direction and distance affects the change of radiance.



(a) solid angle and area (b) geometry factor

Figure 7: *The geometry factor of a medium. Image courtesy of HyperPhysics.*

We have an understanding of the gradient and the physical meanings of the transmittance, phase function, and geometry factor. In the next sections, we provide the methods to compute their gradients respectively.

4.3 Transmittance

Homogeneous Media. In a homogeneous medium, the extinction coefficient σ_t is constant, and the transmittance is influenced by the distance of the ray in the media. The gradient of transmittance is derived from Eq. 6:

$$\begin{aligned} \nabla\left(T_r\left(x_i \leftrightarrow x_{i+1}\right)\right) &= -\sigma_t e^{-\sigma_t \|x_{i+1} - x_i\|} \nabla\left(\|x_{i+1} - x_i\|\right) \\ &= -\sigma_t T_r\left(x_i \leftrightarrow x_{i+1}\right) \nabla\left(\|x_{i+1} - x_i\|\right) \end{aligned} \tag{17}$$

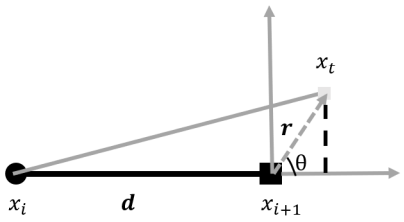


Figure 8: *Influence of distance change on transmittance.*

When we move vertex x_{i+1} to any position x_t (Figure 8), the distance $\|\overrightarrow{x_i x_t}\| = \sqrt{d^2 + 2dr\cos\theta + r^2}$ determines the value of transmittance, and we can get the change of the distance by the offset $\overrightarrow{x_{i+1}x_t}$:

$$\nabla\left(T_r\left(x_i \leftrightarrow x_{i+1}\right)\right) = -\sigma_t T_r\left(x_i \leftrightarrow x_{i+1}\right) \frac{\overrightarrow{x_i x_{i+1}}}{\|\overrightarrow{x_i x_{i+1}}\|} \tag{18}$$

Inhomogeneous Media. Based on Eq. 7, we determine the gradient of $T_r(x_i \leftrightarrow x_{i+1})$. This is illustrated in Figure 9. The green dots are the sampling points. We accumulate the gradient at these points, including the direction and magnitude as the transmittance gradient of this ray $(x_i \leftrightarrow x_{i+1})$.

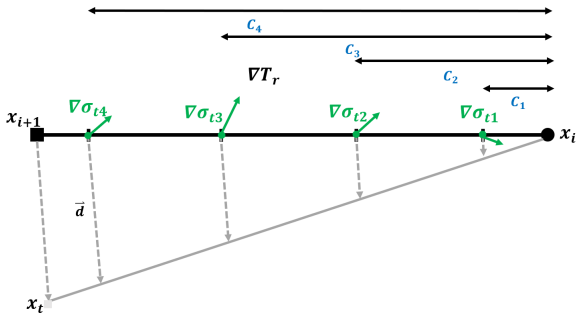


Figure 9: *Gradient of transmittance.*

The equation of gradient of T_r for a vertex, moving in an inhomogeneous medium, from x_{i+1} to any given position x_t along vector $\vec{d}$ is:

$$\begin{aligned}\nabla T_r(x_i \leftrightarrow x_{i+1}) &= -\frac{1}{\bar{\sigma}}\nabla\sigma_t(x_0)(1-\frac{\sigma_t(x_1)}{\bar{\sigma}})\dots(1-\frac{\sigma_t(x_k)}{\bar{\sigma}}) \\ &\quad -\frac{1}{\bar{\sigma}}\nabla\sigma_t(x_1)(1-\frac{\sigma_t(x_0)}{\bar{\sigma}})\dots(1-\frac{\sigma_t(x_k)}{\bar{\sigma}})\dots \\ &= -T_r(x_i \leftrightarrow x_{i+1})\sum_{j=0}^{n}\frac{\nabla\sigma_t(x_j)}{\bar{\sigma}-\sigma_t(x_j)},\end{aligned} \tag{19}$$

We notice that the moving distance of each sampling point is proportional per the theory of similar triangles. We can define it as $\frac{c_i}{c_n}\vec{d}$, shown in Figure 9. Considering this factor, the gradient of transmittance is:

$$\nabla T_r(x_i \leftrightarrow x_{i+1}) = -\frac{T_r(x_i \leftrightarrow x_{i+1})}{c_n}\sum_{j=0}^{n}\frac{c_j\nabla\sigma_t(x_j)}{\bar{\sigma}-\sigma_t(x_j)}, \tag{20}$$

Furthermore, we only consider the influence from the change of sampling points in inhomogeneous media, the change of distance also influences the transmittance. Here, we choose an average density σ_{avg}. Then, we can apply Eq. 18 to measure the gradient by the change of length. We get the general equation of the gradient of transmittance:

$$\nabla\left(T_r\left(x_i \leftrightarrow x_{i+1}\right)\right) = -T_r\left(x_i \leftrightarrow x_{i+1}\right)\left(\sum_{j=0}^{n}\frac{\frac{c_j}{c_n}\nabla\sigma_t\left(x_j\right)}{\bar{\sigma}-\sigma_t\left(x_j\right)} + \sigma_{avg}\frac{\overrightarrow{x_i x_{i+1}}}{\|\overrightarrow{x_i x_{i+1}}\|}\right), \tag{21}$$

where:

$\sigma_{avg} = \frac{1}{n}\sum_{j=0}^{n}\sigma_t\left(x_j\right)$.

We can also apply the Ray Marching method to compute transmittance. For this method, the gradient of transmittance is given by Jarosz et al. [JDZJ08]. Ratio Tracking is preferred for highly heterogenous media [NSJ14]. For both methods we can now determine the gradient.

4.4 Phase function

Isotropic Media. The phase function in isotropic Media is constant for any angular distribution. Therefore, the gradient of the phase function is 0:

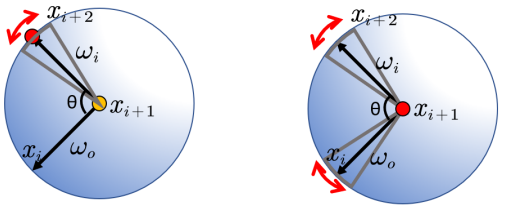
$$\nabla f_p(x_i \leftrightarrow x_{i+1} \leftrightarrow x_{i+2}) = 0, \tag{22}$$

Anisotropic Media. In anisotropic media, the gradient of Eq. 9 is [JDZJ08]:

$$\nabla f_p(x_i \leftrightarrow x_{i+1} \leftrightarrow x_{i+2}) = \frac{1}{4\pi} \frac{3g(1-g^2)}{(1+g^2-2g(\cos\theta))^{5/2}} \nabla \cos\theta, \tag{23}$$

where:

θ is $\angle x_i x_{i+1} x_{i+2}$.



(*a*) one side is changed (*b*) both sides are changed

Figure 10: *Gradient of the phase function in two cases.*

We need to compute the gradient of $\cos\theta$. There are two cases, shown in Figure 10, moving vertex x_{i+2} or moving vertex x_{i+1}. We denote $\omega_o = \overrightarrow{x_{i+1}x_i}$, $\hat{\omega}_o = \frac{\overrightarrow{x_{i+1}x_i}}{\|\overrightarrow{x_i x_{i+1}}\|}$, $\omega_i = \overrightarrow{x_{i+1}x_{i+2}}$, $\hat{\omega}_i = \frac{\overrightarrow{x_{i+1}x_{i+2}}}{\|\overrightarrow{x_{i+1}x_{i+2}}\|}$.

In the first case in which we move vertex x_{i+2}, ω_o is constant, the gradient of $\cos\theta$ is:

$$\nabla \cos\theta(\omega_i) = \frac{\hat{\omega}_o}{|\omega_i|} - \cos\theta \frac{\omega_i}{|\omega_i|^2} \tag{24}$$

In the second case where we move vertex x_{i+1}, both ω_o and ω_i are changed. The gradient of $\cos\theta$ is:

$$\nabla \cos\theta(\omega_i, \omega_o) = \frac{\hat{\omega}_o}{|\omega_i|} - \cos\theta \frac{\omega_i}{|\omega_i|^2} + \frac{\hat{\omega}_i}{|\omega_o|} - \cos\theta \frac{\omega_o}{|\omega_o|^2} \tag{25}$$

4.5 Geometry factor

We derive the gradient of the geometry factor from Eq. 10:

$$\nabla G(x_i \leftrightarrow x_{i+1}) = -\frac{2}{A^2} \overrightarrow{x_i x_{i+1}} \tag{26}$$

We have determined the gradient of the Transmittance Eq. 21, the phase function Eq. 23, and the geometry factor Eq. 26. The final step is to incorporate these terms as a whole, and compute the gradient of the light path.

4.6 Gradient of the Volume Rendering Equation

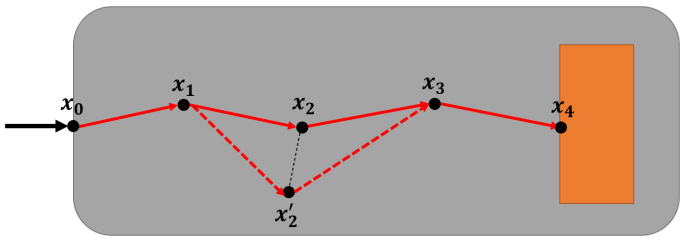


Figure 11: *Gradient of the Volume Rendering Equation. We compute the gradient of the path* $(x_0 \leftrightarrow x_1 \leftrightarrow x_2 \leftrightarrow x_3 \leftrightarrow x_4)$ *with respect to vertex* x_2*. Then, we move vertex* x_2 *to* x_2' *and generate a new path (dash red line). We use the gradient to estimate the radiance of this new path* $(x_0 \leftrightarrow x_1 \leftrightarrow x_2' \leftrightarrow x_3 \leftrightarrow x_4)$.

In the previous section, we provided a full set of gradient functions, one for each term in the VRE. We can use them to determine the gradient of the VRE as a whole. Based on Eq. 5, we describe the light path shown in Figure 11 as:

$$L(\bar{x}) = W_e f(x_0 \leftrightarrow x_1) f(x_1 \leftrightarrow x_2) f(x_2 \leftrightarrow x_3) f(x_3 \leftrightarrow x_4) L_e \tag{27}$$

When we move vertex x_2 to x_2', the change affects two path segments, $(x_1 \rightarrow x_2)$ and $(x_2 \rightarrow x_3)$. We measure the gradient of the light path by combining the individual gradients:

$$\begin{aligned} \nabla L(\bar{x}) &= L_e W_e \big((\nabla f(x_1 \leftrightarrow x_2)) f(x_0 \leftrightarrow x_1) f(x_2 \leftrightarrow x_3) f(x_3 \leftrightarrow x_4) \\ &\quad + (\nabla f(x_2 \leftrightarrow x_3)) f(x_0 \leftrightarrow x_1) f(x_1 \leftrightarrow x_2) f(x_3 \leftrightarrow x_4) \big) \\ &= L(\bar{x}) \left(\frac{\nabla f(x_1 \leftrightarrow x_2)}{f(x_1 \leftrightarrow x_2)} + \frac{\nabla f(x_2 \leftrightarrow x_3)}{f(x_2 \leftrightarrow x_3)} \right) \end{aligned} \tag{28}$$

The gradient of ray $(x_1 \rightarrow x_2)$ includes the gradients of the transmittance $T_r(x_1 \leftrightarrow x_2)$, the phase function $f_p(x_0 \leftrightarrow x_1 \leftrightarrow x_2)$, the geometry factor $G(x_1 \leftrightarrow x_2)$. The gradient of ray $(x_2 \rightarrow x_3)$ includes the gradients of the transmittance $T_r(x_2 \leftrightarrow x_3)$, the phase function $f_p(x_2 \leftrightarrow x_3 \leftrightarrow x_4)$, and the geometry factor $G(x_2 \leftrightarrow x_3)$. With an additional shared phase function $f_p(x_1 \leftrightarrow x_2 \leftrightarrow x_3)$, we can expand the gradient of the two rays to the related terms in the VRE:

$$\begin{aligned} \nabla L(\bar{x}) = L(\bar{x}) \Bigg(& \frac{\nabla T_r(x_1 \leftrightarrow x_2)}{T_r(x_1 \leftrightarrow x_2)} + \frac{\nabla T_r(x_2 \leftrightarrow x_3)}{T_r(x_2 \leftrightarrow x_3)} \\ & + \frac{\nabla f_p(x_0 \leftrightarrow x_1 \leftrightarrow x_2)}{f_p(x_0 \leftrightarrow x_1 \leftrightarrow x_2)} + \frac{\nabla f_p(x_1 \leftrightarrow x_2 \leftrightarrow x_3)}{f_p(x_1 \leftrightarrow x_2 \leftrightarrow x_3)} + \frac{\nabla f_p(x_2 \leftrightarrow x_3 \leftrightarrow x_4)}{f_p(x_2 \leftrightarrow x_3 \leftrightarrow x_4)} \\ & + \frac{\nabla G(x_1 \leftrightarrow x_2)}{G(x_1 \leftrightarrow x_2)} + \frac{\nabla G(x_2 \leftrightarrow x_3)}{G(x_2 \leftrightarrow x_3)} \Bigg) \end{aligned} \tag{29}$$

Once we obtain the gradient of path $\bar{x}$ with respect to vertex x_2, we estimate the radiance of the new path $\bar{x}' : (x_0 \leftrightarrow x_1 \leftrightarrow x_2' \leftrightarrow x_3 \leftrightarrow x_4)$:

$$L(\bar{x}') = L(\bar{x}) + \left\langle \nabla L(\bar{x}), \overrightarrow{x_2 x_2'} \right\rangle \tag{30}$$

Eq. 29 tells us we only need to focus on the terms which are changed. We thus do not need to store the information of unrelated terms in the equation. This allows us to apply the gradient algorithm in the standard path tracing algorithm with next event estimator (NEE).

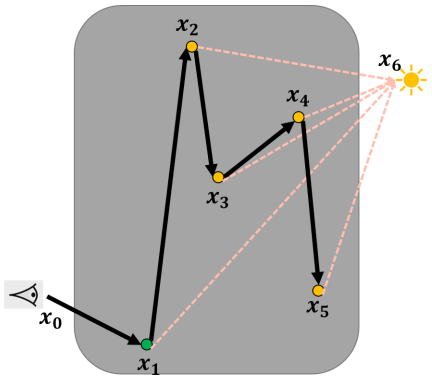


Figure 12: *Path Tracing with NEE.*

This is shown in Figure 12. If we build a light path along the black rays with N vertices, not including the camera and light source, for each vertex, we will emit a direct ray (pink) to the light source. There are thus N or $N+1$ (if we include the implicit path) light paths of different length. If we move one vertex x_1, there are N or $N+1$ corresponding new paths generated, but there are only three related rays we need to compute the gradients of: $(x_0 \leftrightarrow x_1)$, $(x_1 \leftrightarrow x_2)$, and $(x_1 \leftrightarrow x_6)$. There are seven terms we need to compute the gradients of in a light path based on Eq. 29. In a standard path tracing with NEE, there are only three extra terms we need to compute the gradients for N or $N+1$ paths: the transmittance $T_r(x_1 \leftrightarrow x_6)$, the geometry factor $G(x_1 \leftrightarrow x_6)$, and the phase function $f_p(x_1 \leftrightarrow x_2 \leftrightarrow x_6)$.

5 Rendering

In the previous section we have derived the gradient of the VRE. To make our contribution practical, we apply it in a volume rendering algorithm. In this thesis, we apply the gradient algorithm in the MMLT (see Section 2). The inner loop of the MMLT is a bidirectional path tracer (BDPT) ([Vea97] and [LW93]). We thus start by adapting our algorithm to the BDPT. We then design a path space mutation strategy to apply this algorithm, with a radius-reduction scheme.

5.1 BDPT

In this part, we first apply our gradient algorithm to BDPT. Then, we extend it to support other types including camera, light, and surface, which make this algorithm more general.

5.1.1 BDPT Connection

The gradient algorithm only focuses on the radiance of path $f(\bar{x}_i)$. For BDPT, we also need to get the values of $p(\bar{x}_i)$ and $\omega(\bar{x}_i)$ (Eq. 11), i.e., we need to consider the PDF of sampling in the medium, including distance and direction.

Distance. When a ray in a medium is from x_i to x_{i+1}, we compute its transmittance as $T_r(x_i \leftrightarrow x_{i+1})$, the PDF for sampling at x_{i+1} is $p_{dis}(x_{i+1}|x_i)$. We need to compute the value of $\frac{T_r(x_i \leftrightarrow x_{i+1})}{p_{dis}(x_{i+1}|x_i)}$, which is denoted as the sampling weight β_{dis}

For homogeneous media, the equation of transmittance is Eq. 6. Based on this equation, we derive the sampling method for this exponential distribution defined over the domain $(0, \infty)$, and a PDF $p_{dis}(x_{i+1}|x_i)$ for the distance is obtained by normalizing this integral of the exponential function:

$$p_{dis}(x_{i+1}|x_i) = \sigma_t e^{-\sigma_t \|x_{i+1} - x_i\|}, \tag{31}$$

So, in homogeneous media, the value of β_{dis} in the medium interaction case is:

$$\beta_{dis} = \frac{T_r(x_i \leftrightarrow x_{i+1})}{p_{dis}(x_{i+1}|x_i)} = \frac{1}{\sigma_t} \tag{32}$$

In inhomogeneous media, the medium interaction case resembles this process and has the same result [PJH16].

Direction. It is usually assumed that phase functions are sampled with PDFs that perfectly match their distributions. This means the values of $f_p(x_i \leftrightarrow x_{i+1} \leftrightarrow x_{i+2}))$ and its PDF p_ω are the same:

$$\beta_\omega = \frac{f_p(x_i \leftrightarrow x_{i+1} \leftrightarrow x_{i+2})}{p_\omega(x_i \leftrightarrow x_{i+1} \leftrightarrow x_{i+2})} = 1 \tag{33}$$

Shown in Figure 11, when we move the vertex x_2 to x_2', the transmittance is changed from $T_r(x_1 \leftrightarrow x_2)$ to $T_r(x_1 \leftrightarrow x_2')$, and the PDF is changed from $p_{dis}(x_2|x_1)$ to $p_{dis}(x_2'|x_1)$. The sampling weight $\beta_{dis}(x_1 \leftrightarrow x_2)$ cancels this change, $\beta_{dis}(x_1 \leftrightarrow x_2') = \beta_{dis}(x_1 \leftrightarrow x_2)$. This is the same in the phase function. In BDPT, the connection ray between the eye path and light path does not need sampling, so, we only need to re-calculate the vertices related to the connection ray and ignore others. There are thus six distinctive situations, which are illustrated in Figure 13.

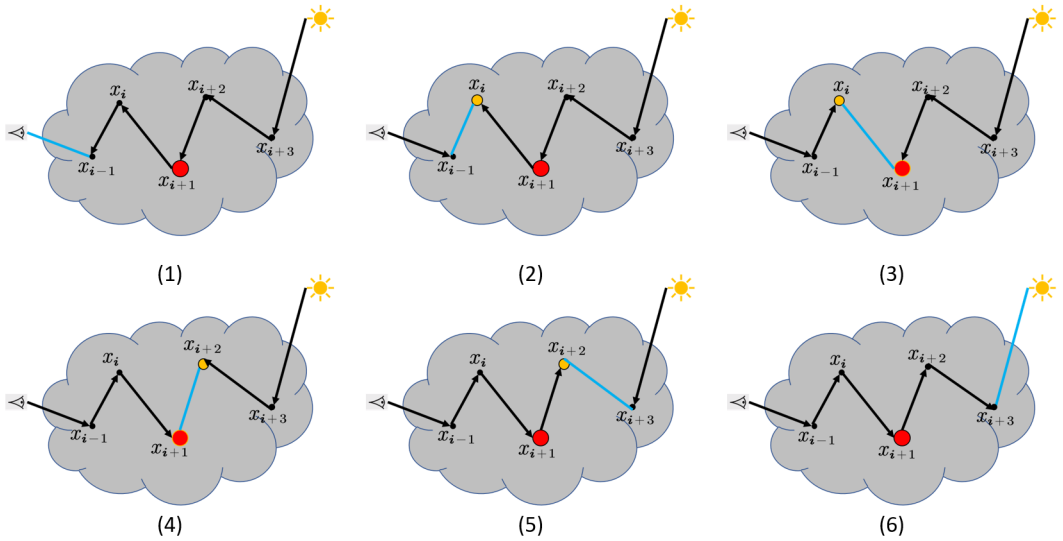


Figure 13: *six distinctive situations in BDPT.*

In Figure 13, we move vertex x_{i+1} to generate a new light path. Vertex x_i and x_{i+2} are its neighbor vertices. We only need to update these orange vertices. So, in cases 1 and 6, no matter how we move vertex x_{i+1} and generate a new path $\bar{x}'$, $\frac{f(\bar{x}')}{p(\bar{x}')}$ is the same as $\frac{f(\bar{x})}{p(\bar{x})}$. In other cases, the orange vertices are changed. The terms we need to update are:

1. *Case 1*: nothing need to update, unless the mutated path intersects geometry or transmittance is 0
2. *Case 2*: $f_p(x_{i-1} \leftrightarrow x_i \leftrightarrow x_{i+1})$
3. *Case 3*: $T_r(x_i \leftrightarrow x_{i+1})$, $f_p(x_{i-1} \leftrightarrow x_i \leftrightarrow x_{i+1})$, $f_p(x_i \leftrightarrow x_{i+1} \leftrightarrow x_{i+2})$, and $G(x_i \leftrightarrow x_{i+1})$
4. *Case 4*: $T_r(x_{i+1} \leftrightarrow x_{i+2})$, $f_p(x_i \leftrightarrow x_{i+1} \leftrightarrow x_{i+2})$, $f_p(x_{i+1} \leftrightarrow x_{i+2} \leftrightarrow x_{i+3})$, and $G(x_{i+1} \leftrightarrow x_{i+2})$

5. *Case 5*: $f_p(x_{i+1} \leftrightarrow x_{i+2} \leftrightarrow x_{i+3})$
6. *Case 6*: nothing need to update, unless the mutated path intersects geometry or transmittance is 0

So, in each case, we only need to compute the gradient of these mentioned terms, update the MIS weight of the new path. Finally, we can get the value $\omega(\bar{x}_i')\frac{f(\bar{x}_i')}{p(\bar{x}_i')}$ of the new path.

5.1.2 Neighbor Type

There is a strict requirement in our gradient algorithm that vertex x_{i+1} and its two neighbor vertices x_i and x_{i+2} must be in the medium. Including light source and camera, this path has at least 5 vertices. The greater the length, the less the throughput of the path. This limits the generalization of the gradient algorithm. So, we only require the vertex x_{i+1} in the medium, and vertices x_i and x_{i+2} can be *Camera*, *Light Medium*, or *Surface*

Camera

There are two scenarios where neighbor vertex x_i is the camera, shown in Figure 14. The first is case 3, we create an explicit connection from x_{i+1} to the camera. The second is case 4, the camera generates a ray from x_i to x_{i+1}. When we move the vertex at x_{i+1}, both the value of $W_e(x_i \leftrightarrow x_{i+1})$ and its PDF $p(x_i \leftrightarrow x_{i+1})$ are changed. We compute the gradient of $W_e(x_i \leftrightarrow x_{i+1})$, and use the gradient to estimate the new value of W_e due to the offset of vertex at x_{i+1}. The final value we need is $\frac{W_e(x_i \leftrightarrow x_{i+1})}{p(x_i \leftrightarrow x_{i+1})}$, which is denoted as β_{camera}.

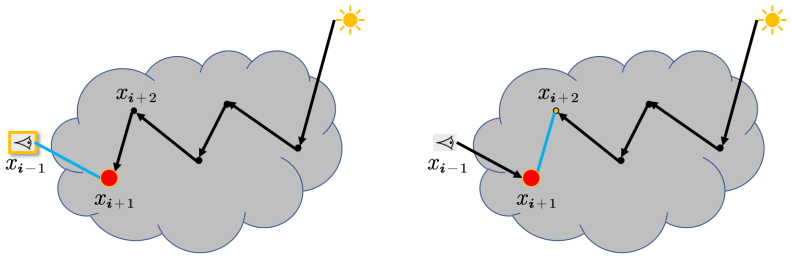


Figure 14: *Gradient of camera importance.*

Case 3. We need to measure the importance of the ray from any possible direction. The camera uniformly generates samples over the image plane area A, so the area-measure PDF for any point at x_t on the image plane is $p(x_t) = \frac{1}{A}$. The relationship between differential solid angle and differential area is [PJH16]:

$$d\omega = \frac{dA\cos\theta}{r^2} \tag{34}$$

The directional PDF:

$$p(\omega) = \frac{1}{A\cos\theta^3} \tag{35}$$

We must satisfy the ray-space normalization in ray-space $A_{lens} * S^2$. Here, A_{lens} is the lens area, S^2 is the hemisphere of directions about the normal of camera:

$$\int_{A_{lens}} \int_{s^2} W_e(x_i \leftrightarrow x_{i+1}) |\cos\theta| d\omega dA(p) = 1, \tag{36}$$

where:

θ is the angle between the normal of the lens and $\overrightarrow{x_i x_{i+1}}$.

W_e is the importance emitted from the point x_i on the camera in a direction $\overrightarrow{x_i x_{i+1}}$. W_e is calculated as follows:

$$W_e\left(x_i \leftrightarrow x_{i+1}\right) = \frac{p\left(\omega\right)}{\pi r^2 \cos\theta} \tag{37}$$

r is the radius of lens on the camera, which is constant. And we can get the weight β_{camera}:

$$\beta_{camera} = \frac{W_e\left(x_i \leftrightarrow x_{i+1}\right)}{p\left(\omega\right)} = \frac{1}{\pi r^2 \cos\theta} \tag{38}$$

Eq. 24 is the method to compute the gradient of $\cos\theta$. Furthermore, we can get the gradient of β_{camera}.:

$$\nabla\beta_{camera} = \frac{-1}{\pi r^2 \cos^2\theta}\nabla\cos\theta \tag{39}$$

Case 4. The ray is generated from the camera uniformly, the importance of each ray is the same, which is 1. Since this is constant, its gradient is 0.

Surface

When the vertex is surface type, we need to use f_r the $BRDF$ instead of f_p the phase function. And the materials are either *diffuse* or *specular*. The introduction of surface interactions for neighboring vertices x_i and x_{i+2} is shown in Figure 15.

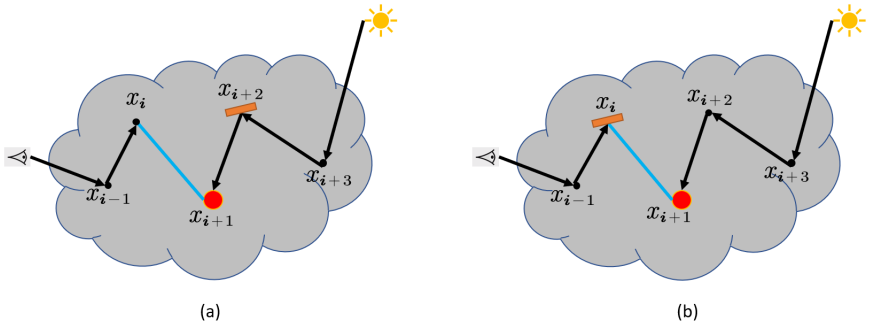


Figure 15: *Gradient of surface.*

Diffuse. Shown in Figure 15, case(a): the vertex at x_{i+2} is surface type and the vertex at x_{i+1} is medium type, the ray $(x_{i+2} \leftrightarrow x_{i+1})$ is an output ray from vertex x_{i+2}. We need to compute the throughput of the related ray $(x_{i+2} \leftrightarrow x_{i+1})$ caused by the offset of vertex x_{i+1} including the radiance and its PDF: which is denoted as $\beta_{surface}$:

$$\beta_{surface} = \frac{f_r(x_{i+3} \leftrightarrow x_{i+2} \leftrightarrow x_{i+1})G(x_{i+2} \leftrightarrow x_{i+1})}{p_{area}(x_{i+1})} = \frac{f_r(x_{i+3} \leftrightarrow x_{i+2} \leftrightarrow x_{i+1})\cos\theta}{p_\omega(x_{i+2})}, \tag{40}$$

where:

θ is the angle between the normal of the surface at vertex x_{i+2} and the ray $(x_{i+2} \leftrightarrow x_{i+1})$

f_r is the BRDF function

$p_{area}(x_{i+1})$ is the probability density with respect to surface area at x_{i+1}

$p_{\omega}(x_{i+2})$ is the probability density with respect to solid angle at x_{i+2}.

We use Lambertian reflectance to represent $BRDFs$ in this thesis for simplification, and the PDF is the cosine function:

$$\beta_{surface} = \frac{\frac{R}{\pi}\cos\theta}{\frac{\cos\theta}{\pi}} = R, \tag{41}$$

where:

R is the color on the surface at x_{i+2}.

So, in this situation, $\beta_{surface}$ is constant, and its gradient is 0.

Shown in Figure 15 case(b) , another situation is the vertex at x_i is surface type. The ray $(x_i \leftrightarrow x_{i+1})$ is a connection ray. So, there is no sampling, and the $\beta_{surface}$ is:

$$\beta_{surface} = f_r(x_{i+1} \leftrightarrow x_i \leftrightarrow x_{i-1})G(x_{i+1} \leftrightarrow x_i) = \frac{R}{\pi}G(x_{i+1} \leftrightarrow x_i) \tag{42}$$

Eq. 26 computes the gradient of the geometry factor in the medium, which can be extended to the surface case simply. And the gradient is here:

$$\nabla\beta_{surface} = \frac{R}{\pi}\nabla G(x_{i+1} \leftrightarrow x_i) \tag{43}$$

Specular. If only one neighbor vertex x_i or x_{i+2} is specular, which is illustrated in Figure 16, we offset the vertex x_{i+1} along the ray $(x_{i+1} \leftrightarrow x_{i+2})$ in positive or negative direction randomly. In this situation, we only need to measure the gradient of the geometry factor $\nabla G(x_{i+1} \leftrightarrow x_{i+2})$.

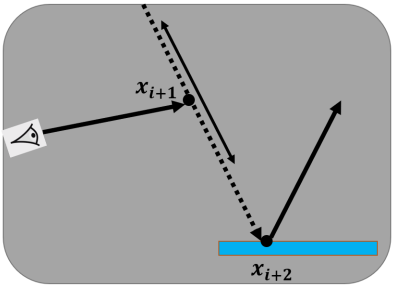


Figure 16: *Gradient of specular surface.*

If both vertex x_i and x_{i+2} are specular, either x_i or x_{i+2} must be shifted when we offset the vertex x_{i+1}. Because the lack of continuity on the surface. we can not apply the gradient algorithm when both vertices x_i and x_{i+2} are specular.

Light

In this thesis, we consider two light types: environment light and area lights.

Environment light. Environment light is defined as an infinite area light which means an infinitely far away area light source that surrounds the entire scene in the *PBR* book [PJH16]. The radiance from any point in the light source can reach anywhere in the scene equally. It seems like you are always in the center of this sky box and the radiance is constant to anywhere in the scene. There is no influence from distance and direction, and the gradient is 0.

Area lights. Shown in Figure 17, the radiance of the area light path is constant, but it is influenced by the distance and direction, which is the same as the diffuse surface type.

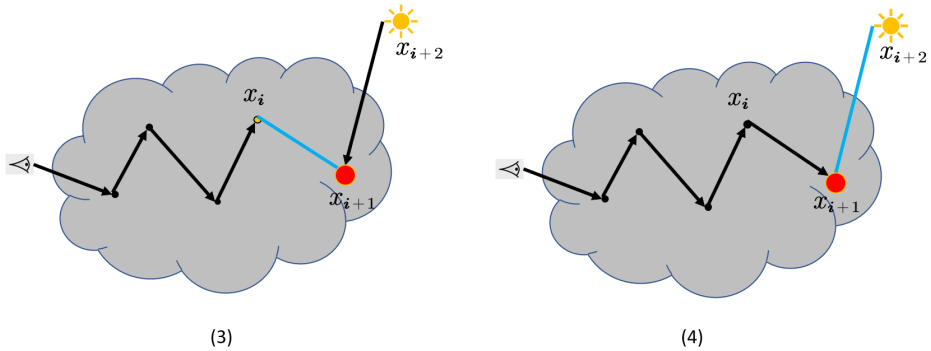


Figure 17: *Gradient of area light.*

5.2 Multiplexed MLT

Now that we have determined how BDPT can be adapted to support the use of gradients, we can design a mutation strategy to apply this algorithm in MMLT.

Shown in Figure 18, (a) is a traditional mutation strategy, the transition is a Markov Chain from $\bar{x}_i$ to $\bar{x}_{i+1}$ using the PSS mutation strategy. In each mutation, we satisfy the detailed balance to maintain equilibrium distribution. We extend it to our gradient mutation strategy (b): in each iteration, $\bar{x}_{i+1}$ is generated from $\bar{x}_i$ using the PSS mutation strategy (red lines), which is the same as a tradition mutation. Our gradient mutation only perturbs one vertex in the path $\bar{x}_i$ and generates a new path (blue line). We do this repeatedly and obtain a set X_i, which contains n correlated paths ($\bar{x}_{(i,0)}$,$\bar{x}_{(i,1)}$,...,$\bar{x}_{(i,n)}$) from $\bar{x}_i$. We do the same to $\bar{x}_{i+1}$, and get another set X_{i+1}, which contains m correlated paths ($\bar{x}_{(i+1,0)}$,$\bar{x}_{(i+1,1)}$,...,$\bar{x}_{(i+1,m)}$) from $\bar{x}_{i+1}$. Then, we select two paths from sets X_i and X_{i+1} respectively. For a pair of paths ($\bar{x}_{(i,j)}$,$\bar{x}_{(i+1,k)}$), we make a transition and measure the acceptance probability value (red dash line), and add the contributions in both locations. If the set X_i has n elements and X_{i+1} has m elements, there are mn pairs to make the contributions instead of one pair in a traditional mutation strategy. Finally, the state of the Markov Chain is still determined by the PSS mutation strategy (red line): the gradient mutation strategy only makes the chain wider and gets more contributions; it does not alter the acceptance or rejection process. In this way, we do not need to reverse the path from path space to PSS.

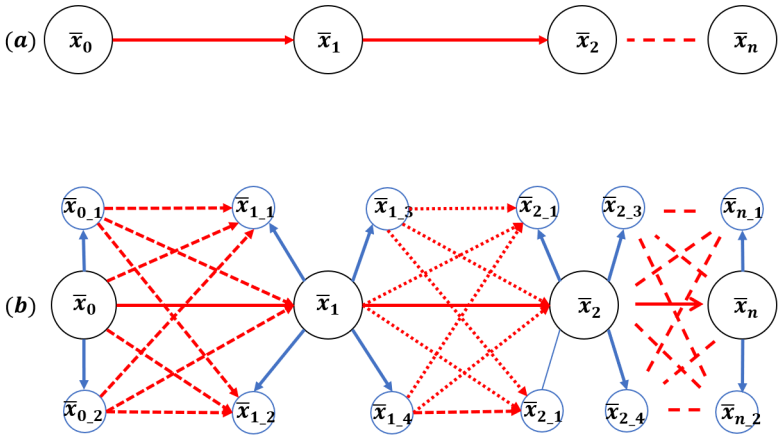


Figure 18: *Traditional mutation vs. Gradient mutation. In each iteration, gradient mutation provides many short chains (red dash line) while keeping the main Markov Chain (red line) unchanged.*

The algorithm that generates new paths (the blue lines in Figure 18(b)), using our gradient mutation strategy, is shown in algorithm 2:

Algorithm 2: mutationByBatch($\bar{x}$)

```
1  X_array = [];
2  idx = getAvailableVertex(x̄);
3  v_gradient = calculatePathGradient(x̄, idx);
4  r = initialRadius;
5  for i = 1 to n do
6      v_dir = Random_SphericalDistribution());
7      v_offset = r * v_dir;
8      x̄_i = estimateNewPath(x̄, v_gradient, v_offset);
9      updateMISWeight(x̄_i);
10     X_array.push(x̄_i);
11     r* = getRadiusRatio();
12 end
13 returnX_array;
```

We generate n new paths from the existing path $\bar{x}$, and estimate the contributions by applying the gradient algorithm. First, we choose one available vertex in path $\bar{x}$, compute the gradient of $\bar{x}$ with respect to this vertex (line 3), and get the value of initial radius (line 4). Then, in the loop, we use a uniform spherical distribution to get a direction (line 6), with the radius and direction, we get the vector for the perturbation offset (line 7), estimate the contribution of the new path $\bar{x}_i$ (line 8), and update its MIS weight (line 9). Finally, we push the new path in array X_array (line 10), and reduce the value of r for the next iteration (line 11).

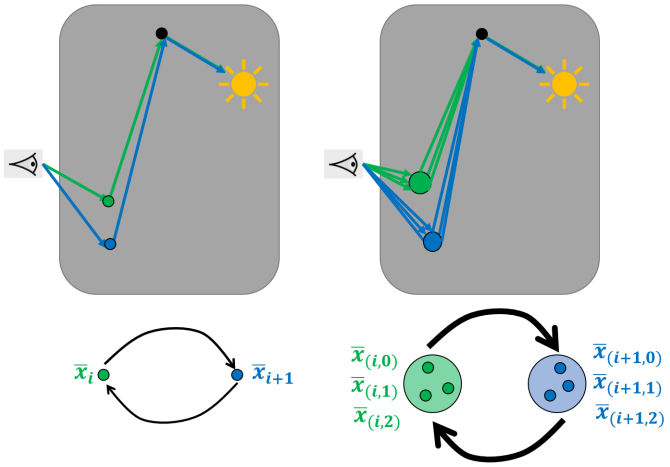


Figure 19: *Gradient mutation strategy.*

Analysis. Imagine laying a pipe from $\bar{x}_i$ to $\bar{x}_{i+1}$ along which radiance can flow, as shown in Figure 19 (left). The proposed path $\bar{x}_{i+1}$ is generated from the existing path $\bar{x}_i$ by PSS mutation strategy. The detailed balance of the transition from $\bar{x}_i$ to $\bar{x}_{i+1}$ is guaranteed, and we denote $\Phi\left(\bar{x}_i \to \bar{x}_{i+1}\right)$ as the radiance flows from $\bar{x}_i$ to $\bar{x}_{i+1}$:

$$\begin{cases} \Phi\left(\bar{x}_i \to \bar{x}_{i+1}\right) = f\left(\bar{x}_i\right) T\left(\bar{x}_{i+1} | \bar{x}_i\right) \alpha\left(\bar{x}_{i+1} | \bar{x}_i\right) \\ \Phi\left(\bar{x}_{i+1} \to \bar{x}_i\right) = f\left(\bar{x}_{i+1}\right) T\left(\bar{x}_i | \bar{x}_{i+1}\right) \alpha\left(\bar{x}_i | \bar{x}_{i+1}\right) \\ \Phi\left(\bar{x}_i \to \bar{x}_{i+1}\right) = \Phi\left(\bar{x}_{i+1} \to \bar{x}_i\right) \end{cases} \tag{44}$$

Our gradient mutation strategy trying to widen this pipe and more radiance can flow through this pipe, shown in Figure 19 (right). First, the transition is symmetric; second, the transition probability in the gradient mutation strategy is a uniform spherical distribution:

$$\begin{cases} T\left(\bar{x}_{i+1} | \bar{x}_i\right) = T\left(\bar{x}_i | \bar{x}_{i+1}\right) \\ T\left(\bar{x}_{(i,j)} | \bar{x}_i\right) = p\left(\bar{x}_i\right) \\ T\left(\bar{x}_{(i+1,k)} | \bar{x}_{i+1}\right) = p\left(\bar{x}_{i+1}\right) \end{cases}, \tag{45}$$

where $p\left(\bar{x}_i\right)$ and $p\left(\bar{x}_{i+1}\right)$ are only determined by two known paths $\bar{x}_i$ or $\bar{x}_{i+1}$ respectively.

Furthermore, our gradient mutation strategy only moves one vertex followed a uniform spherical distribution and estimate its contribution by the gradient algorithm. In this situation, an important property of the expected value is satisfied:

$$\begin{aligned} E(f(\bar{x}_{(i,k)})) &\approx f(\bar{x}_i) \\ E(f(\bar{x}_{(i+1,j)})) &\approx f(\bar{x}_{i+1}) \end{aligned} \tag{46}$$

We measure the radiance flowing from $\bar{x}_{(i,j)}$ to $\bar{x}_{(i+1,k)}$ along this widened pipe:

$$\begin{aligned} &\iint \Phi\left(\bar{x}_{(i,j)} \to \bar{x}_{(i+1,k)}\right) d\mu\left(\bar{x}_{(i,j)}\right) d\mu\left(\bar{x}_{(i+1,k)}\right) \\ &= E\left[\Phi\left(\bar{x}_{(i,j)} \to \bar{x}_{(i+1,k)}\right)\right] \\ &= E\left[f\left(\bar{x}_{(i,j)}\right) T\left(\bar{x}_{(i+1,k)} | \bar{x}_{i+1}\right) T\left(\bar{x}_{i+1} | \bar{x}_i\right) T\left(\bar{x}_i | \bar{x}_{(i,j)}\right) \alpha\left(\bar{x}_{(i+1,k)} | \bar{x}_{(i,j)}\right)\right] \\ &= p\left(\bar{x}_i\right) p\left(\bar{x}_{i+1}\right) E\left[f\left(\bar{x}_{(i,j)}\right) T\left(\bar{x}_{i+1} | \bar{x}_i\right) \alpha\left(\bar{x}_{(i+1,k)} | \bar{x}_{(i,j)}\right)\right] \end{aligned} \tag{47}$$

The radiance flowing from $\bar{x}_{(i+1,k)}$ to $\bar{x}_{(i,j)}$ can be measured in the same way by Eq. 47. Because the gradient mutation strategy applies the gradient algorithm, based on Eq. 44, Eq. 45, and Eq. 46, the expected flow is equal, and the detailed balance of the entire pipe is guaranteed:

$$\begin{gathered} p\left(\bar{x}_i\right) p\left(\bar{x}_{i+1}\right) f\left(\bar{x}_i\right) T\left(\bar{x}_{i+1} | \bar{x}_i\right) \alpha\left(\bar{x}_{i+1} | \bar{x}_i\right) = p\left(\bar{x}_i\right) p\left(\bar{x}_{i+1}\right) f\left(\bar{x}_{i+1}\right) T\left(\bar{x}_i | \bar{x}_{i+1}\right) \alpha\left(\bar{x}_i | \bar{x}_{i+1}\right) \\ \Downarrow \\ E\left[\Phi\left(\bar{x}_{(i,j)} \rightarrow \bar{x}_{(i+1,k)}\right)\right] = E\left[\Phi\left(\bar{x}_{(i+1,k)} \rightarrow \bar{x}_{(i,j)}\right)\right] \end{gathered} \tag{48}$$

Meanwhile, Eq. 47 shows, we can make $\alpha\left(\bar{x}_{(i+1,j)} \left| \bar{x}_{(i,k)} \right.\right)$ equal to $\alpha\left(\bar{x}_{i+1} \left| \bar{x}_i \right.\right)$, and the detailed balance of the entire pipe is still guaranteed in the statistical sense. Considering there are many contributions from multiple transitions in our method, Eq. 13 becomes:

$$\begin{aligned} h\left(\bar{x}_{i+1}\right) &= \frac{1}{mn} \sum_{j=0}^{m} \left(\sum_{k=0}^{n} \left(f\left(\bar{x}_{(i+1,j)}\right) \alpha\left(\bar{x}_{(i+1,j)} \left| \bar{x}_{(i,k)} \right.\right) + \left(f\left(\bar{x}_{(i,k)}\right) \left(1 - \ \alpha\left(\bar{x}_{(i+1,j)} \left| \bar{x}_{(i,k)} \right.\right)\right)\right)\right)\right) \\ &\approx \alpha\left(\bar{x}_{i+1} \left| \bar{x}_i \right.\right) \sum_{j=0}^{m} \omega(\bar{x}_{(i+1,j)}) f\left(\bar{x}_{(i+1,j)}\right) + \left(1 - \ \alpha\left(\bar{x}_{i+1} \left| \bar{x}_i \right.\right)\right) \sum_{k=0}^{n} \omega(\bar{x}_{(i,k)}) f\left(\bar{x}_{(i,k)}\right), \end{aligned} \tag{49}$$

where $\omega(\bar{x}_{(i+1,j)})$ is the weight of the path $\bar{x}_{(i+1,j)}$.

Now, we apply our gradient mutation strategy, which is an extension of Alg.1:

Algorithm 3: Gradient based Metropolis-Hastings()

```
if available(x̄) then
    array[] X_0 = mutationByBatch(x̄);
end
for i = 1 to n do
    x̄' = mutate(x̄);
    if available(x̄') then
        array[] X_1 = mutationByBatch(x̄');
    end
    α = accept(x̄,x̄');
    record(X_0,1-α);
    record(X_1,α);
    if random() < α then
        x̄ = x̄';
    end
end
```

In Alg.3, function *available()* tells whether the path $\bar{x}$ is available to apply our gradient algorithm. If it is available, we execute function *mutationByBatch* to generate new paths, which builds a set X_0 with length n. The path $\bar{x}'$ is mutated by PSS mutation strategy. We do the same to this path, and get another set X_1 with length m. Then, based on Eq. 49, function *accept* is used to determine the acceptance probability value between $\bar{x}$ and $\bar{x}'$. We finally add the contributions from all locations in these two sets X_0 and X_1.

Advantages. As a path space strategy, we can design some strategies to adapt different situations such as the *lens perturbation*; as a combination of PSS and path space strategies, the correlation between paths is weakened, which reduces variance. The next state of the chain is still determined by the transition from the PSS mutation strategy. The strategies are independent from each other; we gain the benefits from both strategies and do not need to reverse the path from one space to another space. By applying our gradient

algorithm, which is computationally simple and efficient, the cost of a well-designed implementation is nearly negligible.

5.3 Radius reduction scheme

The expected value $h(\bar{x}_{i+1})$ depends on the sum of two similar parts $\sum_{j=0}^{m} \omega(\bar{x}_{(i+1,j)}) f(\bar{x}_{(i+1,j)})$ and $\sum_{k=0}^{n} \omega(\bar{x}_{(i,k)}) f(\bar{x}_{(i,k)})$, Eq. 49. Each part is quite similar to the density estimator in photon mapping.

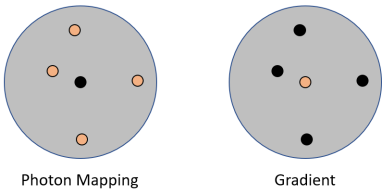


Figure 20: *Density estimator vs. gradient estimator.*

Figure 20 shows the difference between the photon mapping density estimator and our gradient estimator. In photon mapping, the radiance of the orange dots is accurate, and we use them to estimate the black dot in the center of the sphere. In gradient estimation, the radiance of the orange dot in the center is accurate and we use one orange dot and its gradient to estimate the black dots. Similarly, the nearer the dot is, the more accurate the estimator is. So, in our algorithm, we can also build a relationship between distance r to offset and the number n of paths to generate, progressively reduce the radius r and accumulate the contribution to the result image.

Knaus et al. [KZ11] provided an analysis that the variance and bias of density estimator for surface (2-dimension) are $O(\frac{1}{r^2})$ and bias $O(r^2)$ respectively. Based on this analysis, Georgiev et al. [GKDS12] provided a simple radius reduction scheme: $r_i = O(i^{\frac{\alpha-1}{2}})$ where $\alpha \in (0, 1)$, which is a user parameter. the larger α is, the slower the ratio is to reduce the radius, which will lead to higher bias, but lower variance.

In our gradient strategy, we use the same radius reduction scheme: $r_i = O(i^{\frac{\alpha-1}{3}})$ for the media (3-dimension). Meanwhile, we can customize the number of light paths to generate in each iteration. So, we can achieve consistency theoretically by increasing this number while reducing the radius. There are three user parameters to set: the initial radius r; the alpha α; and the number n of samples to generate.

6 Result and discussion

An empirical evaluation of our method is presented in this section. We provide four scenes: smoke, smoke-filled indoor scene, cloud and volumetric caustic scene. We compare our proposed algorithm to MMLT without gradients to evaluate the effectiveness of the gradient scheme. All comparisons are equal-spp rendered with an Intel Core i7-8550U at 1.8GHz using 8 cores. Our code is based on PBRT [pbrt-v3].

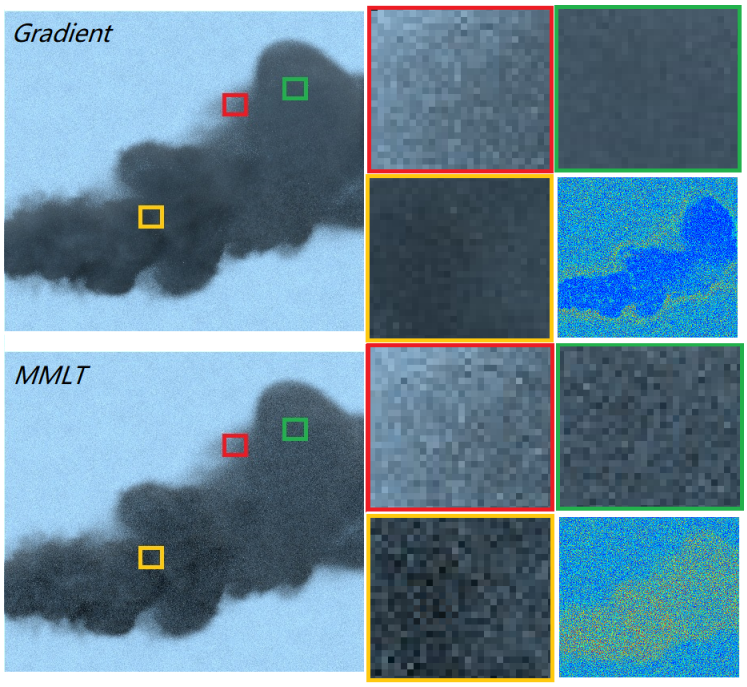


Figure 21: *A comparison of our algorithm against MMLT after* 128*spp. The red square is the edge of the medium. In the green part, the medium density does not change much while in the orange part, the density fluctuates wildly. The medium density image shows the difference with the reference image. Blue color means the difference is small while a red color means the difference is significant.*

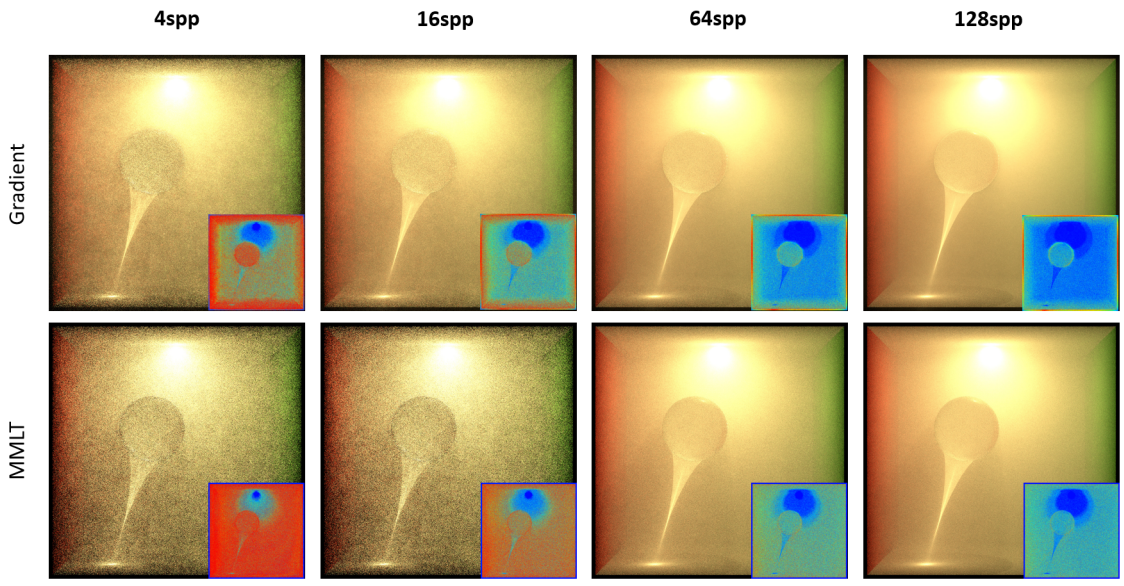


Figure 22: *We compare our algorithm to MMLT on this volumetric caustic scene. The results after iterations are shown for each algorithm.*

The algorithms are compared, using the smoke scene in Figure 21. We highlight three local details and the difference with the reference image. The smoke medium is inhomogeneous and anisotropic with $g = 0.7$. The background is the sky, which provides the only environment light source. Meanwhile, the smoke

medium is darker than the background. As MLT naturally avoids areas where radiance is low, few samples will be accumulated in the medium part. The result shows our gradient strategy performs better than MMLT while keeping the same quality outside the medium.

Figure 22 provides the comparisons in the volumetric caustic scene. The medium is homogeneous, there are walls and a glass sphere, the materials are diffuse or specular respectively. An area light source is on the ceiling of the scene. We provide the results of two algorithms in different sample counts. We can visually notice how the variance is reduced while samples are added. Especially at the early stage, it is clear that the gradient algorithm reduces variance by radiance redistribution among correlated paths.

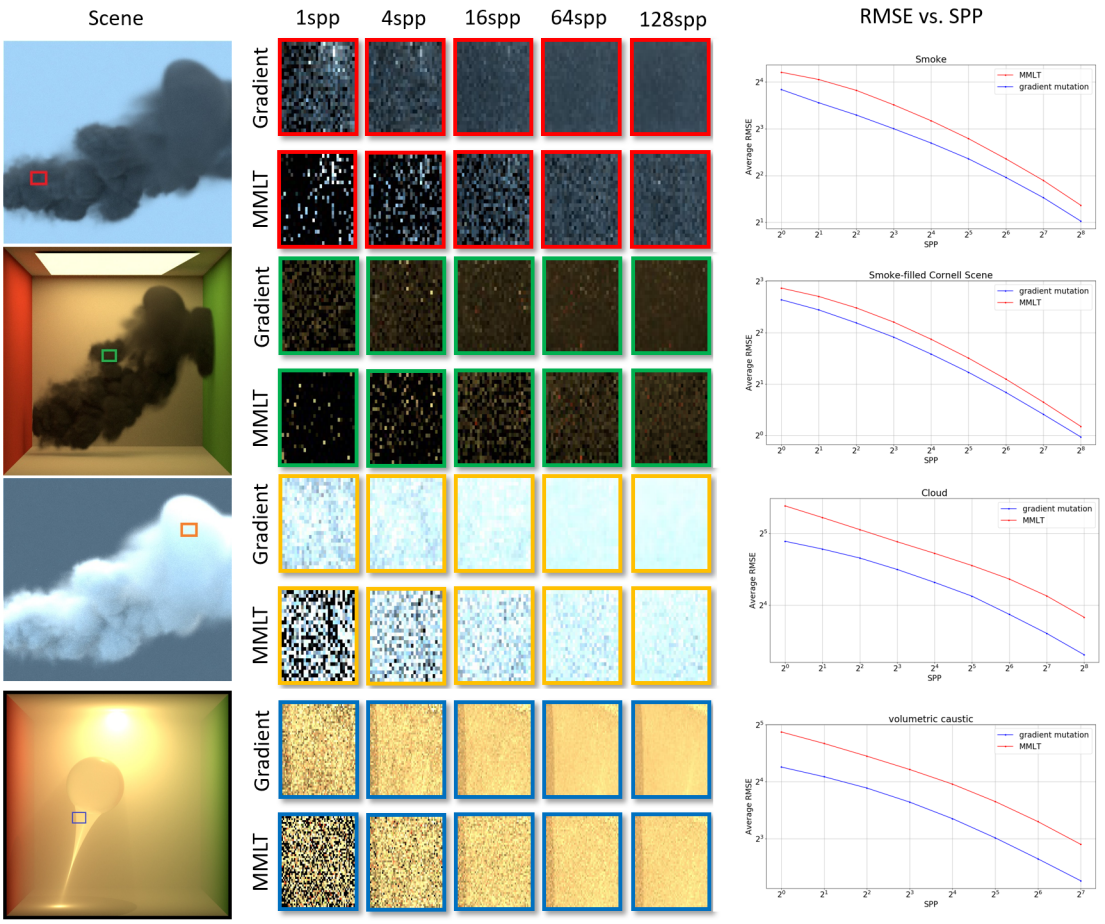


Figure 23: *Four scenes rendered using MMLT and our algorithm. The convergence plot on the right shows the average RMSE as a function of spp. We randomly choose four partial details to compare the quality of convergence.*

Figure 23 provides the comparisons of different samples per pixel on all four scenes with the convergence plot. The cloud is inhomogeneous and istropic. Our algorithm improves the convergence in all four scenarios. Note that the improvement in $RMSE$ is not uniform in the different scenes. This is due to the different media. In the cloud scene, the medium is brighter, leading to more sample distributions in the medium part where there are more opportunities to apply the gradient mutation strategy, which improves the convergence. The smoke scene is darker, and the smoke-filled indoor scene is darkest among these four scenes. Here, there are fewer opportunities to apply the gradient mutation strategy, which reduces the effectiveness of the gradient mutations. Also of interest is the convergence ratio. The expected variance reduction is proportional to $O(\frac{1}{\sqrt{N}})$, but as we increase N, bias increases due to the estimator, which is why we need the radius

reduction scheme to reach the balance. However, when the radius r is smaller, the reduced space ($O(r^3)$) makes it harder to generate new paths. It is thus important to set a suitable initial value and ratio to reduce the radius progressively.

Initial radius and reduction ratio. Figure 24 shows the influence of the initial radius on the final image. On one side, the smaller the initial radius is, the higher the variance is; on the other side, the larger the initial radius is, the higher the bias is. The result shows the MSE is more sensitive to bias than variance ($MSE = Var + Bias^2$). So, a smaller initial radius is safer to guarantee a good quality of the result.

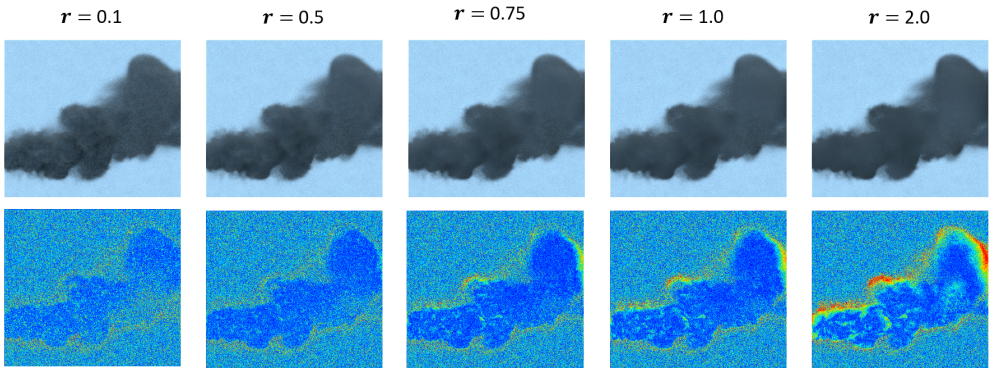


Figure 24: *We compare the results after* 128 *spp with different initial radii. From left to right, we increase the radius value iteratively. When radius*$= 0.1$*, variance is higher; when radius*$= 0.5$*, it improves; when radius*$= 0.75$*, compared to the result when radius*$= 0.5$*, in most part of the medium, it is better, but in the edge or the region where the density changes sharply, it is worse due to over blur; when radius*$= 1.0$ *or radius*$= 2.0$*, it is worst due to over blur in many regions.*

In Figure 25, the left graph shows the influence of α, which controls the reduction ratios of the radii.. A larger value of α yields a reasonable quality at an early stage while a smaller value can be good at later stages. The right graph shows the RMSE outside the medium in four scenes. The result shows our algorithm does not affect the quality outside the medium.

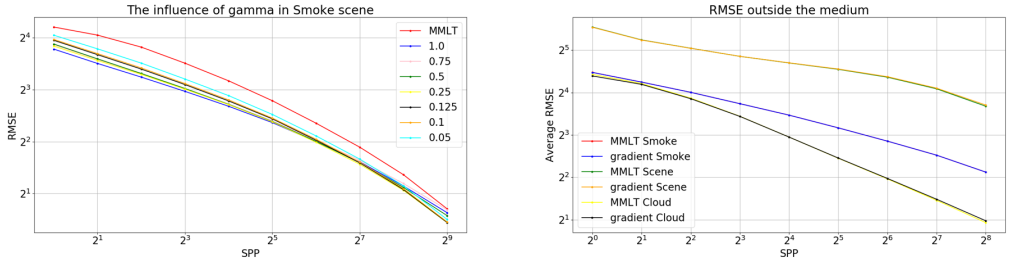


Figure 25: *Left: We compare the influence of gamma α to the result. When $\alpha = 1$, it means the radius is constant, and the smaller α is, the larger the reduction ratio is. In this plot, $\alpha = 0.25$ is a better choice, which provides a good result to reach the balance between variance and bias. Right: we compare the RMSEs of our gradient algorithm and the MMLT outside the medium in four scenes. In each iteration, the value in each iteration is almost the same.*

Discussion. The results show our gradient mutation outperforms MMLT in these four scenes. First, our MLT algorithm integrates path space mutations with the PSS mutation strategy, and it provides many samples at low cost. Second, our gradient algorithm estimates the contribution of a new correlated sample efficiently.

We apply the gradient mutation strategy and gradient algorithm together to improve the convergence in scenes with participating media.

The gradient algorithm is biased. We proposed a progressive radius-reduction scheme to achieve consistency. Due to the complexity of scenes with participating media, some details are still unsolved. For example, our method does not work effectively at the edge of the medium. When the strategy is the lens perturbation case, we can limit the maximum perturbation.

7 Conclusion and future work

Our work addresses light transport for volume rendering. The key idea is to exploit correlated paths to reduce variance. We show that our gradient algorithm is robust, and can efficiently estimate the result. The gradient strategy can be smoothly integrated with PSS strategy in MMLT. It is not necessary to reverse the path from path space to PSS and vice versa. We believe our algorithm has a utility in a wide range of applications, such as supporting standard path tracing algorithms (NEE and BDPT).

There are a number of practical and interesting questions that remain. For example, we do not introduce an occlusion test. It is also a considerable challenge to compute the gradient on a surface with a complicated BRDF function. Finally, gradient method is not only an efficient way for computation, but also reveals the region with higher radiance. We can explore this property for local path guiding.

8 Acknowledgements

I sincerely appreciate people who provide me help to finish this two-year master degree and this thesis.

First of all, I would like to show my sincere gratitude to my great supervisors, Jacco Bikker and Alexandru C. Telea. I really like Jacco's two courses, teaching me programming and leading me into the field of ray tracing. Thank you so much for your support, guidance, giving me freedom to research on the MLT, and correcting my writing problems. I want to thank Professor Alex for your patient feedback and sharing. You pointed out a lot of details to help me improve, and also made me realize the importance of details. The paper you shared looks attractive, when I have time, I will read it.

Secondly, I would like to express my sincere appreciation to PhD student Jerry and Professor Elmar. Thank you for your suggestions, experience sharing, and correction. I read several papers from your group. It is meaningful to discuss with you, especially about how to find problem (idea) and how to do experiments to check the idea. I have learned a lot of lessons. Although this thesis did not completely solve all the problems, these experiences will guide me to do better next time.

Thirdly, thank all of you for your kindness to guide my thesis during the summer vacation.

Finally, a big thanks to my wife, my mother, my friends and my son for supporting me.

Many years later, when I wipe off the dust and find this thesis in the corner by accident, it is your shadows in my mind. This feeling will be very beautiful.

References

Subrahmanyan Chandrasekhar. *Radiative transfer*. Courier Corporation, 2013.

Eva Cerezo, Frederic Pérez, Xavier Pueyo, Francisco J Seron, and François X Sillion. A survey on participating media rendering techniques. *The Visual Computer*, 21(5):303–328, 2005.

David Cline, Justin Talbot, and Parris Egbert. Energy redistribution path tracing. *ACM Transactions on Graphics (TOG)*, 24(3):1186–1195, 2005.

Julian Fong, Magnus Wrenninge, Christopher Kulla, and Ralf Habel. Production volume rendering: Siggraph 2017 course. In *ACM SIGGRAPH 2017 Courses*, pages 1–79. 2017.

Iliyan Georgiev, Jaroslav Křivánek, Tomas Davidovic, and Philipp Slusallek. Light transport simulation with vertex connection and merging. *ACM Trans. Graph.*, 31(6):192–1, 2012.

Binh-Son Hua, Adrien Gruson, Victor Petitjean, Matthias Zwicker, Derek Nowrouzezahrai, Elmar Eisemann, and Toshiya Hachisuka. A survey on gradient-domain rendering. In *Computer Graphics Forum*, volume 38, pages 455–472. Wiley Online Library, 2019.

Toshiya Hachisuka, Anton S Kaplanyan, and Carsten Dachsbacher. Multiplexed metropolis light transport. *ACM Transactions on Graphics (TOG)*, 33(4):1–10, 2014.

Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. Progressive photon mapping. In *ACM SIGGRAPH Asia 2008 papers*, pages 1–8. 2008.

Wojciech Jarosz, Craig Donner, Matthias Zwicker, and Henrik Wann Jensen. Radiance caching for participating media. *ACM Transactions on Graphics (TOG)*, 27(1):1–11, 2008.

Henrik Wann Jensen. Global illumination using photon maps. In *Eurographics workshop on Rendering techniques*, pages 21–30. Springer, 1996.

James T Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986.

Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. Gradient-domain path tracing. *ACM Transactions on Graphics (TOG)*, 34(4):1–13, 2015.

Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. A simple and robust mutation strategy for the metropolis light transport algorithm. In *Computer Graphics Forum*, volume 21, pages 531–540. Wiley Online Library, 2002.

Malvin H. Kalos and Paula A. Whitlock. *Monte Carlo Methods. Vol. 1: Basics*. Wiley-Interscience, USA, 1986.

Claude Knaus and Matthias Zwicker. Progressive photon mapping: A probabilistic approach. *ACM Transactions on Graphics (TOG)*, 30(3):1–13, 2011.

Jaakko Lehtinen, Tero Karras, Samuli Laine, Miika Aittala, Frédo Durand, and Timo Aila. Gradient-domain metropolis light transport. *ACM Transactions on Graphics (TOG)*, 32(4):1–12, 2013.

Eric P Lafortune and Yves D Willems. Bi-directional path tracing. 1993.

Eric P Lafortune and Yves D Willems. Rendering participating media with bidirectional path tracing. In *Rendering techniques' 96*, pages 91–100. Springer, 1996.

Jan Novák, Andrew Selle, and Wojciech Jarosz. Residual ratio tracking for estimating attenuation in participating media. *ACM Trans. Graph.*, 33(6):179–1, 2014.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.

Mark Pauly, Thomas Kollig, and Alexander Keller. Metropolis light transport for participating media. In *Rendering Techniques 2000*, pages 11–22. Springer, 2000.

Eric Veach. *Robust Monte Carlo methods for light transport simulation*, volume 1610. Stanford University PhD thesis, 1997.

Eric Veach and Leonidas J Guibas. Metropolis light transport. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 65–76, 1997.

Adolf N Witt. Multiple scattering in reflection nebulae. i-a monte carlo approach. *The Astrophysical Journal Supplement Series*, 35:1–6, 1977.