Metropolis Light Transport for Participating Media

Mark Pauly Thomas Kollig Alexander Keller

ETH Zürich
pauly@inf.ethz.ch

University of Kaiserslautern
{kollig, keller}@informatik.uni-kl.de

Abstract. In this paper we show how Metropolis Light Transport can be extended both in the underlying theoretical framework and the algorithmic implementation to incorporate volumetric scattering. We present a generalization of the path integral formulation that handles anisotropic scattering in non-homogeneous media. Based on this framework we introduce a new mutation strategy that is specifically designed for participating media. Our algorithm includes effects such as volume caustics and multiple volume scattering, is not restricted to certain classes of geometry and scattering models and has minimal memory requirements. Furthermore, it is unbiased and robust, in the sense that it produces satisfactory results for a wide range of input scenes and lighting situations within acceptable time bounds.

1 Introduction

Many global illumination algorithms have been developed for solving the light transport problem, yet the majority of these methods focuses on scenes without participating media. Volumetric effects due to clouds, fog, smoke or fire can greatly enhance the realism of a rendered image, however, and in many applications are the decisive factor of the simulation. Visibility analysis for traffic or building design, fire research, flight simulation, and high-quality special effects in animation systems all rely on a realistic depiction of volumetric phenomena [Rus94].

Global illumination algorithms for participating media can be classified according to the directional behaviour (isotropic/anisotropic, single/multiple scattering) and spatial variation (homogeneous/inhomogeneous) of the supported media. Finite element methods for isotropic scattering include zonal methods [RT87] and other extensions to the classical radiosity approach, such as hierarchical radiosity [Sil95, Bha93]. Anisotropic scattering was modeled deterministically using spherical harmonics [KVH84], discrete ordinates [LBC94] and point collocation [BT92]. All these algorithms require some discretization of the volume or the directional space into finite elements and compute the interactions between these elements. Thus excessive amounts of memory are required to effectively capture sharp discontinuities of the illumination (e.g. caustics) or uneven directional distributions (e.g. glossy reflections).

Monte Carlo methods are a promising alternative and have been used extensively in global illumination. In the context of participating media, various extensions to existing Monte Carlo approaches have been proposed. Pattanaik and Mudur [PM93] presented a Monte Carlo light tracing algorithm that generates random walks starting from the light sources. They sample interaction points in the volume according to the transmittance of the medium. Lafortune and Willems [LW96] improved on this approach by creating paths both from the light sources and the eye and combining all valid connections between these paths in a multiple sample estimate [VG95]. This led to a bidirectional path tracing algorithm for non-emitting media. A two-pass algorithm based on photon density estimation was presented by Jensen and Christensen in [JC98]. Although simple and efficient the method suffers from various artifacts (e.g. blurred shadow and caustic borders) and requires substantial amounts of memory for difficult lighting situations. In [VG97] a new global illumination algorithm was proposed, the Metropolis light transport (MLT) algorithm. This first application of the Metropolis sampling technique [MRR$^+$53] to the field of computer graphics resulted in a versatile Monte Carlo method for image synthesis.

In this paper we show how the MLT algorithm can be extended to include volumetric scattering. Section 2 briefly reviews the fundamental equation governing the equilibrium distribution of light in scenes with participating media. In section 3 we extend the path integral framework and show how it can be applied to solve the light transport problem. Section 4 is concerned with different aspects of sampling and presents an improved ray marching algorithm. Rendering with the Metropolis light transport algorithm is described in section 5, where we introduce a new mutation strategy for participating media. We present our results in section 6 and draw the conclusions in the final section.

2 Light Transport for Participating Media

We consider the radiance distribution in a finite volume $\mathcal{V} \subset \mathbb{R}^3$. $\partial\mathcal{V}$ is the boundary of $\mathcal{V}$, i.e. a finite set of surfaces describing the objects of the scene. The space between objects is denoted by $\mathcal{V}^0 := \mathcal{V} \setminus \partial\mathcal{V}$, and can be filled with participating media. According to the theory of radiative transfer [Cha50], the equilibrium distribution of radiance L in $\mathcal{V}^0$ is given by the *global balance equation*

$$\begin{aligned}\omega \cdot \nabla L(x,\omega) \quad = \quad & L_{e,\mathcal{V}^0}(x,\omega) + \sigma_s(x) \int_{\mathcal{S}^2} f_p(\omega,x,\omega')L(x,\omega')\, d\omega' \\ & -\sigma_a(x)L(x,\omega) - \sigma_s(x)L(x,\omega). \qquad (1)\end{aligned}$$

It describes the spatial variation of radiance due to emission, in-scattering, absorption and out-scattering. $L_{e,\mathcal{V}^0}$ is the volume emittance function that defines volumetric light sources such as fire or plasma. $\mathcal{S}^2$ is the unit sphere of all directions, σ_s and σ_a are the scattering and absorption coefficients, respectively, and f_p is the phase function, which describes the scattering characteristics of the medium. If f_p is independent of direction, we have isotropic scattering, which is analogous to perfectly diffuse reflection on surfaces. If σ_a and σ_s are independent of position, we have a homogeneous medium. To obtain a complete description of L in $\mathcal{V}$ we need to specify the boundary conditions for $x \in \partial\mathcal{V}$, given by the *local scattering equation*

$$L(x,\omega) = L_{e,\partial\mathcal{V}}(x,\omega) + \int_{\mathcal{S}^2} f_s(\omega,x,\omega')L(x,\omega') \cos\Theta_x \, d\omega', \qquad (2)$$

where f_s is the bidirectional scattering distribution function (BSDF), Θ_x is the angle between ω' and the surface normal in x, and $L_{e,\partial\mathcal{V}}$ defines the emittance on surfaces. We now derive the Fredholm integral equation of the second kind, describing the light transport in the presence of particpating media. The resulting integral equation will be solved using the Neumann series. Incorporating the boundary conditions (2) into equation (1) [Arv93] yields the integral equation

$$\begin{aligned} L(x,\omega) &= \tau(x_{\partial\mathcal{V}},x)L(x_{\partial\mathcal{V}},\omega) \\ &\quad + \int_{x_{\partial\mathcal{V}}}^{x} \tau(x',x) \left[L_{e,\mathcal{V}^0}(x',\omega) + \sigma_s(x') \int_{\mathcal{S}^2} f_p(\omega,x',\omega')L(x',\omega')d\omega' \right] dx', \end{aligned} \tag{3}$$

where $x_{\partial\mathcal{V}} := h(x,-\omega) \in \partial\mathcal{V}$ is the closest surface point from x in direction $-\omega$ determined by the ray casting function h. Equation (3) expresses radiance as the sum of the exitant radiance at $x_{\partial\mathcal{V}}$ and the accumulated emitted and in-scattered radiance between $x_{\partial\mathcal{V}}$ and x. These components are attenuated by the path transmittance

$$\tau(x,x') := e^{-\int_x^{x'} \sigma_e(x'')dx''},$$

which accounts for absorption and out-scattering with the extinction coefficient $\sigma_e := \sigma_a + \sigma_s$. We define the *incident surface emittance* (with $x_{\partial\mathcal{V}}$ as above) as

$$L_{i,\partial\mathcal{V}}(x,\omega) := \tau(x_{\partial\mathcal{V}},x)L_{e,\partial\mathcal{V}}(x_{\partial\mathcal{V}},\omega),$$

the *incident volume emittance* as

$$L_{i,\mathcal{V}^0}(x,\omega) := \int_{x_{\partial\mathcal{V}}}^{x} \tau(x',x)L_{e,\mathcal{V}^0}(x',\omega)dx',$$

the *surface light transport operator* as

$$(\mathbf{T}_{\partial\mathcal{V}}L)(x,\omega) := \tau(x,x_{\partial\mathcal{V}}) \int_{\mathcal{S}^2} f_s(\omega,x_{\partial\mathcal{V}},\omega')L(x_{\partial\mathcal{V}},\omega')\cos\Theta_{x_{\partial\mathcal{V}}}d\omega',$$

and the *volume light transport operator* as

$$(\mathbf{T}_{\mathcal{V}^0}L)(x,\omega) := \int_{x_{\partial\mathcal{V}}}^{x} \tau(x',x)\sigma_s(x') \int_{\mathcal{S}^2} f_p(\omega,x',\omega')L(x',\omega')\ d\omega'dx'.$$

Using these definitions we can rewrite equation (3) in operator notation as

$$L = \underbrace{(L_{i,\partial\mathcal{V}} + L_{i,\mathcal{V}^0})}_{=:L_i} + \underbrace{(\mathbf{T}_{\partial\mathcal{V}} + \mathbf{T}_{\mathcal{V}^0})}_{=:\mathbf{T}} L,$$

which clearly exhibits the Fredholm integral equation structure. Given that $||\mathbf{T}^\alpha|| < 1$, $\alpha \in \mathbb{N}$, which holds for all physically valid scene models where no perfect reflectors or transmitters exist, the Neumann series can be applied:

$$\begin{aligned} L &= \sum_{j=0}^{\infty} \mathbf{T}^j L_i = \sum_{j=0}^{\infty} (\mathbf{T}_{\partial\mathcal{V}} + \mathbf{T}_{\mathcal{V}^0})^j (L_{i,\partial\mathcal{V}} + L_{i,\mathcal{V}^0}) \\ &= L_{i,\partial\mathcal{V}} + L_{i,\mathcal{V}^0} + \mathbf{T}_{\partial\mathcal{V}}L_{i,\partial\mathcal{V}} + \mathbf{T}_{\mathcal{V}^0}L_{i,\partial\mathcal{V}} + \cdots \end{aligned} \tag{4}$$

3 Generalized Path Integral Formulation

To generate an image of size M we need to compute a set of measurements $I_1,\dots,I_M$, where each I_j corresponds to a pixel value. By defining a set of sensor responsivity

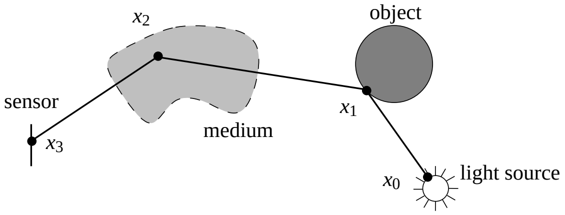


Fig. 1. A typical transport path, defined as an ordered sequence of vertices. In this example, x_0, x_1 and x_3 are on surfaces ($\in \partial\mathcal{V}$), while x_2 is in the volume ($\in \mathcal{V}_0$). The path $\bar{x} = x_0x_1x_2x_3$ thus has the characteristic $l = 1011_b = 11$, i.e. $\bar{x} \in \Omega_3^{11}$.

functions[1] $W_e^{(j)}$, we can express I_j as a scalar product in the *measurement equation*

$$I_j := \int_{\mathcal{V}\times\mathcal{S}^2} W_e^{(j)}(x,\omega)L(x,\omega)\, d\omega\, dV(x). \tag{5}$$

In order to apply the Metropolis sampling algorithm, we need to represent I_j as a path integral. This formulation has been introduced in [VG97] for scenes without participating media. We now generalize this scheme, i.e. we define the path space and the measurement contribution function not only for interaction points on surfaces but also for points in the volume.

3.1 Path Space, Measure and Characteristic

A light transport path $\bar{x}$ of length k is represented by $k+1$ vertices x_i, and is classified according to its *path characteristic* $l \in \mathbb{N}$, which determines whether vertices are in the volume or on a surface. For this purpose, let $b_i(l) \in \{0,1\}$ represent the value of the i-th bit of the binary representation of l, such that b_0 denotes the least significant bit. We define the path characteristic l of a path $\bar{x}$ such that $b_i(l) = 1$ if vertex x_i is on a surface and $b_i(l) = 0$ if x_i is in the volume. The set of all paths of length k with charateristic l is then given as

$$\Omega_k^l := \left\{ \bar{x} = x_0x_1\cdots x_k \;\middle|\; x_i \in \begin{cases} \partial\mathcal{V} & \text{if } b_i(l) = 1 \\ \mathcal{V}^0 & \text{if } b_i(l) = 0 \end{cases} \right\},$$

for $1 \leq k < \infty$ and $0 \leq l < 2^{k+1}$ (see figure 1). A measure μ_k^l on Ω_k^l is defined by

$$\mu_k^l(D) := \int_D \prod_{i=0}^{k} d\mu_{k,i}^l(\bar{x}),$$

where $D \subseteq \Omega_k^l$ and

$$d\mu_{k,i}^l(\bar{x}) := \begin{cases} dA(x_i), & \text{if } b_i(l) = 1 \\ dV(x_i), & \text{if } b_i(l) = 0 \end{cases}$$

[1] These define the response of the sensor, e.g. a film plane, to light incident upon it.

for a path $\bar{x} = x_0 \cdots x_k$. Now we can define the *path space*

$$\Omega := \bigcup_{k=1}^{\infty} \bigcup_{l=0}^{2^{k+1}-1} \Omega_k^l,$$

as the set of all finite-length paths with the associated *path space measure*

$$\mu(D) := \sum_{k=1}^{\infty} \sum_{l=0}^{2^{k+1}-1} \mu_k^l(D \cap \Omega_k^l).$$

3.2 Measurement Contribution Function

The measurement contribution function can be defined directly in terms of paths and path vertices by transforming the integration domain of the inner integration of equation (3) from $\mathcal{S}^2$ to $\mathcal{V}$. The corresponding conversion of measures is reflected in the generalized geometric term[2]

$$G(x \leftrightarrow y) := V(x \leftrightarrow y) \frac{D_x(y) \cdot D_y(x)}{||x-y||^2} \tau(x \leftrightarrow y),$$

where $D_x(y) := |\omega_{xy} \cdot \hat{n}(x)|$, if $x \in \partial\mathcal{V}$. ω_{xy} is the unit direction vector from x to y and $\hat{n}(x)$ the surface normal in x. For $x \in \mathcal{V}^0$ we set $D_x(y)$ equal to one. $D_y(x)$ is defined symmetrically. The visibility function $V(x \leftrightarrow y)$ is one if x and y are mutually visible, i.e. if the connecting ray is not blocked by an object, and zero otherwise. We define the measurement contribution function as

$$\begin{aligned} f_j(\bar{x}) \quad := \quad & L_e(x_0 \to x_1) G(x_0 \leftrightarrow x_1) \cdot \\ & \cdot \left[\prod_{i=1}^{k-1} \left(\hat{f}(x_{i-1} \to x_i \to x_{i+1}) G(x_i \leftrightarrow x_{i+1}) \right) \right] \cdot W_e^{(j)}(x_{k-1} \to x_k), \end{aligned} \tag{6}$$

where $\bar{x} = x_0 \ldots x_k$,

$$L_e(x \to x') := \begin{cases} L_{e,\partial\mathcal{V}}(x \to x') & x \in \partial\mathcal{V} \\ L_{e,\mathcal{V}^0}(x \to x') & x \in \mathcal{V}^0 \end{cases}$$

and

$$\hat{f}(x \to x' \to x'') := \begin{cases} f_s(x \to x' \to x'') & x' \in \partial\mathcal{V} \\ \sigma_s(x') f_p(x \to x' \to x'') & x' \in \mathcal{V}^0 \end{cases} .$$

Now we can insert the Neumann series (4) into the measurement equation (5), yielding

$$I_j = \sum_{k=1}^{\infty} \sum_{l=0}^{2^{k+1}-1} \int_{\Omega_k^l} f_j(\bar{x}) d\mu_k^l(\bar{x}) = \int_{\Omega} f_j(\bar{x}) d\mu(\bar{x}). \tag{7}$$

Each integral over Ω_k^l of the above equation corresponds to exactly one addend of equation (4). In physical terms, f_j describes the differential flux that is transported along a path towards pixel j. Equation (7) defines a measurement as an integral over the infinite-dimensional path space. This allows for a whole new set of integration techniques to be applied for solving the light transport problem in the presence of participating media.

[2] We use the common arrow notation for specifying a direction. The $\leftrightarrow$ symbol indicates symmetry of the arguments.

4 Sampling

In order to evaluate the path integral (7) we need to build transport paths with respect to an appropriate probability density function (pdf). We split the generation of paths into an alternating sequence of scattering and propagation events: A scattering event chooses a direction at a given vertex x by sampling according to the phase function f_p (for $x \in \mathcal{V}^0$) or the BSDF f_s (for $x \in \partial\mathcal{V}$). A propagation event determines the next interaction point x' in a given direction ω starting from x. This is done by sampling the distance d between x and x' according to the path transmittance τ. The pdf of the whole path is then simply the product of all scattering and propagation pdfs, as these are independent of each other.

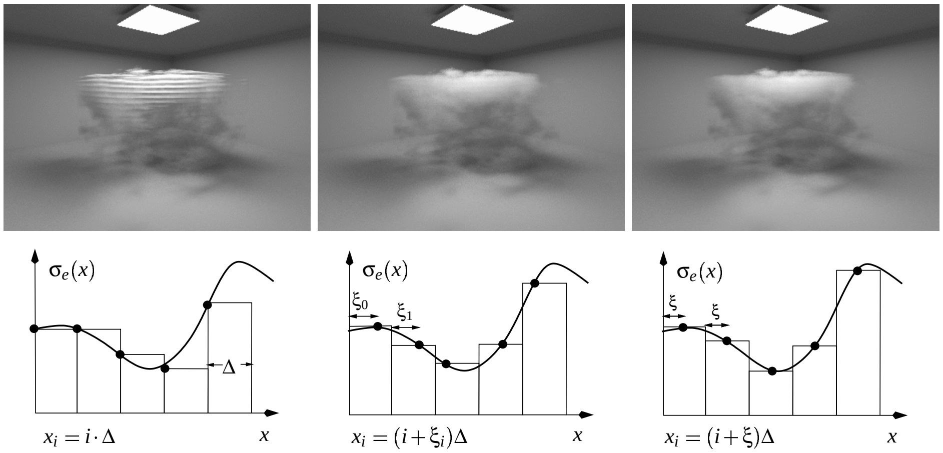
4.1 Line Integral Computation

Propagation in the absence of participating media is straightforward, as the new interaction point is uniquely determined by the ray casting function h. If a ray passes through a medium, we generate the next interaction point with the inversion method [HM72], and we obtain an expression for the distance d by normalizing, integrating and inverting τ.

Homogeneous Media. The homogeneous case is simple, because here we have the explicit expression $d = -\ln(1-\xi)/\sigma_e$, where ξ is a uniformly distributed random variable in $[0,1)$. All we need to do is compare the sampled d with the distance s to the closest surface point $x_{\partial\mathcal{V}}$. If $d < s$, we set $x' := x + d \cdot \omega$, otherwise we choose $x' := x_{\partial\mathcal{V}}$ and adapt the probability density accordingly.

Inhomogeneous Media. These require more work, since we need to compute d from the implicit equation $\ln(1-\xi) = \int_0^d \sigma_e(x + t\omega)dt$. This is done with a *ray marching* algorithm [PH89], which accumulates σ_e along the ray (x, ω) until the threshold $\ln(1-\xi)$ is reached or the surface point $x_{\partial\mathcal{V}}$ is hit. In effect, a ray marching algorithm approximates a one-dimensional integral by dividing the ray into a number of disjoint segments and evaluating σ_e at certain points within each segment. *Equidistant sampling* traverses the ray with constant stepsize Δ, which produces visible artifacts due to aliasing, as depicted in figure 2 (a). The explanation for the layers in the cloud is simple: Light is emitted downwards from the light source at the ceiling and hits the cloud. As the traversal of the cloud data[3] starts at the top surface of its cubic bounding box, the interactions in the medium occur roughly within the same horizontal layers, whose vertical spacing is determined by the size of the ray segments Δ. Consequently, different transport paths that contribute to the same pixel are correlated. These effects can be eliminated by randomly perturbing the sample point within each ray segment, a method known as *jittering*. This leads to *stratified sampling*, a Monte Carlo technique for numerical integration. While stratified sampling reduces aliasing (see figure 2 (b)), it is not a particularly efficient sampling method for this kind of integration problem. Monte Carlo integration is particularly suitable for high-dimensional integrals with discontinuities in the integrand. Here we have a one-dimensional, rather smooth continuous function, favouring deterministic approaches. Therefore we have implemented a combination of equidistant and stratified sampling. Instead of using independent random samples in each ray segment, we choose an initial random offset that is applied to all subsequent

[3] We store inhomogeneous media on a three-dimensional grid with intermediate values being computed through tri-linear interpolation.



(a) equidistant (b) stratified (c) random offset

Fig. 2. Different ray marching strategies. The lower picture shows the sampling method, illustrated for box integration. In the upper row is an image generated with this method. Equidistant sampling clearly reveals aliasing artifacts which are no longer visible in the randomized versions of the ray marching algorithm.

samples of the current ray (see 2 (c)). This breaks the correlation of different transport paths (and hence reduces aliasing) but keeps the integration essentially deterministic and thus more efficient. In general we found an efficiency gain of 30-45% for random offset sampling as compared to stratified sampling. Since about 25% of the total computation time of figure 2 are spent on sampling the medium, this leads to an essential decrease in overall rendering time of about 10% for this scene. The samples generated this way can be used as input for an adaptive ray marching scheme as used in [JC98].

5 Rendering

In [VG97] Veach and Guibas presented the Metropolis Light Transport (MLT) algorithm for scenes in vacuum. We have extended this approach to incorporate participating media, based on the generalized version of the path integral as defined in section 3. MLT makes use of the *Metropolis sampling algorithm* [MRR$^+$53], a very powerful method for the simulation of random variables. The basic idea is to generate a random walk $\bar{x}_0, \bar{x}_1, \ldots$ through the path space Ω and deposit a certain constant amount of energy at each pixel a path passes through. The desired image is obtained by distributing the paths proportionally to their contribution to the final image. Metropolis sampling generates this distribution by first proposing a mutation of the current path and computing the corresponding acceptance probability α. Sampling α then determines whether the mutated path is accepted or rejected as the next sample of the random walk. Note that the paths generated this way are correlated, which allows various forms of coherence to be exploited. On the other hand we are faced with a potential increase in variance as compared to independent sampling.

MLT requires an initialization step which determines the total image brightness and generates the seed path for the Markov chain of paths. Similar to [VG97], our initialization uses bidirectional path tracing, which we extended to incorporate participating media [LW96]. A more detailed description of Metropolis sampling and its application to evaluate the path integral can be found in [Vea97].

5.1 Mutation Strategies

Generating a new mutation and computing the corresponding acceptance probability is central to the MLT algorithm. We use a set of different mutation strategies for this purpose and randomly select one of them to create the proposed mutation.

1. *Bidirectional mutations* delete a contiguous section of the current path and replace it with a new path section by appending vertices to both ends of the created subpaths. Adapting the bidirectional mutation strategy described in [VG97] to our generalized path integral framework is straightforward, so we omit a detailed discussion here.
2. *Perturbations* exploit the fact that small variations to the path most likely lead to similar image contributions and hence a high acceptance probability. We distinguish two types of perturbations:

   (a) *Scattering perturbations* displace the direction vector at a certain vertex,

   (b) *Propagation perturbations* displace the interaction point along a certain ray segment.

   The mutated path is then created by retracing the original path, while preserving the path characteristic. In a sense, scattering and propagation perturbations are complementary. The first perturbs a direction hoping to obtain a similiar interaction point, while the latter perturbs an interaction point hoping to obtain a similar direction. The idea of both is to sample path space locally. Once an important path has been found, neighboring paths are explored as well. This is especially beneficial for bright areas of the image, such as caustics. Another important feature of perturbations is that they alter the image location. This leads to a better distribution of paths over the image plane and significantly reduces the variance of the generated images.

We have implemented two scattering perturbations: *Sensor perturbations* alter the location on the image plane[4] and retrace the path towards the light source. This mutation strategy combines the lens and multi-chain perturbations of [VG97]. *Caustic perturbations* retrace the path towards the eye, after perturbing the direction vector of the second path edge from the eye.

Propagation Perturbation. This mutation strategy is specifically designed for participating media. Let $\overline{x} = x_0 \ldots x_k$ denote the current transport path, where x_0 is a point on a light source and x_k is a point on the sensor. Similarly, $\overline{y} = y_0 \ldots y_k$ is the proposed mutation of $\overline{x}$. If x_{k-1} is an interaction point in the medium, i.e. $x_{k-1} \in \mathcal{V}^0$, this vertex is displaced along the line from x_{k-2} to x_{k-1} to obtain y_{k-1}. This new vertex is then connected with the eyepoint to determine the new sensor location y_k. x_{k-1} is moved a distance D in either direction along $x_{k-2}x_{k-1}$ according to the pdf

$$p(D) \propto \frac{1}{D}, \quad D \in [D_{min}, D_{max}],$$

[4] We use the common pinhole camera model, specified by an eye point and an image plane, such that each point of the image plane corresponds to exactly one pixel of the final image.

where D_{min} and D_{max} specify the minimal and maximal distance, respectively (see figure 3). If y_{k-1} falls outside the medium, or $x_{k-1} \notin \mathcal{V}^0$ the mutation is rejected, i.e. its acceptance probability is set to zero. Note that propagation perturbation is computationally very cheap as it only requires one occlusion test to check whether the connection with the eyepoint is unobstructed.

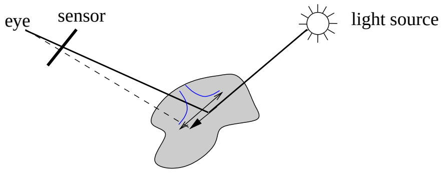


Fig. 3. Propagation perturbation. The interaction point is spatially displaced according to the indicated distribution.

6 Simulation Results

We have implemented our version of the MLT algorithm based on the experimental ray tracing kernel McRender [Kel98], which supports fast BSP ray intersections and occlusion testing. We use a convex combination of Schlick's base functions to model the phase function of the medium, as described in [BLSS93]. The BSDF is modeled with an extension of Ward's reflection model [War92] for isotropic scattering, which includes singular scattering. These scattering models allow a new direction to be generated with the inversion method, which is essential for efficient sampling.

As a minor optimization we estimate paths of length one, i.e. directly visible light sources, with standard ray tracing techniques. Explicit direct lighting calculation has not been implemented so far and it might be worthwhile to incorporate methods such as those described in [War91] or [SWZ96]. This should lead to significant efficiency gains for scenes dominated by direct light. The pinhole camera model imposes some constraints on the use of perturbation strategies, e.g. the caustic perturbation is not effective for caustics seen through a mirror. For the case of a more general camera model with a finite aperture the perturbation strategies can easily be adapted so that such situations are covered. If the light source and the camera aperture are small, however, paths that contain two or more singular scattering interactions separated by a diffuse interaction are not handled well by the MLT algorithm. When perturbing a direction vector (in either direction) and re-tracing the path, the diffuse interaction point will move. From this displaced position we will most likely not hit the sensor respectively the light source, because we must enforce a singular scattering to preserve the path characteristic. Thus the acceptance probability will be low on the average, leading to increased variance. All images were rendered on a single processor HP C3000 with a PA 8500 CPU at 400 MHz. We have only used scenes without surface textures so that the Monte Carlo variance (i.e. noise) and illumination details can be observed more clearly.

Figure 4 (see the color page) features a rendered cloud lit by an approximation of the CIE clear sky model. Figure 5 shows a test scene with a difficult lighting situation. The room is entirely illuminated by indirect light passing through the half-open door. Note that the light source is located at the far end of the adjacent room, i.e. no light can reach the eye without being scattered at least twice. The scene contains glossy surfaces, e.g. the floor, transparent objects, e.g. the glass ball, and an inhomogeneous medium ”streaming” through the door. Most other existing global illumination algorithms would perform poorly in this scene. Bidirectional path tracing, for instance, creates transport paths by connecting subpaths that start both from the eye and from the light sources. Most of these connections will be blocked, however, which leads to increased noise in the image. The photon map method [JC98] fails for this scene, because most photons will be located in the adjacent room and thus cannot contribute to the radiance estimate. Here, even the importance driven generation of the photon map [PP98] does not help, because the door slit is too narrow for a sufficient number of photons to pass through (see [KW00] for a detailed discussion of these topics). Metropolis light transport is far superior in this setting. The locality of the perturbation strategies leads to a better coverage of the relevant transport paths. The image of figure 5 is 720 by 576 pixels and has been rendered with 700 mutations per pixel in approximately 6 hours. Note that the table legs are thin metal plates angled towards the center of the table, which explains the different extends of the shadows. MLT will in general perform better if substantial amounts of the transport paths with a high image contribution are clustered in a ”small region” of path space. The strong correlation of subsequent samples of the random walk ensures that these regions are sampled adequately.

Figures 6 to 8 show different views of a realistic architectural model with more than 240,000 geometrical primitives and various surface materials. These images clearly demonstrate the robustness of the Metropolis light transport algorithm for participating media in complex environments. The night scene of figures 6 to 8 illuminated by spotlights and street lamps contains more than 700 area light sources and illustrates that MLT easily handles scenes with many light sources. In figure 6 the church is surrounded by a thin homogeneous medium, simulating a foggy atmosphere. This image is 720 by 490 pixels and has been generated with 640 mutations per pixel in 15 hours. Figure 7 shows an example of a volume caustic, created by light being focused from the glass sphere of the sculpture into the medium. This image was rendered in 18 hours with 640 mutations per pixel at 720 by 576 pixels. In Figure 8 the homogeneous medium has been replaced by a cloud modelled with a very large inhomogeneous medium. Using 640 mutations per pixel the image has been rendered in 8 hours at 380 by 490 pixels.

7 Conclusions

We have presented an extension of the Metropolis light transport algorithm that provides a physically-based simulation of global illumination for radiatively participating media. Using an improved version of ray marching, the algorithm handles inhomogeneous media with multiple, anisotropic scattering and can simulate volumetric effects such as volume caustics and color bleeding between media and surfaces. The results show that high quality images are obtained, even for difficult lighting situations, such as strong indirect light or large numbers of light sources.

Since Metropolis light transport is based on point sampling, no discretization of the scene geometry or the directional space is necessary and no memory-intensive data structures are required. This makes the algorithm suitable for complex scenes, e.g. models represented procedurally, by fractals, or acyclic graphs. Furthermore, it easily supports participating media that are defined implicitly or by procedural models. Parallelizing the algorithm is straightforward, e.g. different processes compute separate images that are then averaged to obtain the final result.

We believe that many optimizations of the algorithm are still possible. For instance, different mutation strategies are selected randomly according to a discrete pdf that assigns a constant weight to each mutation strategy. The optimal values for these weights strongly depend on the specific scene, however. For several test scenes (e.g. simple scenes like the cornell box) best results were obtained by weighting the perturbations a hundred times stronger than bidirectional mutations. In these cases the average acceptance probability α for bidirectional mutations is high enough to guarantee an even sampling of all paths $\bar{x}$ with $f_j(\bar{x}) > 0$. Yet in other scenes, e.g. figure 5, bidirectional mutations will on average produce a much lower α. So weighting the perturbations a hunderd times stronger leads to an uneven sampling, as the space of paths is not adequately covered. For the scene of figure 5, for instance, single vertices of the path degenerate to point light sources because their probability of becoming mutated is too low. These vertices located in the adjacent room result in sharp shadow boundary artifacts instead of yielding the correct smooth illumination transition. In order to avoid such severe artifacts a balanced pdf is much more appropriate. All images in this paper have been rendered with equal weights, as we found this to be the most robust setting in general. We are currently working on a heuristic for adaptively determining these weights, which will further increase efficiency.

The propagation perturbation is just one possible mutation strategy that is specifically designed for participating media. Other variations are conceivable, e.g. swapping from the medium to a surface and vice versa.

Acknowledgements

The authors would like to thank Gerald Maitschke at Compaq Computers for supporting this research work by the donation of an Alpha Workstation. Special thanks go to Christa Marx for the models rendered on the colorpage.

References

Arv93. J. Arvo, *Transfer Functions in Global Illumination*, ACM SIGGRAPH ’93 Course Notes - Global Illumination, 1993, pp. 1–28.

Bha93. N. Bhate, *Application of Rapid Hierarchical Radiosity to Participating Media*, Proceedings of ATARV-93: Advanced Techniques in Animation, Rendering, and Visualization (1993), 43–53.

BLSS93. P. Blasi, B. Le Saec, and C. Schlick, *A Rendering Algorithm for Discrete Volume Density Objects*, Computer Graphics Forum (Eurographics ’93) **12** (1993), no. 3, C201–C210.

BT92. N. Bhate and A. Tokuta, *Photorealistic Volume Rendering of Media with Directional Scattering*, Third Eurographics Workshop on Rendering (1992), 227–245.

Cha50. S. Chandrasekhar, *Radiative Transfer*, Clarendon Press, Oxford, UK, 1950.

HM72. E. Hlawka and R. Mück, *Über eine Transformation von gleichverteilten Folgen II*, Computing (1972), no. 9, 127–138.

JC98. H. Jensen and P. Christensen, *Efficient Simulation of Light Transport in Scenes with Participating Media using Photon Maps*, SIGGRAPH 98 Conference Proceedings (Michael Cohen, ed.), Annual Conference Series, ACM SIGGRAPH, Addison Wesley, July 1998, pp. 311–320.

Kel98. A. Keller, *Quasi-Monte Carlo Methods for Photorealistic Image Synthesis*, Ph.D. thesis, Shaker Verlag Aachen, 1998.

KVH84. J. Kajiya and B. Von Herzen, *Ray Tracing Volume Densities*, Computer Graphics (ACM SIGGRAPH ’84 Proceedings) **18** (1984), no. 3, 165–174.

KW00. A. Keller and I. Wald, *Efficient importance sampling techniques for the photon map*, Interner Bericht 302/00, University of Kaiserslautern, 2000.

LBC94. E. Languenou, K. Bouatouch, and M. Chelle, *Global Illumination in Presence of Participating Media with General Properties*, Fifth Eurographics Workshop on Rendering (1994), 69–85.

LW96. E. Lafortune and Y. Willems, *Rendering Participating Media with Bidirectional Path Tracing*, Rendering Techniques '96 (Proc. 7th Eurographics Workshop on Rendering) (1996), 91–100.

MRR$^+$53. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, *Equation of state calculations by fast computation machines*, Journal of Chemical Physics **21** (1953), 1087–1092.

PH89. K. Perlin and E. Hoffert, *Hypertexture*, Computer Graphics (SIGGRAPH Journal, vol. 23), 1989, pp. 253 – 262.

PM93. S. Pattanaik and S. Mudur, *Computation of Global Illumination in a Participating Medium by Monte Carlo Simulation*, The Journal of Visualization and Computer Animation **4** (1993), no. 3, 133–152.

PP98. I. Peter and G. Pietrek, *Importance driven Construction of Photon Maps*, Rendering Techniques '98, 1998, pp. 269–280.

RT87. H. Rushmeier and K. Torrance, *The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium*, Computer Graphics (ACM SIGGRAPH '87 Proceedings) **21** (1987), no. 4, 293–302.

Rus94. H. Rushmeier, *Rendering Participating Media: Problems and Solutions from Application Areas*, Fifth Eurographics Workshop on Rendering (1994), 35–56.

Sil95. F. Sillion, *A Unified Hierarchical Algorithm for Global Illumination with Scattering Volumes and Object Clusters*, IEEE Transactions on Visualization and Computer Graphics **1** (1995), no. 3.

SWZ96. P. Shirley, C. Wang, and K. Zimmerman, *Monte Carlo Techniques for Direct Lighting Calculations*, ACM Trans. Graphics **15** (1996), no. 1, 1–36.

Vea97. E. Veach, *Robust monte carlo methods for light transport simulation*, Ph.D. thesis, Stanford University, 1997.

VG95. E. Veach and L. Guibas, *Optimally Combining Sampling Techniques for Monte Carlo Rendering*, SIGGRAPH 95 Conference Proceedings, Annual Conference Series, 1995, pp. 419–428.

VG97. E. Veach and L. Guibas, *Metropolis light transport*, SIGGRAPH 97 Conference Proceedings (Turner Whitted, ed.), Annual Conference Series, ACM SIGGRAPH, Addison Wesley, August 1997, pp. 65–76.

War91. G. Ward, *Adaptive Shadow Testing for Ray Tracing*, 2nd Eurographics Workshop on Rendering (Barcelona, Spain), 1991.

War92. G. Ward, *Measuring and Modeling Anisotropic Reflection*, Computer Graphics (SIGGRAPH 92 Conference Proceedings), 1992, pp. 265 – 272.