Interner Bericht

Metropolis Light Transport for Participating Media

M. Pauly, T. Kollig, A. Keller

303/00

Fachbereich Informatik

Universität Kaiserslautern · Postfach 3049 · D-67653 Kaiserslautern

Metropolis Light Transport for Participating Media

M. Pauly, T. Kollig, A. Keller

303/00

Universität Kaiserslautern
AG Numerische Algorithmen
Postfach 30 49
67653 Kaiserslautern
Germany

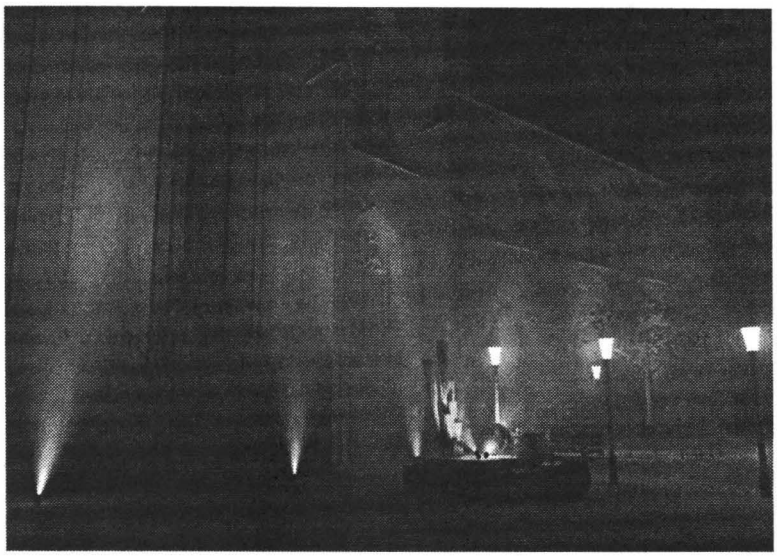
Januar 2000

Herausgeber: AG Numerische Algorithmen
Leiter: Professor Dr. S. Heinrich

Metropolis Light Transport for Participating Media

Mark Pauly Thomas Kollig Alexander Keller *

Department of Computer Science
University of Kaiserslautern



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. It exploits the locality of light propagation by perturbing certain interaction points within the medium. To efficiently sample inhomogeneous media a new ray marching method has been developed that avoids aliasing artefacts and is significantly faster than stratified sampling.

The resulting global illumination algorithm provides a physically correct simulation of light transport in the presence of participating media that includes effects such as volume caustics and multiple volume scattering. It 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. In particular, we found that it is well suited for complex scenes with many light sources.

CR Categories and Subject Descriptors: I.3.3 [Computer Graphics]: Picture/Image Generation; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism; G.1.9 [Numerical Analysis]: Integral Equations - Integro-differential equations.

Keywords: Atmospheric Effects, Illumination, Monte Carlo Techniques, Rendering, Volume Rendering, Ray Tracing.

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 for participating media [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

* {m_pauly, kollig, keller}@informatik.uni-kl.de

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 artefacts (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 radiatively 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 our new 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 section 7.

2 Light Transport for Participating Media

We consider the radiance distribution in a finite volume $\mathcal{V} \subset \mathbb{R}^3$. $\partial\mathcal{V}$ defines 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*

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

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. Note that the above model assumes that σ_a and σ_s do not vary with direction. This is a reasonable assumption for media such as fog or smoke as these can be approximated by a distribution of identical, spherical particles[1]. If, additonally, σ_a and σ_s are independent of position, we have a homogeneous medium.

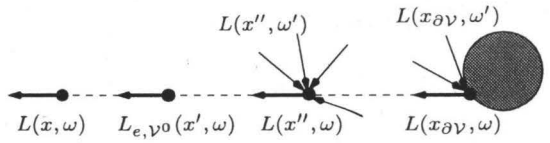


Figure 1: The radiance $L(x,\omega)$ can be expressed as an integral equation by integrating the global balance equation along a ray starting in x until the point $x_{\partial\mathcal{V}}$ on the boundary is hit.

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 at x, and $L_{e,\partial\mathcal{V}}$ defines the emittance on surfaces.

2.1 Integral Form of the Light Transport Equation

We now derive the Fredholm integral equation of the second kind, describing the light transport in the presence of particpating media. Using operator notation, the resulting integral equation will be solved with the Neumann series.

Incorporating the boundary conditions (2) into equation (1) [Arv93] yields the integral equation

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

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 (see figure 1). 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

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

the *surface light transport operator* as

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

[1] Regular structures such as crystals are not accounted for in this model.

and the *volume light transport operator* as

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

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 $\|\mathbf{T}_{\partial\mathcal{V}} + \mathbf{T}_{\mathcal{V}^0}\| < 1$, which holds for all physically valid scene models where no perfect reflectors or transmitters exist, the Neumann series can be applied:

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

3 Path Integral Formulation

To generate an image of size M we need to compute a set of measurements $I_1, \ldots, I_M$, where each I_j corresponds to a pixel value. By defining a set of sensor responsivity functions[2] $W_e^{(j)}$, we can express I_j as a scalar product in the *measurement equation*

$$I_j := \int_{\partial\mathcal{V}\times\mathcal{S}^2} W_e^{(j)}(x,\omega)L(x,\omega)\cos\Theta_x\, d\omega\, dA(x). \quad (5)$$

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

$$\Omega_k^l := \left\{ \bar{x} = x_0 x_1 \cdots x_k \;\middle|\; x_i \in \begin{cases} \partial\mathcal{V} & \text{if } b_i(l) = 1 \\ \mathcal{V}^0 & \text{if } b_i(l) = 0 \end{cases} \right\},$$

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

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

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

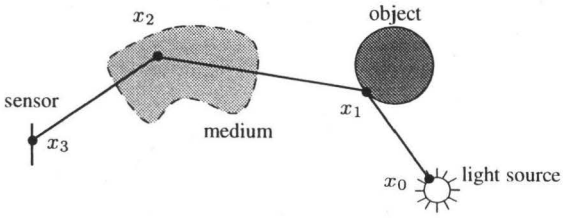


Figure 2: A typical transport path, defined as an ordered sequence of vertices. In this example, x_0, x_1 and x_3 are on surfaces ($\in \partial\mathcal{V}$), while x_2 is in the volume ($\in \mathcal{V}_0$). The path $\bar{x} = x_0x_1x_2x_3$ thus has the characteristic $l = 1011_b = 11$, i.e. $\bar{x} \in \Omega_3^{11}$.

where $D \subseteq \Omega_k^l$ and

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

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 geometric term[3]

$$G(x \leftrightarrow y) = V(x \leftrightarrow y)\frac{|\omega_{xy}\cdot\hat{n}(x)|\cdot|\omega_{xy}\cdot\hat{n}(y)|}{\|x-y\|^2},$$

where ω_{xy} is the unit direction vector from x to y and $\hat{n}(x)$ is the surface normal at x for $x \in \partial\mathcal{V}$, or equal to ω_{yx} for $x \in \mathcal{V}^0$; $\hat{n}(y)$ is defined analogously. The visibility function

$$V(x \leftrightarrow y) := \begin{cases} 1 & \|x-y\| \le \|x - h(x,\omega_{yx})\| \\ 0 & \text{otherwise,} \end{cases}$$

checks if x and y are mutually visible, i.e. if the connecting ray is not blocked by an object. Now we can insert the Neumann series (4) into the measurement equation (5), yielding

$$\begin{aligned} I_j = \sum_{k=1}^{\infty} \sum_{l=0}^{2^{k+1}-1} \int_{\Omega_k^l} & L_e(x_0 \to x_1)G(x_0 \leftrightarrow x_1)\tau(x_0 \leftrightarrow x_1) \\ & \cdot \prod_{i=1}^{k-1} \hat{f}(x_{i-1} \to x_i \to x_{i+1})G(x_i \leftrightarrow x_{i+1})\tau(x_i \leftrightarrow x_{i+1}) \\ & \cdot W_e^{(j)}(x_{k-1} \to x_k) d\mu_k^l(x_0 \cdots x_k), \end{aligned}$$

where

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

[3] We use the common arrow notation [LW96] for specifying a direction. The $\leftrightarrow$ symbol indicates symmetry of the arguments.

and

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

Each integral over Ω_k^l of the above equation corresponds to exactly one addend of equation (4). By defining the measurement contribution function as

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

we obtain the path integral formulation

$$I_j = \int_\Omega f_j(\bar{x}) d\mu(\bar{x}). \tag{7}$$

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.

4 Sampling

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 straigthforward, 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: We obtain the pdf p for sampling a point in the medium by normalizing τ. The cumulative distribution function F is derived by integrating p. Inverting F leads to the equation for the sample distance d. Table 1 shows these quantities for sampling homogeneous and inhomogeneous media.

Homogeneous Media

The homogeneous case is simple, because here we have an explicit expression for d. 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[4].

[4]This introduces a Dirac delta distribution into the propagation pdf. A similar case occurs with the scattering pdf when using singular scattering, i.e. when only one reflected or transmitted direction is allowed for each incoming direction (e.g. for a perfect mirror).

	homogeneous	inhomogeneous
$\tau(x, x')$	$e^{-\sigma_e d}$	$e^{-\int_0^d \sigma_e(x+t\omega)dt}$
$p(x, \omega, x')$	$\sigma_e e^{-\sigma_e d}$	$\sigma_e(x') e^{-\int_0^d \sigma_e(x+t\omega)dt}$
$F(x, \omega, x')$	$1 - e^{-\sigma_e d}$	$1 - e^{-\int_0^d \sigma_e(x+t\omega)dt}$
distance	$d = -\frac{\ln(1-\xi)}{\sigma_e}$	$\ln(1-\xi) = \int_0^d \sigma_e(x+t\omega)dt$

Table 1: Sampling the position of the next interaction point x', starting from x. ξ is a uniformly distributed random variable on $[0, 1)$, ω is the direction vector from x to x' and $d := \|x' - x\|$.

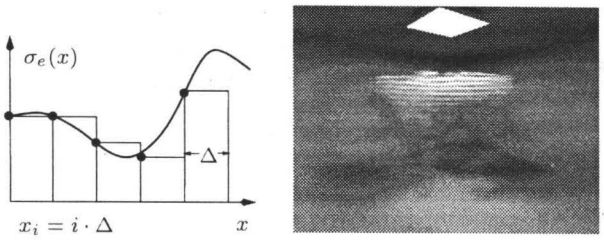
Inhomogeneous Media

These require more work, since we need to compute d from the implicit equation of table 1. 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 disjunct segments and evaluating σ_e at certain points within each segment. *Equidistant sampling* traverses the ray with constant stepsize Δ, which is derived from a user-specified base stepsize, taking into account various factors such as the extend of the medium and the distance to the observer. A common problem of equidistant sampling is that it produces visible artefacts due to aliasing, as depicted in figure 3 (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[5] 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, i.e. the stepsize of the traversal. 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 3 (b)), it is not a particular efficient sampling method for this kind of integration problem. As described in [KW86], 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 samples (see 3 (c)). This breaks the correlation of different transport paths (and hence reduces aliasing) but keeps the integration essentially deterministic and thus more efficient. We found an efficiency gain of about 30-45% for random offset sampling as compared to stratified sampling, which leads to a decrease in the total computation time of about 10% for the cloud scene of figure 3.

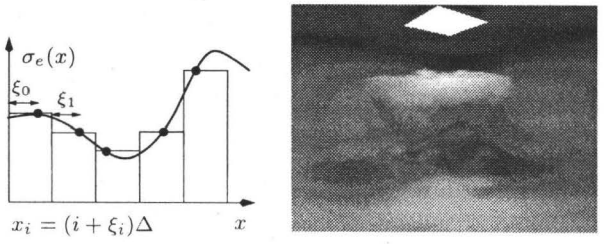
5 Rendering

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

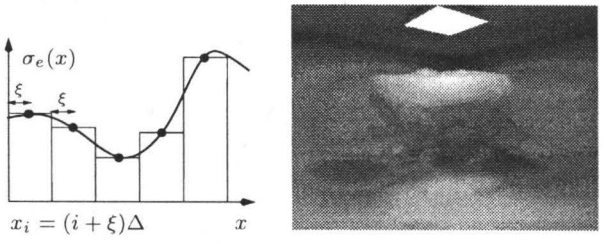
[5]We store inhomogeneous media on a three-dimensional grid with intermediate values being computed through tri-linear interpolation.



(a) Equidistant Sampling



(b) Stratified Sampling



(c) Random Offset Sampling

Figure 3: Different ray marching strategies. The left picture shows the sampling method, illustrated for box integration. On the right is an image generated with this method. Equidistant sampling clearly reveals aliasing artefacts which are no longer visible in the randomized versions of the ray marching algorithm.

the generalized version of the path integral as defined in section 3. MLT makes use of *Metropolis sampling* (also known as M(RT)^2 sampling), a Monte Carlo method for solving difficult sampling problems that was first applied in computational physics [MRR+53].

A unique feature of MLT is that it approximates the intensity distribution of the whole image at once. This is done by generating a random walk $\bar{x}_0, \bar{x}_1, \ldots$ through path space, locating the corresponding pixels of each path and depositing a certain constant amount of energy at these pixels. The brightness of a pixel thus depends on the number of paths that cover its filter support. The desired image is obtained by distributing the paths proportionally to their contribution to the final image, which is quantified in the image contribution function $f(\bar{x})$. f is derived from the measurement contribution function (6) by factoring out all terms that depend on the sensor, i.e. $f_j(\bar{x}) = h_j(\bar{x})f(\bar{x})$, where h_j describes the filter function for pixel j. Metropolis sampling is used to generate a random walk distributed according to f by subsequently creating random variations of the current path. To do so, we first propose a mutation of the current path, which is then either accepted or rejected by sampling the corresponding acceptance probability α. The Metropolis algorithm provides a framework for calculating α as a function of the current path and the proposed mutation [KW86]:

$$\alpha(\bar{x} \to \bar{y}) := \min\left\{1, \frac{f(\bar{y})T(\bar{y} \to \bar{x})}{f(\bar{x})T(\bar{x} \to \bar{y})}\right\}, \tag{8}$$

where T is the transition pdf for mutating $\bar{x}$ to $\bar{y}$. Note that the paths of the random walk 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. A more detailed description of Metropolis sampling and its application to evaluate the path integral can be found in [VG97].

5.1 Initialization

As mentioned above, all paths contribute equally to the final image, i.e. each path of the random walk carries the same weight w. To determine w we need to compute the total power b that is received by the image plane, given as the integral

$$b := \int_\Omega f(\bar{x}) d\mu(\bar{x}).$$

b is then evenly distributed to all paths, i.e. $w = \frac{b}{N}$, where N is the total number of paths of the random walk. In effect, b is the normalization constant for f, i.e. our goal is to distribute paths according to the pdf f/b. We use bidirectional path tracing (BDPT) [VG94], [LW93] with multiple importance sampling [VG95] for calculating b. An extension of BDPT that includes participating media has been presented in [LW96]. We use a slightly different approach based on the path integral (7), although the resulting algorithm should be similar.

Additionally to estimating b, we need to compute the intial path $\bar{x}_0$ of the random walk. $\bar{x}_0$ is chosen from the paths of the BDPT pre-pass, by sampling according to the discrete density $p_n(\bar{x}) = \frac{f(\bar{x})}{b_n}$, where b_n is the BDPT approximation of b with n samples. This means that paths with a high image contribution are more likely to be chosen as initial path. Note that $\lim_{n\to\infty} p_n = f/b$ and that if $\bar{x}_0$ was distributed according to f/b, all subsequent samples would be too. Thus by increasing N we not only improve the accuracy of b, but also diminish the initialization error, which is caused by assuming that all paths are distributed according to f/b, which is only true in the limit. In [VG97] it was shown that the MLT algorithm using the above initialization step is unbiased, i.e. when averaging over many runs, including the initialization, the result will converge towards the correct solution.

5.2 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. In the following, $\bar{x} = x_0 \ldots x_k$ denotes the current transport path, where x_0 is a point on a light source and x_k is a point on the sensor. Similarly, $\bar{y} = y_0 \ldots y_l$ is the proposed mutation of $\bar{x}$.

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 (see figure 4). Since any possible path can be created in this way, the resulting Markov process is ergodic, i.e. converges to the desired stationary distribution. Adapting the bidirectional mutation strategy described in [VG97] to our generalized path integral framework is straightforward, so we omit a detailed discussion here and focus instead on perturbation strategies. These mutations exploit the fact that small variations to the path most likely lead to similar image contributions and hence a high acceptance probability (see equation (8)). We distinguish two types of perturbations:

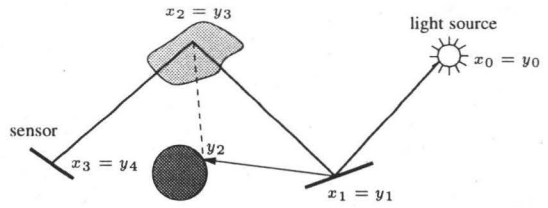


Figure 4: Example of a bidirectional mutation. First the path section between x_1 and x_2 is deleted. This creates the two subpaths x_3x_2 and x_0x_1, the latter of which is extended by vertex y_2. The mutated path $\bar{y}$ is then created by connecting x_2 and y_2 by a shadow ray.

- *Scattering perturbations* displace the direction vector at a certain vertex.
- *Propagation perturbations* displace the interaction point along a certain ray segment.

The mutated path is then created by retracing the original path with the same scattering interaction types. This means that $\bar{y}$ will consist of the same number of vertices as $\bar{x}$ and each vertex of $\bar{y}$ will employ the same scattering mode (singular/regular, reflection/transmission) as the corresponding vertex of $\bar{x}$. Furthermore, the path characteristic of the original path is preserved.

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, neighbouring paths are explored as well. This is especially beneficial for bright areas of the image, such as caustics. An 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.

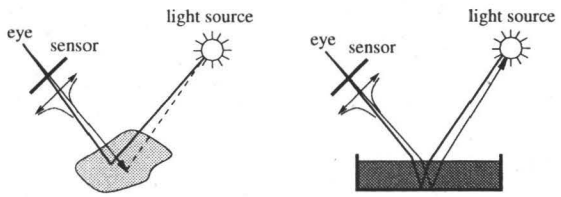


Figure 5: Sensor perturbation. The distribution of the directional displacement is indicated at the perturbed edge. The simple case without singular scattering is depicted on the left. Paths that include singular scattering are shown on the right.

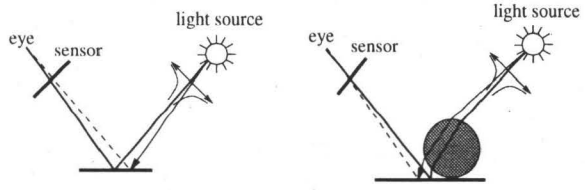


Figure 6: Two examples of light perturbations with the same distribution as the sensor perturbation.

5.2.1 Scattering Perturbation

We have implemented two scattering perturbations:

- *Sensor perturbation* alters the location x_k on the image plane[6] by moving it a distance D in a uniformly distributed direction. The distance D is sampled according to the pdf

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

  where D_{min} and D_{max} specify the minimal and maximal distance, respectively.
- *Light perturbation* perturbs the direction vector from x_{k-2} to x_{k-1} using the same distribution as sensor perturbation. The path is then retraced towards the eye and the new sensor location y_k is determined through the ray from y_{k-1} to the eye point.

Figures 5 and 6 illustrate the two types of scattering perturbations, which realize the same functionality as the lens, caustic and multichain-perturbations that were introduced in [VG97]. The right picture of figure 5 shows a sensor perturbation of a path with two singular scatterings. Starting from the eye the ray hits the surface of the water and is singularly refracted towards the diffuse pool bottom. As the subsequent scattering is singular as well, we enforce this scattering mode on the corresponding vertex of $\bar{y}$, in this case creating a completely new path.

5.2.2 Propagation Perturbation

This mutation strategy is specifically designed for participating media. 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 in either direction along $x_{k-2}x_{k-1}$, using the same distribution as for the scattering perturbations (see figure 7). 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 to connection with the eyepoint is unobstructed.

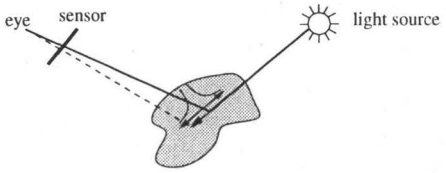


Figure 7: Propagation perturbation. The interaction point is spatially displaced according to the indicated distribution.

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

[6] 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.

Figure 8: One versus one hundred rooms. Both images are 352 by 288 pixels and have been rendered with 500 mutations per pixel.

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. To this point we have not implemented emitting media such as flames; the necessary extensions would be straigthforward, however.

All images were generated 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.

As a minor optimization we estimate paths of length one, i.e. directly visible light sources, with standard ray tracing techniques. We have not implemented explicit direct lighting calculation, however, as we want to keep the algorithm robust for scenes with many light sources, a setting generally considered difficult for standard direct lighting methods. In contrast, Metropolis light transport easily handles such configurations without requiring any specific optimizations. Figure 8 illustrates this feature. On the left we have a simple room with a table and four chairs filled with a homogeneous medium. On the right we have a 10×10 array of these rooms, interconnected through openings in the walls. Each room has its own light source, yet only very few of these contribute noticeably to the perceived illumination. Thus the visual importance of this scene is concentrated in a small portion of the whole environment, while the illumination is distributed evenly. Rendering time for the single room scene was 24 minutes, for the 100 rooms scene 27 minutes, which corresponds to an increase of only 13%. Most of this increase is caused by more expensive ray intersections due to the greater number of geometrical primitives.

Figure 9 shows another scene with a difficult lighting situation. The room is entirely illuminated by indirect light passing through the half-open door. 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. 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 9 is 720 by 576 pixels and has been rendered with 700 mutations per pixel in approximately 6 hours. 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.

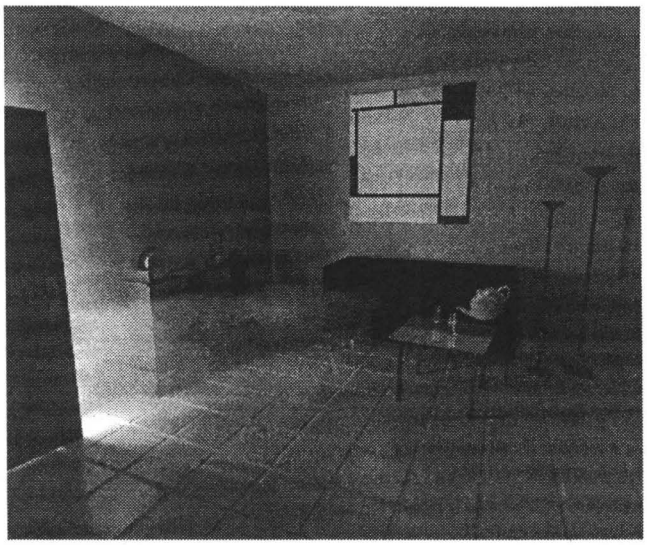


Figure 9: The scene *Invisible Date* with smoke, lit through the door slit.

A realistic architectural model with an inhomogeneous medium is depicted in figure 10. The scene consists of more than 240.000 geometrical primitives and clearly demonstrates the robustness of the Metropolis light transport algorithm for participating media in complex environments. The left image is 480 by 384 pixels and was generated using 1000 mutations per pixel in approximately 6.5 hours. The high oversampling rate is due to a complex daylight simulation and a very large inhomogeneous medium. The night scene on the right contains more than 700 light sources and was rendered with 500 mutations per pixel in 42 minutes at 190 by 245 pixels. Figure 11 shows an example of a volume caustic, created by light being focused from the glass object of the sculpture into the

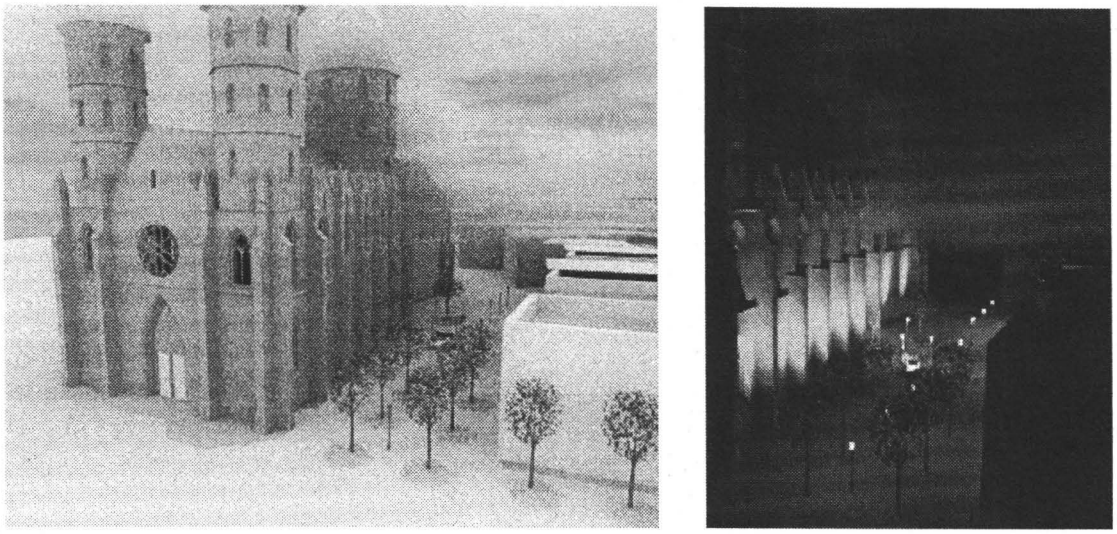


Figure 10: The scene *Stiftsplatz* with daylight simulation (left) and at night (right), illuminated by spotlights and street lamps. Note that the clouds in the background sky are not a texture.

medium. This image was rendered in approximately 8.8 hours with 320 mutations per pixel at 720 by 576 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.

Future research will concentrate on the optimal choice of the various parameters of the algorithm. For instance, different mutation strategies are selected randomly according to a discrete pdf that assigns each mutation strategy a constant weight. The optimal values for these weights depend on the specific scene, however, and a heuristic for adaptively determining the weights would 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.

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.

[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.

[KW86] M. Kalos and P. Whitlock, *Monte Carlo Methods, Volume I: Basics*, J. Wiley & Sons, 1986.

[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.

[LW93] E. Lafortune and Y. Willems, *Bidirectional Path Tracing*, Proc. 3rd International Conference on Computational Graphics and Visualization Techniques (Compugraphics), 1993, pp. 145–153.

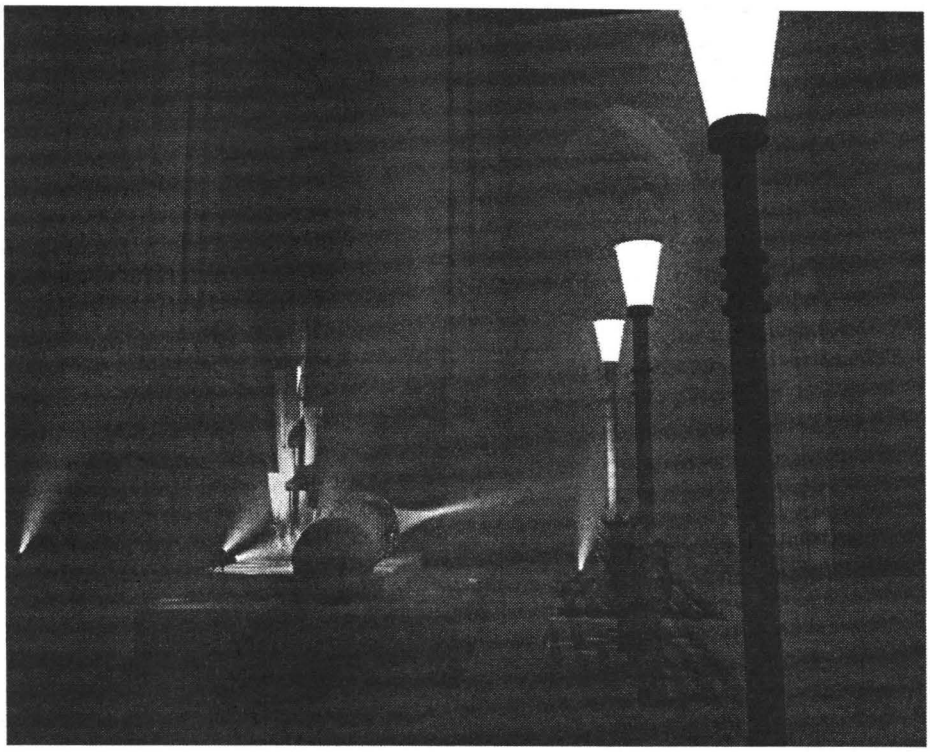


Figure 11: A close-up of the front cover image with a homogeneous medium, featuring a volume caustic.

[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.

[VG94] E. Veach and L. Guibas, *Bidirectional Estimators for Light Transport*, Proc. 5th Eurographics Worshop on Rendering (Darmstadt, Germany), June 1994, pp. 147 – 161.

[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.

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