

# Practical Rendering of Multiple Scattering Effects in Participating Media

Simon Premože<sup>1</sup> Michael Ashikhmin<sup>2</sup> Jerry Tessendorf<sup>3</sup> Ravi Ramamoorthi<sup>1</sup> Shree Nayar<sup>1</sup>

<sup>1</sup> Department of Computer Science, Columbia University, New York, USA  
<sup>2</sup> Department of Computer Science, Stony Brook University, Stony Brook, USA  
<sup>3</sup> Rhythm & Hues, USA

---

## Abstract

*Volumetric light transport effects are significant for many materials like skin, smoke, clouds, snow or water. In particular, one must consider the multiple scattering of light within the volume. While it is possible to simulate such media using volumetric Monte Carlo or finite element techniques, those methods are very computationally expensive. On the other hand, simple analytic models have so far been limited to homogeneous and/or optically dense media and cannot be easily extended to include strongly directional effects and visibility in spatially varying volumes. We present a practical method for rendering volumetric effects that include multiple scattering. We show an expression for the point spread function that captures blurring of radiance due to multiple scattering. We develop a general framework for incorporating this point spread function, while considering inhomogeneous media—this framework could also be used with other analytic multiple scattering models.*

---

## 1. Introduction

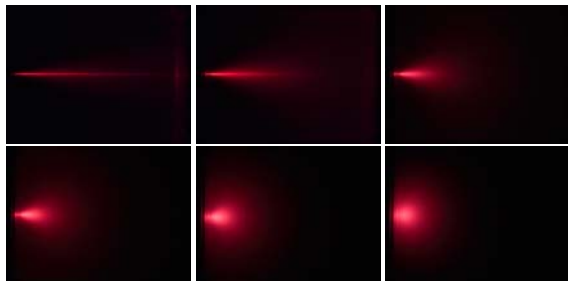
Volumetric scattering effects are important for making realistic computer graphics images of many materials like skin, fruits, milk, clouds, and smoke. In these cases, we cannot make the common assumption that light propagates without scattering in straight lines. Indeed, the *multiple scattering* of light in participating media is important for many qualitative effects [Boh87a, Boh87b] like glows around light sources in foggy weather, or subsurface scattering in human skin, or the spreading of a beam in a scattering medium, as shown in Figure 1.

Light transport, including multiple scattering in arbitrary scattering media, can be accurately computed by solving the radiative transfer equation [Cha60, Ish78]. Volumetric Monte Carlo and finite element techniques (volumetric ray tracing and radiosity) have been used by many researchers [RT87, LBC94, Max94, BSS93, PM93, JC98]. However, volumetric multiple scattering effects are notoriously difficult to simulate. Solution times with the fastest Monte Carlo approach range anywhere from a few hours to a few days. For this reason, these effects are not usually present in computer graphics imagery. Thus, one must look for simpler approximations and models.

This paper describes a practical approach for computing lighting and volumetric effects in spatially varying inhomogeneous

scattering media, taking directionally-varying lighting effects into account. While simulating multiple scattering directly is computationally very expensive, the observable qualitative properties are simple—the incident radiance distribution is blurred and attenuated. There is spatial and angular spreading of incident illumination inside the material. This spreading can be expressed as a *point-spread function* (PSF) that formally measures the spreading of incident radiance in a given medium. Intuitively, the point spread function tells us how the spatial and angular characteristics of light are changed due to scattering events inside the material. This enables us to formalize the light transport in volumes as a convolution of incident illumination and the PSF of the medium. Our approach uses the mathematical path integration framework [PAS03, Tes87] to describe effects of multiple scattering in terms of spatial and angular blurring, thus avoiding costly direct numerical simulation.

The main observable consequences of multiple scattering in volumetric media are spatial and angular spreading of the incident light distribution. We characterize different types of spreading and focus on spatial spreading. We give a simple expression for the spatial spreading width of a collimated beam and compare it with Monte Carlo simulations. Once the PSF of the medium is known, we present a practical rendering algorithm using the PSF to compute effects of multiple scattering. We separate the practical rendering algorithm



**Figure 1:** Effects of multiple scattering. Collimated light source (laser beam) is shone into a container filled with water. Milk has been added to water to add scattering particles. The photographs show effects of multiple scattering with increasing concentration of scattering particles. The effects of multiple scattering are visible as spatial and angular broadening of the laser beam.

and its underlying principles from the mathematical details. The rendering algorithm can therefore be implemented without relying on heavy mathematics.

### 1.1. Overview

A volumetric medium acts as a point spread function that changes spatial, angular and temporal characteristics of the incident radiance (light field). What an observer sees is therefore a convolution between the incident illumination and the point spread function of the medium. The goal is to express the PSF of the medium in terms of basic optical properties (scattering coefficients and phase function) and then show how it can be efficiently used in a rendering algorithm. While the underlying mathematics behind detailed derivations may not be trivial, the final algorithm is quite simple:

1. Given a medium and incident illumination, find how much light is available for redistribution due to multiple scattering. This consists of computing the reduced intensity of light due to simple attenuation.
2. Apply the point spread function to the light volume. This is done by blurring the light volume with kernels of different sizes and storing the light volume at various levels of detail.
3. During the rendering stage, at every point in the volume we compute how much spreading (blurring) occurs due to multiple scattering and we simply look up the result from the blurred representation of the light volume.

### 2. Related Work

The radiative transfer equations and their approximations have been extensively studied in many fields. We only mention recent practical methods in computer graphics. The interested reader is referred to a survey by Perez *et al.* [PPS97] for detailed classification of global illumination algorithms in participating media. Pharr and Hanrahan [PH00] also provide an extensive list of existing methods.

One simple approximation is to only consider single scattering, as done by Ebert and Parent [EP90], Sakas [Sak90],

Max [Max86], Nakamae *et al.* [NKON90], and Nishita *et al.* [NDN96]. However, they cannot easily reproduce important qualitative effects due to multiple scattering like glows around light sources, or subsurface scattering.

Another approach is to use the diffusion approximation, first introduced in graphics by Stam [Sta95], following conceptually similar work by Kajiya and von Herzen [KH84]. More recently, Jensen *et al.* [JMLH01] (also Koenderink and van Doorn [KvD01]) applied it to subsurface scattering, deriving a simple analytic formula. This provides a practical approach to a problem that had previously required an immense amount of computation [HK93, DEJ\*99]. However, the diffusion approximation is valid only for optically dense homogeneous media, where one can assume the angular distribution of radiance is nearly uniform. It is also technically valid only for infinite plane-parallel media. Further, the approach of Jensen *et al.* [JMLH01] is specialized to subsurface effects on objects, and cannot be easily extended to volumes. Our method can be seen as analogous to the diffusion point-spread function for general media, where we handle spreading of light explicitly in arbitrary volumes. We can therefore handle spatially varying general materials, and highly directional effects.

In the context of subsurface scattering, fast integration techniques, bearing some similarity to our approach have been developed. For example, Jensen and Buhler [JB02] extended the diffusion approximation to be computationally more efficient by precomputing and storing illumination in a hierarchical grid. Lensch *et al.* [LGB\*02] implemented this method in graphics hardware. The expensive illumination sampling is alleviated by a simple lookup in the hierarchical grid. Our integration framework can be thought of as extending these ideas to general spatially varying media, with strong directional effects.

Most recently, Narasimhan and Nayar [NN03] solved the spherical radiative transfer equations to derive a general formula for the PSF due to an isotropic point source in a spherical medium. Their formula applies to general homogeneous materials, and does not assume optically dense media, unlike diffusion. Their application was to the inverse problem of estimating weather conditions from the glow around light sources. While the direct solution of the radiative transfer equations is impressive (something which our approach does not explicitly do), the result is practically limited by its restriction to homogeneous isotropic media. By contrast, our rendering algorithm allows for spatially varying participating media, and our framework accounts for anisotropic sources and visibility effects in real scenes.

The idea of using point-spread functions has been explored in areas other than graphics, using statistical and other complex mathematical machinery. Stotts [Sto78] described the pulse stretching in ocean water. Given the starting point and a plane some distance away from the pulse origin, Stotts characterized stretching in terms of extra path lengths the

photons travel due to scattering. McLean *et al.* [MCH87] used a Monte Carlo method to track individual rays. They could compute the density of photons arriving at a fixed plane some distance away from the origin from all directions and times. The results agreed well with computations done by Stotts [Sto78]. Lutomirski *et al.* [LCH95] extended the statistical analysis of Stotts. They provide exact statistical formulations for spatial, angular and temporal spreading. Additional studies of beam spreading were done by Ishimaru [Ish79] and McLean and Voss [MV91] who compared theoretical results with experimental measurements. Gordon [Gor94] rigorously derived equivalence of the point and beam spread functions from the radiative transfer equation. McLean *et al.* [MFW98] described an analytical model for the beam spread function and validated their model with Monte Carlo computations. They also give a comparison with many other analytic models.

Our expression for the point spread function and the blurring width stems from the formulation of light transport as a sum over all paths [Tes87]. However, the derivation itself is quite different and it does not rely on heavy mathematics of path integration [APRN04].

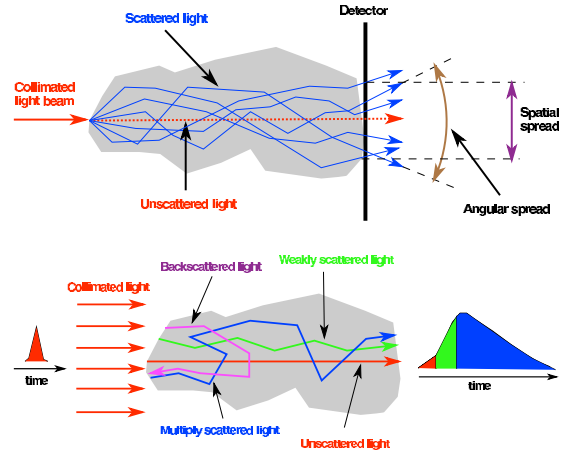
### 3. Point Spread Function

A point spread function (PSF) measures the spreading of incident radiance in a given medium. Intuitively, the point spread function tells us how the spatial and angular characteristics of light are changed due to scattering events inside the material. For example, on a foggy day, a street light has a distinct glow around it. The diameter of the glow is determined by the point spread function of the medium which in turn is a function of the optical properties of the medium. There are different types of spreading that occur as a consequence of the multiple scattering. We discuss these different types of spreading with focus on spatial spreading that is particularly relevant for computer graphics applications. We give a simple expression for the blurring width in the medium and compare it with a Monte Carlo experiment.

#### 3.1. Effects of Multiple Scattering

If we shine a laser beam pulse into a scattering medium, the pulse undergoes a series of absorption and scattering events. The effects of multiple scattering result in significant changes to the pulse:

1. Spatial spreading. The pulse cross-section broadens as it propagates through media, as shown in Figure 1. We incorporate this effect into our algorithm.
2. Angular spreading. The angular divergence of a narrow pulse gets larger as it travels through the medium. We do not currently incorporate such effects into the algorithm but provide necessary mathematical results for doing so with a procedure similar to that used for spatial blurring.
3. Temporal spreading. Scattered photons of the pulse stay behind the original unscattered photons since they have



**Figure 2:** Beam spreading in scattering media due to multiple scattering.

to take longer paths. The direct consequence is that pulse becomes longer as it travels through the medium. While this effect is very important in many fields such as remote sensing [WM99], it is of little interest for computer graphics which deals with stationary solutions of light transport. We will, however, see that explicit treatment of time (or distance) dependence is very useful as an intermediate step.

Figure 2 illustrates spatial, angular and temporal spreading due to multiple scattering. Many of the subtle appearance effects of scattering materials are a direct consequence of beam spreading due to multiple scattering. As we see, it is straightforward to qualitatively understand beam spreading and stretching in the scattering media, but direct simulation of multiple scattering and therefore of spreading (blurring) is computationally expensive. The quantitative analysis of spatial and angular spreading we present could provide more insight into the appearance of scattering materials and could lead to more efficient and simpler rendering algorithms.

#### 3.2. Spreading Width

The width of the beam spread in a medium depends on the amount of scattering in the medium. As shown in Figure 1, the width of the beam is directly proportional to the number of scattering events in the medium. Assuming that scattering paths comprising the beam have Gaussian distribution, we derived the width of the beam spread of a collimated light beam at distance  $S$  as a function of optical properties of the medium (absorption coefficient  $a$ , scattering coefficient  $b$ , and mean square scattering angle  $\langle\theta^2\rangle$ ). We state the final result for the beam width:

$$w^2(S) = \frac{1}{2} \left( \frac{2a}{3S} + \frac{16\alpha}{bS^3} \right)^{-1} = \frac{\langle\theta^2\rangle b S^3}{16(1+S^2/12l^2)}, \quad (1)$$

with  $l$  being a *diffusive path length*  $l^2 = 1/(ab\langle\theta^2\rangle)$  and  $\alpha = 1/(2\langle\theta^2\rangle)$ . A detailed derivation of the expression for

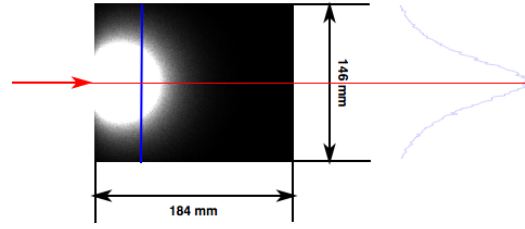
width using the path integration framework is available in a companion document [APRN04].

In a similar fashion to equation 1, one can express the angular spreading of light. In extreme cases, like the diffusion approximation for optically dense media, the angular distribution of light becomes nearly uniform. We currently do not explicitly consider angular blur in our approach, but an analogous expression for angular spreading is presented in [APRN04].

It is clear from equation 1 that the spread width is monotonically increasing with the distance  $S$  in the medium. The beam width also increases with the number of scattering events  $b$ . The more scattering events occur in the medium, the more the beam is spread. This can be seen in Figure 1 where the width of the laser beam is much wider with increased concentration  $b$  of scattering particles. The beam spreading retains the directionality of the incident collimated beam until some distance  $s$  where the transition to diffusion regime occurs. At this point the light field is essentially diffuse and has no directionality that was present upon entering the medium. This effect can also be seen in Figure 1 where in the liquid with large number of scattering particles (*i.e.* larger scattering coefficient  $b$ ), light distribution quickly becomes diffuse. The beam spread width also grows faster with larger mean square scattering angle  $\langle\theta^2\rangle$ . The more forward-peaked the phase function is, the slower the increase in the beam spread width. As expected, in the absence of scattering, the beam spread width is zero and the beam is only attenuated due to absorption. The beam spread width expression also does not have an upper bound. At first glance this seems bothersome, but as we show later, the attenuation increases at an even faster rate.

McLean *et al.* [MFW98] summarize different expressions for spreading width in the medium. Our particular expression in equation 1 has the same functional form as many other expressions with a different constant factor. Due to different assumptions and methods, most derivations differ in the constant factor [MFW98]. Being relatively simple, our approach provides an expression for spatial blurring that is easy to evaluate. Rigorous approaches typically do not provide a general closed form solution and only special cases, discussed below, can be evaluated. When we use these special cases to compare our results against these more sophisticated theoretical methods, we find the accuracy of our method to be sufficient given the needs of computer graphics applications. This lends more solid ground to our approach.

**Limiting Cases.** For long paths ( $S \gg l$ ), the square of the spatial width grows linearly with distance  $S$  along the path:  $w^2 = \frac{3}{4}\langle\theta^2\rangle bSl^2$ . For another special case of no absorption ( $l = \infty$ ), the width is  $w^2 = b\langle\theta^2\rangle S^3/16$ . For these limiting cases, using a much more rigorous derivation, Tessoroff [Tes87] obtained the spatial width of  $w^2 = b\langle\theta^2\rangle S^3/24$  for case of no absorption and  $w^2 = b\langle\theta^2\rangle Sl^2/2$  with absorption. In comparison with Tessoroff, we obtain



**Figure 3:** Monte Carlo simulation of a scattering medium. Density estimation of photons yields energy distribution in the medium. A slice through the volume taken at the laser plane (horizontal slice at the height at which photons were injected into the medium) is shown for a particular configuration. Dimensions of the simulation domain are shown as well as an actual slice through the volume. A vertical profile at some distance from the source is shown.

a correct functional dependence on both  $S$  and medium parameters, but are off by a constant factor of  $3/2$ . This discrepancy can be offset by multiplying our expression by a constant correction factor (in this case the correction factor is  $2/3$ ).

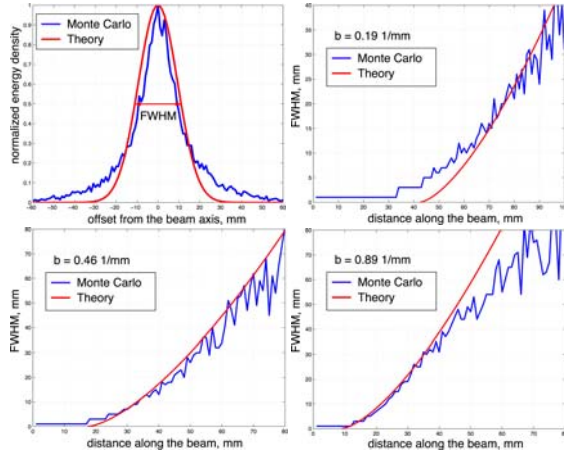
**Absorption Effects.** Note that unlike many comparable formulations for the beam spread, our expression does have dependence on absorption in the medium. Intuitively, if absorption is included, some light gets absorbed and consequently the spread of the beam is narrower than with scattering only. This is consistent with observations that scattering redistributes light in all directions while absorption has just the opposite effect and tries to keep the beam collimated to minimize attenuation. For many materials of practical interest (*e.g.* clouds, water, snow), the absorption is often orders of magnitude smaller than the scattering.

### 3.3. Monte Carlo Comparison

To check spatial spread width predictions from the model, we compare blurring widths with Monte Carlo simulations. Scattering of a laser beam entering participating media was chosen for this testing since it provides the closest real-life analogy with the idea of a single ray. A simulation was performed by shooting 30 million photons into the medium and storing their energies at the position of each scattering event. Energy density estimation is done for the photons satisfying the necessary boundary conditions, in this case those with propagation direction within a small angle of the horizontal axis. A 2D slice through the simulation volume in the plane of the laser beam produces images of the type shown in Figure 3. A single vertical column in this image is also shown. The shape of this function can be directly compared with our expressions for spatial spreading of the beam.

Monte Carlo simulation was preferred to the real experimental curves because it is much easier to obtain the information we need for the most direct comparison with the theory. Direct numerical comparison between actual experiment and Monte Carlo simulation and our model is very difficult because exact optical properties for real materials are





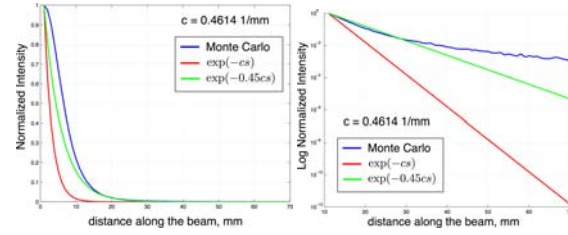
**Figure 4:** Top left plot shows the average case normalized intensity profile compared to a Gaussian with corresponding theoretical width. Other plots show spatial spread widths for different scattering coefficients  $b$  as a function of distance along the beam. The scattering medium was simulated using the following parameters:  $g = 0.9$  ( $\langle\theta^2\rangle = 0.24$ ),  $a = 0.0003\text{mm}^{-1}$ .

rarely available. To validate Monte Carlo simulation results we, in turn, use them to create side views of the propagating beam from outside the container and compare them with real photographs shown on Figure 1. Profiles of such images were found to be very similar in shape.

Normalized cross-sections of the beam at different distances from the entry point along with those predicted by the model were plotted. An example is shown in Figure 4. While a significant portion of the profile is well approximated by a Gaussian, the figure also demonstrates that a wider non-Gaussian component not captured by our model is also present. Since this component has low magnitude compared with the main profile, we believe it will not significantly affect rendered images. A more accurate model would certainly be needed for applications where such details of scattering are important. Figure 4 shows our average case—some curve shapes agree better with our predictions while others deviate more.

We extracted Full Width at Half Maximum (FWHM) of the beam cross-section as the function of distance from the entry point for several Monte Carlo simulations using different scattering coefficient  $b$ , while other values were fixed to their typical values. The results are shown in Figure 4 along with corresponding theoretical predictions. Noticeable blurring starts only after the beam travels some distance in the medium. We therefore compute blurring width by plugging in a shifted length value into our expressions which corresponds to a horizontal offset of calculated curves. A value of  $8/b$  for the amount of shift gives good agreement with the experiment.

To study beam attenuation (discussed in more detail in Section 4.5), we plot the dependence of light energy den-



**Figure 5:** Laser beam attenuation. Left: beam attenuation comparison between Monte Carlo simulation, Beer-Lambert exponential attenuation law, and modified exponential attenuation. Right: log-scale plot of beam attenuation.

sity along the beam axis from the distance to the beam entry point (Figure 5). We found that for the first several orders of magnitude, the attenuation is well described by the Beer-Lambert exponential attenuation law. The effective attenuation coefficient, however, is noticeably lower than our simple  $\exp(-cs)$  prediction and deviation is increasing for denser media reaching about  $0.45c$  for the densest medium we used. The attenuation coefficient given by diffusion approximation does even worse job due to inapplicability of this approximation to the clearly direction-dependent case we study. Obtaining a better expression for attenuation is left as an important area of future research.

#### 4. Mathematical Details

In this section, we introduce the mathematical preliminaries of the radiative transfer equation and path integration, necessary for relating the traditional light transport equation to a less commonly used convolution formulation. We provide some intuition behind the point spread function and how it relates to traditional ray tracing.

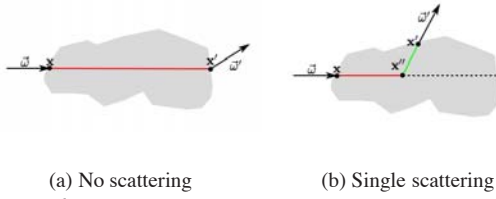
Readers more interested in implementation may wish to skip to Section 5, that will use the point spread function for a practical algorithm to compute effects of multiple scattering.

##### 4.1. Radiative Transfer Equation

Optical properties of volumetric materials can be characterized by density  $\rho(\mathbf{x})$ , their scattering and absorption coefficients  $b(\mathbf{x})$  and  $a(\mathbf{x})$ , the extinction coefficient  $c(\mathbf{x}) = a(\mathbf{x}) + b(\mathbf{x})$ , and the phase function  $P(\mathbf{x}, \vec{\omega}, \vec{\omega}')$ . The phase function  $P$  describes the probability of light coming from incident direction  $\vec{\omega}$  scattering into direction  $\vec{\omega}'$  at point  $\mathbf{x}$ . The phase function is normalized,  $\int_{4\pi} P(\vec{\omega}, \vec{\omega}') d\omega' = 1$  and only depends on the phase angle  $\cos\theta = \vec{\omega} \cdot \vec{\omega}'$ . The mean cosine  $g$  of the scattering angle is defined as  $g = \int_{4\pi} P(\vec{\omega}, \vec{\omega}') (\vec{\omega} \cdot \vec{\omega}') d\omega'$  and the average square of the scattering angle  $\langle\theta^2\rangle$  is

$$\langle\theta^2\rangle = 2\pi \int_0^\pi \theta^2 P(\vec{\omega}, \vec{\omega}') \sin\theta d\theta. \quad (2)$$

The most general case of light transport in arbitrary media is described by the time-dependent radiative transport equa-



**Figure 6:** Green's propagator for the light transport equation can be related to marching along a ray and computing effects of attenuation operator  $G_{noscatter}$  and single-scattering operator  $G_{single}$ .

tion [Cha60, Ish78],

$$\left( \frac{\partial}{\partial s} + \vec{\omega} \cdot \nabla + c(\mathbf{x}) \right) L(s, \mathbf{x}, \vec{\omega}) = b(\mathbf{x}) \int_{4\pi} P(\vec{\omega}, \vec{\omega}') L(s, \mathbf{x}, \vec{\omega}' d\Omega') + Q(s, \mathbf{x}, \vec{\omega}), \quad (3)$$

where we have expressed time  $t$  in units of length  $s$ , with  $s = vt$ . As compared to the standard time-independent equation, we have introduced the term  $\partial/\partial s$  on the left-hand side.  $Q(s, \mathbf{x}, \vec{\omega})$  is the source term, accounting for emitted illumination from light sources.

From the general theory of linear integral equations [BG70], it is known that the solution of equation 3 can be expressed as a convolution of the initial source radiance distribution  $Q = L_0(\mathbf{x}', \vec{\omega}')$  with a Green's function or propagator (evolution operator)  $G(s, \mathbf{x}, \mathbf{x}', \vec{\omega}, \vec{\omega}')$ :

$$L(s, \mathbf{x}, \vec{\omega}) = \int G(s, \mathbf{x}, \mathbf{x}', \vec{\omega}, \vec{\omega}') L_0(\mathbf{x}', \vec{\omega}') d\mathbf{x}' d\vec{\omega}'. \quad (4)$$

Physically, the Green propagator  $G(s, \mathbf{x}, \mathbf{x}', \vec{\omega}, \vec{\omega}')$  represents radiance at point  $\mathbf{x}$  in direction  $\vec{\omega}$  at time  $s$  due to light emitted at time zero by a point light source located at  $\mathbf{x}'$  shining in direction  $\vec{\omega}'$ . Mathematically, it is the solution of the homogeneous version of equation 3 (i.e. with source term set to zero) with initial condition expressed using the Dirac delta function  $\delta$  as

$$G(s=0, \mathbf{x}, \vec{\omega}, \mathbf{x}', \vec{\omega}') = \delta(\mathbf{x} - \mathbf{x}') \delta(\vec{\omega} - \vec{\omega}'). \quad (5)$$

## 4.2. Green's propagator and relation to raytracing

In the absence of scattering ( $b=0$ ), the solution for the complete propagator  $G$  is almost trivial:

$$G(s, \mathbf{x}, \vec{\omega}, \mathbf{x}', \vec{\omega}') \equiv G_{noscatter}(s, \mathbf{x}, \vec{\omega}, \mathbf{x}', \vec{\omega}') = \delta(\mathbf{x} - \vec{\omega}s - \mathbf{x}') \delta(\vec{\omega} - \vec{\omega}') \times \exp\left(-\int_0^s c(\mathbf{x} - \vec{\omega}(s-s')) ds'\right). \quad (6)$$

Here the light travels in a straight line and is attenuated by the absorption coefficient  $a(\mathbf{x}) = c(\mathbf{x})$ . One can see that in this case, the formulation using the propagator is equivalent to simple raytracing (Figure 6). This simple attenuation model is quite popular in computer graphics and it is often part of popular APIs like OpenGL (fog attenuation).

We can also write the propagator  $G$  to include an arbitrary number of scattering events. For example, single scattering propagator  $G_{single}$  includes light that has been scattered only once and the light that has not been scattered at all (as above). To formalize it, we note that propagation from starting position and direction  $(\mathbf{x}', \vec{\omega}')$  to final position and direction  $(\mathbf{x}, \vec{\omega})$  requires three steps. First, light is attenuated over distance  $s'$  to an intermediate point  $\mathbf{x}''$ . Second, the light scatters at point  $\mathbf{x}''$  from initial direction  $\vec{\omega}''$  to final direction  $\vec{\omega}$ . Only a fraction that is determined by the phase function  $P$  of the incident radiance scatters into the new direction. Third, light is further attenuated from the intermediate point  $\mathbf{x}''$  to the final point  $\mathbf{x}$ . To include all possible intermediate points where a scattering event occurs, the propagator  $G_{single}$  is given by integration over all intermediate points:

$$G_{single}(s, \mathbf{x}, \vec{\omega}, \mathbf{x}', \vec{\omega}') = G_{noscatter}(s, \mathbf{x}, \vec{\omega}, \mathbf{x}', \vec{\omega}') + \int_0^s ds' \int_{|\mathbf{x}'' - \mathbf{x}'|=s'} \int_{S^2} (G_{noscatter}(s-s', \mathbf{x}, \vec{\omega}, \mathbf{x}'', \vec{\omega}''') \times b(\mathbf{x}'') \int_{S^2} P(\mathbf{x}'', \vec{\omega}'', \vec{\omega}''') G_{noscatter}(s', \mathbf{x}'', \vec{\omega}'', \mathbf{x}, \vec{\omega}') d\omega''') d\omega'' dx''. \quad (7)$$

This expression again directly corresponds to the standard single scattering ray marching algorithm commonly used in computer graphics. Marching along the viewing ray and sending shadow rays (that are also attenuated) toward a light source corresponds to the three steps discussed above (Figure 6).

We could further rewrite the propagator to include higher order of scattering events by recursively expanding the propagator. But, as demonstrated by equation 7, the expression would quickly become unmanageably complex due to additional angular integrations that have to be performed to account for higher orders of scattering. Therefore, it is often useful in practice to split the propagator  $G$  into two parts: unscattered and single-scattered (or “direct”) light  $G_d = G_{single}$  and one for multiply scattered (or “indirect”) light  $G_s$  and solve them separately:

$$G(s, \mathbf{x}, \vec{\omega}, \mathbf{x}', \vec{\omega}') = G_d(s, \mathbf{x}, \vec{\omega}, \mathbf{x}', \vec{\omega}') + G_s(s, \mathbf{x}, \vec{\omega}, \mathbf{x}', \vec{\omega}') \quad (8)$$

The initial condition for the scattered light propagator  $G_s$  is straightforward, because there is no multiply scattered light in the beginning:

$$G_s(s=0, \mathbf{x}, \vec{\omega}, \mathbf{x}', \vec{\omega}') = 0 \quad (9)$$

We will use existing techniques for computing direct lighting described by  $G_d$ . Our main goal in this work will be to efficiently deal with propagator  $G_s$  (multiply-scattered light) that could lead to faster rendering algorithms for participating media.

## 4.3. Path Integral Formulation

The path integral (PI) approach provides a particular way to express the propagator  $G(s, \mathbf{x}, \vec{\omega}, \mathbf{x}', \vec{\omega}')$ . It is based on the simple observation that the full process of energy transfer from one point to another can be thought of as a sum over

transfer events taking place along many different paths connecting points  $\mathbf{x}$  and  $\mathbf{x}'$ , each subject to boundary conditions restricting path directions at these points to  $\vec{\omega}$  and  $\vec{\omega}'$ , respectively. The full propagator is then just an integral of individual path contributions over all such paths. This object is called the *path integral*. One can further notice that the intensity of light traveling along each path will be only diminishing due to absorption and scattering events along the path. This is because in-scattering into the path, which is generally treated as a process increasing light intensity, in a particular direction will be due to photons traveling a *different* path in the medium (we ignore here exact backscattering which can return photons exactly to the same path). Therefore, if we introduce effective attenuation  $\tau$  along the path, we can write the individual path contribution (weight) as  $\exp(-\tau)$ , and the complete propagator as

$$G \sim \int \exp(-\tau(\text{path})) \mathcal{D}x, \quad (10)$$

where the attenuation  $\tau$  is analogous to the classical *action*  $A$  along the path, with  $\exp(-\tau(\text{path})) \sim \exp(-A(\text{path}))$ .

Because the integration is performed over the infinite-dimensional path space using highly non-intuitive differential measure  $\mathcal{D}x$  defined for it, the mathematics of path integrals is exceptionally complex [FH65, LRT82]. Tessendorf [Tes89] derived a path integral expression for the propagator  $G$  in homogeneous materials. Interested readers are referred to his further work [Tes91, Tes92] for detailed derivations of the path integral formulation. Using much simpler tools, one can still obtain some useful results [PAS03] of the PI theory, which we will present here without detailed derivations.

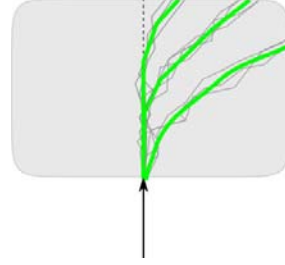
First, part of path weight or action due to multiple scattering in the integral in equation 10 can be shown to be proportional to:

$$\exp(-A(\text{path})) \sim \exp\left(-\int_0^s \left[ a(\vec{\gamma}_P(s')) + \frac{\alpha}{b(\vec{\gamma}_P(s'))} \left| \frac{d\theta}{ds'} \right|^2 \right] ds' \right), \quad (11)$$

where  $\vec{\gamma}(s)$  is a path length parameterized path,  $d\theta/ds$  is its curvature, and  $\alpha = 1/4(1-g) = 1/(2\langle\theta^2\rangle)$ , where  $\langle\theta^2\rangle$  is the mean square scattering angle. Integration is performed along the path.

One can find a path which maximizes this expression (i.e. has minimal attenuation or action among all possible paths). We call it the most probable path (MPP). For the important special case of homogeneous media under boundary conditions when path directions are specified at both ends, one can determine the shape of MPP of given length analytically with the standard Euler-Lagrange minimization procedure [Fox87]. The result is a “uniformly turning” path of constant curvature which is changing its direction at a constant rate.

We further assume that only a small fraction of paths



**Figure 7:** Once we find a set of most probable paths for given initial conditions, we compute contributions along those paths and some neighborhood around these paths. All other paths are ignored because they are deemed not important. In practice, we do not even consider the full set of MPPs, but rather a subset of those which are the easiest to treat.

“around” the MPP contribute significantly to the integral and will restrict the integral to account for the contribution of these important paths only. Formally, this constitutes a Wentzel-Kramer-Brillouin (WKB) expansion [LRT82] of the path integral while physically and visually it accounts for the fact that blurring of the radiance distribution is the most notable characteristic of participating media. The basic idea of our approach, using the most probable path, and a neighborhood around it, is shown in Figure 7.

#### 4.4. Surrounding Path Contribution

In computer graphics, a boundary condition of particular interest is a *single-sided* one, which requires the path to start at a particular point in space with particular initial direction (an example is the eye position and primary viewing ray direction) but applies no additional restrictions on the second end of the path. The path will usually terminate once it reaches an object (or a light source) in the scene. Suppose we found the MPP for this boundary condition and computed its contribution. We would now like to approximate the complete path integral by taking into account only the contribution of “surrounding” paths. This operation formally constitutes a WKB expansion of the integral.

If we parameterize the family of nearby paths using some vector of parameters  $\vec{\xi}$  (with  $\vec{\xi}=0$  at the MPP), the path integral can be written as an integral over these parameters. Note that because  $A$  in equation 11 has the global minimum at the MPP, its expansion in terms of parameters  $\vec{\xi}$  will start from square terms. That is, if there is only one parameter  $\xi$ , a simple Taylor series is,

$$A(\xi) = A(0) + \frac{1}{2}\xi^2 \frac{\partial^2 A}{\partial \xi^2} + \dots,$$

$$\exp(-A(\text{path})) \sim \exp(-A(\text{MPP})) \exp\left(-\frac{1}{2}\xi^2 \frac{\partial^2 A}{\partial \xi^2}\right) \quad (12)$$

where the linear term in the top equation is omitted because  $A(0)$  is a minimum, and  $A(0)$  corresponds to the action for the most probable path. Note that the bottom equation has a Gaussian form, giving weights to nearby paths according to

their “distance” from the MPP. If  $\vec{\xi}$  is now a vector, we may write the propagator in equation 10 as

$$G \sim \exp(-A(MPP)) \int_{\xi} \exp(-(\xi \nabla_{\xi})^2 A/2) d\xi. \quad (13)$$

The first term here is the MPP contribution and the integral is over re-parameterized path space. Although integration space is still infinite-dimensional, we can use this expression to estimate some important properties of the radiance distribution by writing out the expansion of  $A$  for some family of nearby paths in terms of relevant parameters while keeping others fixed. In particular, we will be interested in radiation blurring along the path, which can be measured by the *spatial width* of contributing paths.

#### 4.5. Full Propagator

So far we considered different parts of multiply-scattered light propagator  $G_s$  as suited our needs. We now put everything together and write the full approximate expression which will be directly used by the rendering algorithm. To obtain full energy transfer expression, we multiply the contribution of each path by its probability and sum over all paths. Note that from statistical point of view, this is just the expected value of “standard” single path attenuation  $\exp(-\int_0^s c(\vec{\gamma}(s')) ds')$  over the pdf of paths given by equation 11. We can therefore write  $G_s$  as

$$G_s(s, \mathbf{x}, \vec{\omega}, \mathbf{x}', \vec{\omega}') = \int \exp\left(-\int_0^s c(\vec{\gamma}(s')) ds'\right) \text{Cexp}(-A(\vec{\gamma}(s'))) \mathcal{D}x \quad (14)$$

The first factor simply states that light which scatters out of the path (or absorbed along it) is lost as far as the given path is concerned and is analogous to  $G_{noscatter}$  present in the expression for  $G_{single}$ , equation 7. The second factor is due to accounting for probability of the specific sequence of scattering events (and absence of any absorption ones) which forced this particular path and not any other to be chosen and  $C$  is a normalization constant.

An expression for multiply scattered radiance is obtained using general equation 4. Singly scattered (per unit length) radiance  $bL_0$  serves as the source  $Q$  for multiply scattered light, leading to

$$L_s(s, \mathbf{x}, \vec{\omega}) = \int_{\mathbf{x}', \vec{\omega}'} \int \exp\left(-\int_0^s c(\vec{\gamma}(s')) ds'\right) \text{Cexp}(-A(\vec{\gamma}(s'))) \mathcal{D}x b(\mathbf{x}') L_0(\mathbf{x}', \vec{\omega}') d\mathbf{x}' d\vec{\omega}' \quad (15)$$

We will now replace integral over complete path space with that over only paths surrounding the MPP, treating the first term in equation 14 as approximately the same for all such paths and take it out of the path integral. The second term can be expressed by equation 12 and  $\exp(-A(MPP))$  is also taken out of the path integral. The rest of the path integral gives simply the fraction of all paths which satisfy boundary conditions *regardless* of the specific path shape

(but adjusted for absorption to avoid double-counting since this is already included in the first term in equation 14). For simplicity, we consider here only directional light sources and assume that single scattered light distribution is strongly forward peaked, ignoring angular spreading introduced by single scattering event. This means that path fraction is directly given by the multiply-scattered phase function  $P_{MS}$  which gives the probability of light being scattered by a given angle after given number of scattering events. Only the integration over the support of the Gaussian remains:

$$L(s, \mathbf{x}, \vec{\omega}) = \exp\left(-\int_0^s c(MPP_s(s')) ds'\right) \exp(-A_c(MPP_s)) \times P_{MS}(\vec{\omega} - \vec{\omega}') \int_{plane \perp MPP} \text{gauss}(\mathbf{x}', w(s)) b(\mathbf{x}') L_0(\mathbf{x}', \vec{\omega}') d\mathbf{x}' \quad (16)$$

where  $\text{gauss}(\mathbf{x}', w(s))$  is a Gaussian of width  $w(s)$  given by expression 1 applied at position  $\mathbf{x}'$  and  $A_c$  represents the curvature part of the action function. Parameter  $s$  emphasizes that  $MPP_s$  is constructed for given total length  $s$ . Explicit integration over complete  $(\mathbf{x}', \vec{\omega}')$ -space present in the original expression is replaced by integration over its much reduced subset selected by the set of MPPs we consider. This is the key to efficiency of algorithms based on WKB expansion of the path integral. For arbitrary light sources, directional integration will remain and can be performed using similar procedures and expressions for angular blurring [APRN04].

**INHOMOGENEOUS MEDIA.** The final step is to recall that in computer graphics we need a time-independent solution given by integration of expression 16 over all path lengths  $s$ . We also note that since only integrals along paths are used, the result does not depend on whether the medium is homogeneous or not (but, in general, the set of MPPs will). It is useful therefore to write final expressions using the total number of scattering events along the path  $\ell = \int b(\vec{\gamma}(s')) ds'$ :

$$L(\mathbf{x}, \vec{\omega}) = \int_{\ell} d\ell \exp(-(c/b)\ell(MPP_{\ell})) \exp(-A_c(MPP_{\ell})) \times P_{MS}(\ell, \vec{\omega} - \vec{\omega}') \int_{plane \perp MPP} \text{gauss}(\mathbf{x}', w(\ell)) L_0(\mathbf{x}', \vec{\omega}') d\mathbf{x}', \quad (17)$$

Multiply-scattered phase function in WKB approximation is given by Tessendorf and Wasson [TW94] as  $P_{MS}(\ell, \theta) = (1/N)P(\theta/\sqrt{\ell/(1-\exp(-\ell))})$  with new normalization constant  $N$  introduced to ensure that  $\int P_{MS}(\ell, \theta) d\omega = 1$ . We can also re-parameterize our (2/3)-corrected expression 1 for blurring width as

$$w^2 = \frac{\langle \theta^2 \rangle \ell S^2}{24(1 + \langle \theta^2 \rangle (a/b) \ell^2 / 12)}. \quad (18)$$

As will be clear from our algorithm, we will be avoiding constructing curved MPPs which was shown to be quite complicated [PAS03]. Under such conditions, keeping the curvature term in  $A_c(MPP)$  in its exact form is an overkill and we can simplify it to  $\int \frac{\alpha}{\beta} \left| \frac{d\theta}{ds'} \right|^2 ds' \approx \alpha(\Delta\theta)^2/\ell$  where  $\Delta\theta$  is the total angle change along the path. This completes the derivation. In section 5 we will describe the procedure we use to efficiently implement calculation suggested by equation 17.



**Algorithm 1** Preprocessing

---

```

function PreprocessVolume(Volume V)
for each light source  $S_i$ 
     $L_i = \text{ComputeLightAttenuationVolume}(V, S_i)$ 
    BuildVolumePyramid( $L_i$ )

function ComputeLightAttenuationVolume(Volume V, Light S)
for each voxel  $\mathbf{x}_i$  in  $V$ 
    // Compute transmittance  $T$  from voxel  $\mathbf{x}_i$  to
    // light source  $S(\mathbf{x}_{light})$  – see Figure 8.
     $T = \exp(-c * \text{Distance}(\mathbf{x}_i, \mathbf{x}_{light}))$ 
    Store  $L_i = T * \text{LightIntensity}(S)$  in voxel  $\mathbf{x}_i$ 

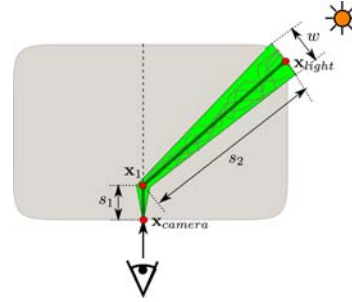
function BuildVolumePyramid(Light Volume LV)
for each level  $l$  (from 1 to  $\log_2(\text{size})$ ) in  $LV$ 
    Resample level  $l - 1$  to half the size
    Apply Gaussian filter with width  $\sigma$  to downsampled volume
    
```

---

**5. Rendering Algorithm**

Our implementation sacrifices the accuracy of computation for efficiency by making a number of further simplifications to deal with already simplified but still rather complex equation 16. We believe that since the main observable consequence of multiple scattering is blurring, it is not crucial to accurately reproduce the result of such a process as long as its most general characteristics (for example, the degree of blurring with the distance to the observer) are captured. A process which results in *loss* of detail should not incur a dramatic computational penalty as is the case with most existing rendering algorithms. The pseudocode for the rendering algorithm is summarized in Algorithm 2.

As a **PRECOMPUTATION STEP**, we first compute the attenuated light volume for every light source (function `ComputeLightAttenuation` in Algorithm 1). From every voxel  $\mathbf{x}_i$  in the volume, we send rays towards a light source ( $\mathbf{x}_{light}$ ), compute attenuation along the ray and store the intensity in the voxel. We currently compute a separate light volume for each light source. The purpose of this precomputation is to build an efficient data structure such that we can quickly lookup how much light is available for redistribution at any point in the volume. We also build a hierarchical representation of this data structure (function `BuildVolumePyramid` in Algorithm 1). This hierarchy allows us to get lighting information at different scales. It is built by applying a set of Gaussian filters with different standard deviations  $\sigma$  to the attenuated light volume. The standard deviation  $\sigma$  can be chosen based on the optical properties of the medium to better cover the expected range of blurring widths  $w$  (from 0 to a user specified maximum width). The filter width for every level is stored with the rest of the data. We use iterative filtering to accelerate the construction process. The blurred light can be stored at full resolution but it is wasteful to do so, because detail is removed by Gaussian blurring. After blurring we also resample the volume with a tent filter to produce a lower level stored in a pyramid. Therefore, the hierarchy

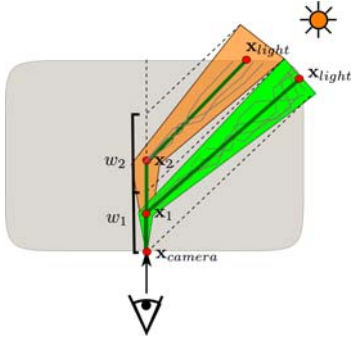


**Figure 8:** To speed up multiple scattering computation we approximate a true curved path with two line segments and compute the blur width that occurred along this path. We then spatially blur the incident radiance and apply proper attenuation to compute the contribution of paths within this cone to multiple scattering.

is essentially a three-dimensional mip-map. There are various data structures that can hold blurred lighting information. Our particular implementation is a variation of the deep shadow map [LV00]. For a reasonably sized volumes ( $128^3$  voxels), creation of the light pyramid takes under 10 seconds.

**DURING RENDERING**, we sum up contributions along light paths. Although the true light path is curved, we instead approximate true path with two linear segments as shown on Figure 8. We refer to computation steps in Algorithm 2. We start by stepping into the media along the viewing ray by some distance  $s_1$  (line 1) to arrive at point  $\mathbf{x}_1$ . We then construct a line segment from  $\mathbf{x}_1$  to the exit point  $\mathbf{x}_{light}$  in the direction of a light source. This segment has length  $s_2$  (line 2) and its corresponding number of scattering events is  $\ell_2$  (line 6). We then get the radiance at point  $\mathbf{x}_1$  from our blurred light volume data structure. The level of the pyramid is chosen based on the blurring width given by equation 1 (line 8) for the total approximate length of the path  $S = s_1 + s_2$ . At each level we have stored the blur width that we used during precomputation. We search through the widths in the light volume pyramid to find the corresponding level. If necessary, we interpolate between two different levels like in standard texture mip-mapping (line 9). The multiple scattering contribution has effectively been reduced to a simple table lookup. Thus far we have determined spatial spreading due to multiple scattering, but have not yet accounted for attenuation. We multiply spatially spread radiance by the necessary attenuation given by remaining terms in equation 16: phase function (line 11), path curvature (line 12), and attenuation (line 13).

To find the next sampling point, we project the blur width onto the viewing ray along the light direction to determine the next sampling point as shown on Figure 9. This avoids counting the same light contribution more than once. Some moderate overlap still occurs but it can be controlled by increasing step size if it presents a problem. The side effect of this process is that the step size gets progressively larger as



**Figure 9:** To choose the next sampling point (red circle), we project a sphere with radius equal to the blur width along the line onto the viewing ray. This process is repeated until the ray exits the volume or entire ray has been covered by contributions due to multiple scattering.

---

**Algorithm 2** *Rendering algorithm summary*


---

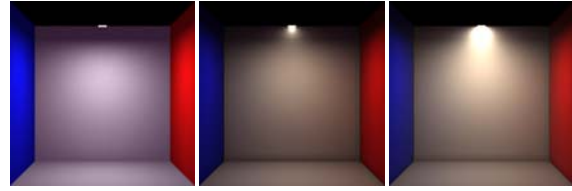
```

function L = MultipleScattering( Light Volume LV )
//  $\alpha = 1/4(1 - g)$ 
//  $P_{MS}(\text{theta}, \ell) = N * P(\text{theta}/\sqrt{\ell/(1 - \exp(-\ell))})$ 
//  $N$  - precomputed  $P_{MS}$  normalization
for each sampling point  $\mathbf{x}_i$ 
    (1)  $s_1 = \text{Distance}(\mathbf{x}_i, \mathbf{x}_{\text{camera}})$ 
    (2)  $s_2 = \text{Distance}(\mathbf{x}_i, \mathbf{x}_{\text{light}})$ 
    (3)  $S = s_1 + s_2$ 
    (4)  $d\ell = \text{Distance}(\mathbf{x}_i, \mathbf{x}_{i-1}) * b$ 
    (5) Compute scattering number  $\ell_1 = \ell_1 + d\ell$ 
    //  $\ell_2$  can be precomputed in ComputeLightAttenuationVolume
    (6) Compute scattering number  $\ell_2$ 
    (7)  $\ell = \ell_1 + \ell_2$ 
    (8) Compute blur width  $w = \sqrt{\frac{(\theta^2)\ell S^2}{24(1 + (\theta^2)(a/b)\ell^2/12)}}$ 
    (9) L = Lookup( $\mathbf{x}_i$ , LV, w)
    // Lookup() returns precomputed attenuated radiance L
    // from the mip-map at point  $\mathbf{x}_i$  and blurring level w
    // Multiple scattering contribution has
    // been reduced to a simple lookup in the table
    (10)  $\text{theta} = \arccos(\vec{\omega} \cdot \vec{\omega}')$  // scattering angle
    (11) P =  $P_{MS}(\text{theta}, \ell)$  // Multiple scattering phase function
    (12) C =  $\alpha * \text{theta}^2 / \ell$  // Path curvature
    // Weight multiple scattering contribution
    (13)  $\text{weight} = \exp(-c/b) \ell) * \exp(-C) * P$ 
    // Add multiple scattering contribution
    (15) Ltotal = Ltotal + L*weight
return Ltotal
    
```

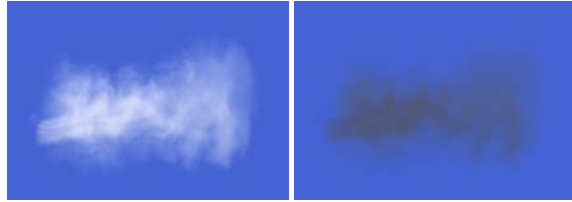
---

we march along the ray since the blur width is monotonically increasing with the distance.

For inhomogeneous media, the computational procedure is similar. The only difference is that due to spatial variation of optical properties, we need to integrate scattering coefficient along the path:  $\ell = \int b(\mathbf{x}_{\text{start}} + s'\vec{\omega})ds'$ . Since  $\ell$  along the path is expensive to compute during rendering, we compute and store  $\ell_2$  (light part of the path) in another data struc-



**Figure 10:** Comparison between no scattering, single scattering and multiple scattering. The room is filled with very thin scattering medium.

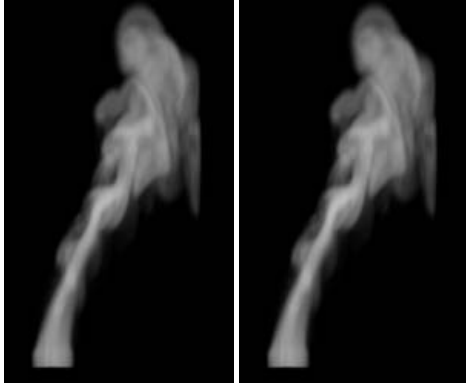


**Figure 11:** Comparison between multiple scattering (left) and single scattering only (right).

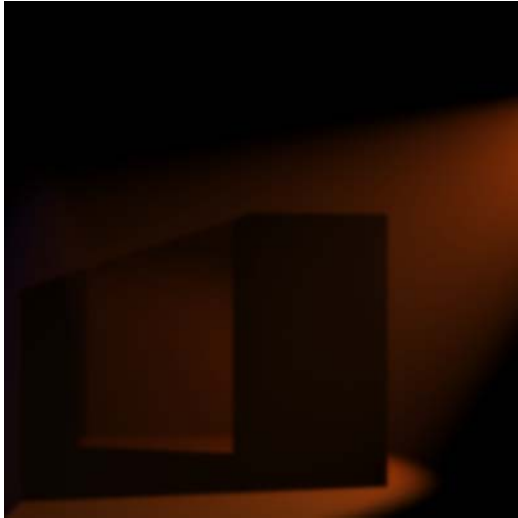
ture during preprocessing. This does not lead to any increase in computation since it is already available during light volume construction phase. As we step further into the volume, we also keep track of  $\ell_1$  for the camera part of the path ( $s_1$ ).

## 6. Results

We show some examples of volumetric media rendered with the algorithm presented in this section. Figure 10 shows an artificial scene with a small area light source that shows basic differences between no scattering (left), single scattering (middle) and multiple scattering (right). The effects of multiple scattering are clearly visible as the glow around the light source. Figure 11 shows a comparison between single scattering and multiple scattering in a cloud (inhomogeneous medium). Single scattering albedo in a cloud is very high and many scattering orders must be considered to get expected appearance. An example in Figure 12 compares a rising smoke (inhomogeneous medium) image rendered with the algorithm presented in this section and traditional Monte Carlo ray marching. The difference between the two images is very subtle (Monte Carlo image is slightly brighter). The Monte Carlo solution required about 3 hours, while the presented algorithm took approximately 10 minutes. Figure 13 shows multiple scattering in homogeneous participating medium illuminated by a spotlight, clearly showing both scattering and shadowing effects. As a final example, we show a more complex example with inhomogeneous participating media in Figure 14. The rendering time for this image (800 by 800 resolution at 9 samples per pixel) was about 30 minutes on a 1 GHz Powerbook G4 with 512 Mb of RAM. It is interesting to note that the single-scattering computation dominated the rendering time. Both components use the precomputed light volume that stored reduced intensity. However, the single scattering computation must resolve very fine features of the participating media and the step size is therefore much smaller than it is in multiple

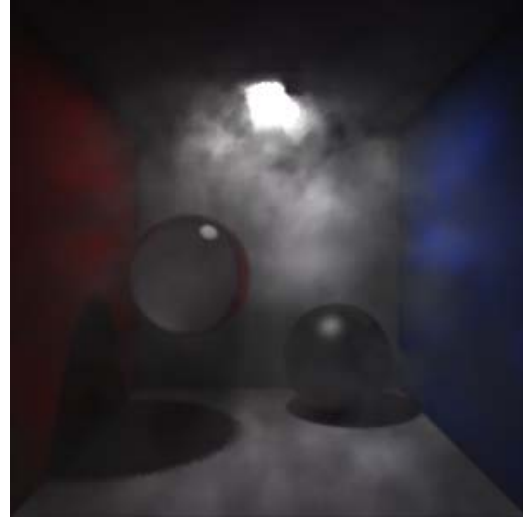


**Figure 12:** Rising smoke column rendered with our algorithm (left) and reference image rendered with Monte Carlo raytracing (right).



**Figure 13:** Multiple scattering in homogeneous participating medium.

scattering computation. The step size in ray marching for multiple scattering component gets progressively larger as we move further away from the camera. The participating medium was stored as density in a volume grid containing  $128^3$  voxels. The scattering coefficients are then computed from density by multiplying it by scattering cross-sections of the volume:  $a(\mathbf{x}) = C_{abs}\rho(\mathbf{x})$  and  $b(\mathbf{x}) = C_{sca}\rho(\mathbf{x})$ , where  $C_{abs}$  is the absorption cross-scattering,  $C_{sca}$  is the scattering cross-section, and  $\rho(\mathbf{x})$  is the density of the medium at point  $\mathbf{x}$ . This requires 8 Mb of storage. The light volume pyramid required 32 Mb, and the precomputed value of  $l_2$  required an additional 8 Mb. While the storage requirements are moderate, they are not insignificant for large volumes. Monte Carlo methods with standard variance reduction techniques (e.g. importance sampling, Russian roulette, etc.) took almost 12 hours to compute, but it required less memory (8 Mb).



**Figure 14:** Cornell box with inhomogeneous participating media. Effects of multiple scattering are clearly visible as the glow around the light source that cannot be captured by single scattering computation only.

## 7. Discussion and Future Work

We have presented a practical technique for rendering effects of multiple scattering in volumetric media. We avoid direct numerical simulation of multiple scattering by taking advantage of spatial spreading. At the heart of our method is the expression for blurring width and the point spread function derived from a path integral analysis of light transport. While the PSF is derived under simple assumptions, effectively calculating the path width for spreading of a beam in a homogeneous medium, it can be applied to inhomogeneous media as well. The expression is reasonably accurate, as shown by comparison to both previous theoretical results, and Monte Carlo simulations. Since we are primarily interested in qualitative effects of multiple scattering and not numerical accuracy, this approximation is reasonable in practice. The main benefit is that the qualitative features of multiple scattering, such as spatial blurring of incident radiance, are computed more efficiently than previous methods. While the mathematics of the derivation are somewhat involved, the practical implementation is relatively simple. Further computational savings are obtained by using a progressively larger step size in ray marching since the blur size monotonically increases with distance. The net result is a practical algorithm that captures the characteristic effects of multiple scattering in a fraction of the time required by Monte Carlo simulation methods.

There are disadvantages to our approach, that form interesting directions of future work. From a theoretical perspective, we consider only the overall statistics of the phase function, not its particular shape. This and other approximations result in some quantitative differences with respect to Monte Carlo simulations. Furthermore, the computation of multiply scattered light attenuation in our algorithm is not

straightforward, and is worth investigating further. From a practical perspective, the main disadvantage of the current implementation is that we precompute the light attenuation volume for each light source. While this is simply an optimization or time-space tradeoff, it can become quite expensive for a large number of light sources. We would benefit from a more efficient data structure for the lighting volume, that stored the incident lighting distribution at each point in a compact representation like spherical harmonics.

In summary, multiple scattering effects like spatial blurring give rise to some of the most distinctive features in volumes and participating media. Previous approaches have involved very expensive Monte Carlo simulations. We have shown that the same qualitative effects can be obtained in a ray marching algorithm using a point spread function that blurs incident illumination.

### Acknowledgements

This work was supported in part by grants from the National Science Foundation (CCF #0305322 on Real-Time Visualization and Rendering of Complex Scenes) and Intel Corporation (Real-Time Interaction and Rendering with Complex Illumination and Materials).

### References

[APRN04] ASHIKHMİN M., PREMOŽE S., RAMAMOORTHY R., NAYAR S.: *Blurring of Light due to Multiple Scattering by the Medium: a Path Integral Approach*. Technical report CUCS-017-04, Columbia University, April 2004. 3, 4, 8

[BG70] BELL G. I., GLASSTONE S.: *Nuclear Reactor Theory*. Van Nostrand Reinhold, New York, 1970. 6

[Boh87a] BOHREN C. F.: *Clouds in a glass of beer*. John Wiley and Sons, 1987. 1

[Boh87b] BOHREN C. F.: Multiple scattering of light and some of its observable consequences. *American Journal of Physics* 55, 6 (June 1987), 524–533. 1

[BSS93] BLASI P., SAEC B. L., SCHLICK C.: A rendering algorithm for discrete volume density objects. *Computer Graphics Forum* 12, 3 (1993), 201–210. 1

[Cha60] CHANDRASEKAR S.: *Radiative Transfer*. Dover, New York, 1960. 1, 6

[DEJ\*99] DORSEY J., EDELMAN A., JENSEN H., LEGAKIS J., PEDERSEN H.: Modeling and rendering of weathered stone. In *Proceedings of SIGGRAPH* (1999), pp. 225–234. 2

[EP90] EBERT D., PARENT R.: Rendering and animation of gaseous phenomena by combining fast volume and scanline a-buffer techniques. In *Proceedings of SIGGRAPH* (1990), pp. 357–366. 2

[FH65] FEYNMAN R. P., HIBBS A. R.: *Quantum mechanics and path integrals*. McGraw-Hill Higher Education, 1965. 7

[Fox87] FOX C.: *An Introduction to the Calculus of Variations*. Dover, New York, 1987. 7

[Gor94] GORDON H. R.: Equivalence of the point and beam spread functions of scattering media: A formal demonstration. *Applied Optics* 33, 6 (February 1994), 1120–1122. 3

[HK93] HANRAHAN P., KRUEGER W.: Reflection from layered surfaces due to subsurface scattering. In *Proceedings of SIGGRAPH* (August 1993), pp. 165–174. 2

[Ish78] ISHIMARU A.: *Wave Propagation and Scattering in Random Media. Volume 1: Single Scattering and Transport Theory*. Academic Press, 1978. 1, 6

[Ish79] ISHIMARU A.: Pulse propagation, scattering, and diffusion in scatterers and turbulence. *Radio Science* 14 (1979), 269–276. 3

[JB02] JENSEN H. W., BUHLER J.: A rapid hierarchical rendering technique for translucent materials. *ACM Transactions on Graphics* 21, 3 (July 2002), 576–581. 2

[JC98] JENSEN H. W., CHRISTENSEN P. H.: Efficient simulation of light transport in scenes with participating media using photon maps. In *Proceedings of SIGGRAPH* (July 1998), pp. 311–320. 1

[JMLH01] JENSEN H., MARSCHNER S., LEVOY M., HANRAHAN P.: A practical model for subsurface light transport. In *Proceedings of SIGGRAPH* (2001), pp. 511–518. 2

[KH84] KAJIYA J., HERZEN B.: Ray tracing volume densities. In *Proceedings of SIGGRAPH* (1984), pp. 165–174. 2

[KvD01] KOENDERINK J. J., VAN DOORN A. J.: Shading in the case of translucent objects. In *Proc. SPIE Vol. 4299 Human Vision and Electronic Imaging VI* (2001), Rogowitz B. E., Pappas T. N., (Eds.), pp. 312–320. 2

[LBC94] LANGUENOU E., BOUATOUCH K., CHELLE M.: Global illumination in presence of participating media with general properties. In *Proceedings of Eurographics Rendering Workshop* (1994), pp. 69–85. 1

[LCH95] LUTOMIRSKI R. F., CIERVO A. P., HALL G. J.: Moments of multiple scattering. *Applied Optics* 34, 30 (October 1995), 7125–7136. 3

[LGB\*02] LENSCH H. P. A., GOESELE M., BEKAERT P., KAUTZ J., MAGNOR M. A., LANG J., SEIDEL H. P.: Interactive rendering of translucent objects. In *Proceedings of Pacific Graphics* (2002). 2

[LRT82] LANGOUCHE F., ROEKAERTS D., TIRAPEGUI E.: *Functional Integration and Semiclassical Expansions*. D. Reidel Publishing Company, 1982. 7

[LV00] LOKOVIC T., VEACH E.: Deep shadow maps. In *Proceedings of ACM SIGGRAPH* (July 2000), pp. 385–392. 9

[Max86] MAX N.: Atmospheric illumination and shadows. In *Proceedings of SIGGRAPH* (1986), pp. 117–124. 2

[Max94] MAX N.: Efficient light propagation for multiple anisotropic volume scattering. In *Proceedings of Eurographics Rendering Workshop* (1994), pp. 87–104. 1

[MCH87] MCLEAN J. W., CRAWFORD D. R., HINDMAN C. L.: Limits of small angle scattering theory. *Applied Optics* 26 (1987), 2053–2054. 3

[MFW98] MCLEAN J. W., FREEMAN J. D., WALKER R. E.: Beam spread function with time dispersion. *Applied Optics-LP* 37, 21 (July 1998), 4701–4711. 3, 4

[MV91] MCLEAN J. W., VOSS K. J.: Point spread function in ocean water: Comparison between theory and experiment. *Applied Optics* 30, 15 (May 1991), 2027–2030. 3

[NDN96] NISHITA T., DOBASHI Y., NAKAMAE E.: Display of clouds taking into account multiple anisotropic scattering and sky light. In *Proceedings of SIGGRAPH* (1996), pp. 379–386. 2

[NKON90] NAKAMAE E., KANEDA K., OKAMOTO T., NISHITA T.: A lighting model aiming at drive simulators. In *Proceedings of SIGGRAPH* (1990), pp. 395–404. 2

[NN03] NARASIMHAN S., NAYAR S.: Shedding light on the weather. In *Computer Vision and Pattern Recognition* (2003). 2

[PAS03] PREMOŽE S., ASHIKHMİN M., SHIRLEY P.: Path integration for light transport in volumetric materials. In *Proceedings of the Eurographics Symposium on Rendering* (2003). 1, 7, 8

[PH00] PHARR M., HANRAHAN P.: Monte Carlo evaluation of non-linear scattering equations for subsurface reflection. In *Proceedings of SIGGRAPH* (2000), pp. 75–84. 2

[PM93] PATTANAIK S., MUDUR S.: Computation of global illumination in a participating medium by Monte Carlo simulation. *Journal of Visualization and Computer Animation* 4, 3 (1993), 133–152. 1

[PPS97] PEREZ F., PUEYO Z., SILLION F.: Global illumination techniques for the simulation of participating media. In *Proceedings of Eurographics Rendering Workshop* (1997), pp. 309–320. 2

[RT87] RUSHMEIER H., TORRANCE K.: The zonal method for calculating light intensities in the presence of a participating medium. In *Proceedings of SIGGRAPH* (1987), pp. 293–302. 1

[Sak90] SAKAS G.: Fast rendering of arbitrary distributed volume densities. In *Eurographics 90* (1990), pp. 519–530. 2

[Sta95] STAM J.: Multiple scattering as a diffusion process. In *Proceedings of Eurographics Rendering Workshop* (1995), pp. 41–50. 2

[Sto78] STOTTS L. B.: Closed form expression for optical pulse broadening in multiple-scattering media. *Applied Optics* 17 (1978), 504–505. 2, 3

[Tes87] TESSENDORF J.: Radiative transfer as a sum over paths. *Phys. Rev. A* 35 (1987), 872–878. 1, 3, 4

[Tes89] TESSENDORF J.: Time-dependent radiative transfer and pulse evolution. *J. Opt. Soc. Am. A*, 6 (1989), 280–297. 7

[Tes91] TESSENDORF J.: The Underwater Solar Light Field: Analytical Model from a WKB Evaluation. In *SPIE Underwater Imaging, Photography, and Visibility* (1991), Spinnrad R. W., (Ed.), vol. 1537, pp. 10–20. 7

[Tes92] TESSENDORF J.: Measures of temporal pulse stretching. In *SPIE Ocean Optics XI* (1992), Gilbert G. D., (Ed.), vol. 1750, pp. 407–418. 7

[TW94] TESSENDORF J., WASSON D.: Impact of multiple scattering on simulated infrared cloud scene images. In *Proceedings SPIE. Characterization and propagation of sources and backgrounds* (1994), vol. 2223, pp. 462–473. 8

[WM99] WALKER R. E., MCLEAN J. W.: Lidar equations for turbid media with pulse stretching. *Applied Optics* 38, 12 (April 1999), 2384–2397. 3