Efficient Simulation of Light Transport in Scenes with Participating Media using Photon Maps

Henrik Wann Jensen *Per H. Christensen*

mental images*

Abstract

This paper presents a new method for computing global illumination in scenes with participating media. The method is based on bidirectional Monte Carlo ray tracing and uses photon maps to increase efficiency and reduce noise. We remove previous restrictions limiting the photon map method to surfaces by introducing a volume photon map containing photons in participating media. We also derive a new radiance estimate for photons in the volume photon map. The method is fast and simple, but also general enough to handle nonhomogeneous media and anisotropic scattering. It can efficiently simulate effects such as multiple volume scattering, color bleeding between volumes and surfaces, and volume caustics (light reflected from or transmitted through specular surfaces and then scattered by a medium). The photon map is decoupled from the geometric representation of the scene, making the method capable of simulating global illumination in scenes containing complex objects. These objects do not need to be tessellated; they can be instanced, or even represented by an implicit function. Since the method is based on a bidirectional simulation, it automatically adapts to illumination and view. Furthermore, because the use of photon maps reduces noise and aliasing, the method is suitable for rendering of animations.

CR Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism — *color, shading, shadowing, and texture; raytracing*; I.6.8 [**Simulation and Modeling**]: Types of Simulation — *Monte Carlo.*

Additional keywords: participating media, light transport, global illumination, multiple scattering, volume caustics, nonhomogeneous media, anisotropic scattering, rendering, photo-realism, photon tracing, photon map, ray marching.

*mental images GmbH & Co. KG, Fasanenstraße 81, D-10623 Berlin, Germany. E-mail: {henrik,per}@mental.com.

1 Introduction

A physically correct simulation of light transport in participating media is necessary to render realistic images of fog, clouds, dusty air, smoke, fire, silty water, etc. Such images are important for applications dealing with the visibility of objects (for example road signs in fog or fire exit signs in smoke-filled rooms), for visualization of underwater scenes, and for high-quality visual special effects. See [30] for a more detailed discussion of the application areas.

The first methods taking participating media into account only considered direct illumination (single scattering) [5, 17, 21]. This approximation is applicable to optically thin media such as the atmosphere (where secondary scattering is less important), but fails to capture important effects in optically thick media such as clouds. Single scattering in water has been simulated in [38] where refracted light from a water surface is approximated using polygonal illumination volumes.

Multiple scattering has been simulated with finite element, point collocation, and Monte Carlo methods.

Finite element methods divide participating media into discrete volume elements and simulate the exchange of light between these volume elements. Point collocation methods choose a set of points in the volume and compute the illumination at these points. Isotropic scattering can be simulated by representing the radiance from each volume element or point by one value [29, 31]. To simulate anisotropic scattering, directional distributions are needed at each volume element or point. Some methods use spherical harmonics [3, 16, 33], while others divide the sphere of directions into a finite number of solid angles [20, 22, 25]. Finite element and point collocation methods are most useful when simulating softly illuminated isotropic media. The simulation of sharp illumination edges requires many volume elements, and directional distributions require complex representations of the illumination. In both cases a huge amount of memory is required.

Monte Carlo methods are more versatile and require less memory. However, pure Monte Carlo path tracing (including bidirectional path tracing) uses significant amounts of computation time to render images without noise, particularly in the presence of participating media [19]. One way to speed up rendering is to use particle tracing combined with a discretization of the scene and the volume for representing the illumination [26]. However, this imposes the same limitations on the algorithm as is the case with finite element methods.

A hybrid method [29] uses a finite element preprocess to increase the efficiency of Monte Carlo rendering. This makes the method less general than pure Monte Carlo techniques and imposes limitations on the supported types of scattering. The finite element preprocess increases memory usage and

makes it necessary to discretize the volumes. Even though this is more efficient than pure Monte Carlo rendering, it still uses considerable amounts of computation time to render images without noise.

A detailed overview of previous solution methods and additional references can be found in [27]. None of the existing methods can efficiently simulate effects such as volume caustics where light is focused in a medium. Furthermore, all of the methods except the Monte Carlo path tracing methods are based on a tight coupling of geometry and volumes with the simulation of light transport. This makes them unsuitable for complex scenes.

In this paper we present an extension of the photon map method for global illumination [13, 14] to scenes with participating media. We introduce a *volume photon map* containing photons (particles with energy) within participating media and derive a formula for estimating radiance in a participating medium based on these photons. We also present techniques for using this radiance estimate to efficiently render effects such as multiple volume scattering, color bleeding between surfaces and volumes, and volume caustics. Volume caustics are formed when light reflected from or transmitted through specular surfaces is scattered in a medium. An example of volume caustics is the beams of light in a silty underwater environment [7]. The use of photons makes it easy to simulate homogeneous as well as nonhomogeneous media, and the fact that we include the incoming direction with each photon allows us to directly render media with anisotropic scattering. The photon map is decoupled from the geometry and the volumes, and it is capable of handling any type of representation that can be ray traced. This makes it possible to render for example an implicit function representing a cloud without using any geometry. Another notable aspect of the method is the automatic adaptation to illumination and view: photons are only stored in the illuminated parts of the scene and photon map lookups are only done where required for rendering. The photon map approach provides the same flexibility as pure Monte Carlo path tracing, but is significantly more efficient. It has very little noise and aliasing, making it suitable for rendering animations with participating media.

The remainder of the paper is organized as follows: In section 2, we give an overview of light transport in participating media. The photon map method for surfaces is summarized in section 3, and in section 4 we describe our new extension for handling participating media. In section 5, we present our results, and in section 6 we discuss the method and ideas for future work. Lastly, in section 7 we close with a conclusion.

2 Light transport in participating media

Light transported through a participating medium is affected by emission, in-scattering, absorption, and out-scattering [32]. Taking these four terms into account, the change in (field) radiance L at a point x in direction $\vec{\omega}$ is

$$\begin{aligned}\frac{\partial L(x,\vec{\omega})}{\partial x} &= \alpha(x)\,L_e(x,\vec{\omega}) + \sigma(x)\,L_i(x,\vec{\omega}) \\ &\quad - \alpha(x)\,L(x,\vec{\omega}) - \sigma(x)\,L(x,\vec{\omega})\,, \qquad (1)\end{aligned}$$

where L_e is the emitted radiance, L_i is the in-scattered radiance, α is the absorption coefficient, and σ is the scattering coefficient. This equation holds for each wavelength separately, but we will ignore wavelength here. If the scattering and absorption coefficients are constant throughout the medium, we call the medium homogeneous or uniform.

The last two terms of equation (1), for absorption and out-scattering, can be combined to an extinction term $\kappa(x)\,L(x,\vec{\omega})$, where the extinction coefficient κ is defined as

$$\kappa(x) = \alpha(x) + \sigma(x)\,. \qquad (2)$$

The in-scattered radiance L_i depends on radiance L from all directions $\vec{\omega}'$ over the sphere Ω. The in-scattered radiance is

$$L_i(x,\vec{\omega}) = \int_\Omega f(x,\vec{\omega}',\vec{\omega})\,L(x,\vec{\omega}')\,d\omega'\,. \qquad (3)$$

While the scattering coefficient σ determines how much of the incident light is scattered at point x, the normalized phase function f determines how much of the scattered light at x is scattered in direction $\vec{\omega}$. The product $\sigma(x)\,f(x,\vec{\omega}',\vec{\omega})$ expresses the fraction of the differential irradiance from direction $\vec{\omega}'$ to point x that is scattered (as radiance) in direction $\vec{\omega}$. The normalized phase function integrates to 1 over the sphere. If the phase function is constant ($1/4\pi$), we call the scattering isotropic or diffuse; if not, we call the scattering anisotropic or directional.

Inserting equations (2) and (3) into equation (1) we get the integro-differential equation

$$\begin{aligned}\frac{\partial L(x,\vec{\omega})}{\partial x} &= \alpha(x)\,L_e(x,\vec{\omega}) \\ &\quad + \sigma(x)\int_\Omega f(x,\vec{\omega}',\vec{\omega})\,L(x,\vec{\omega}')\,d\omega' \\ &\quad - \kappa(x)\,L(x,\vec{\omega})\,.\end{aligned}$$

Integrating both sides of the equation along a straight path from x_0 to x (in direction $\vec{\omega}$) gives the following integral equation [32]:

$$\begin{aligned}L(x,\vec{\omega}) &= \int_{x_0}^{x}\tau(x',x)\,\alpha(x')\,L_e(x',\vec{\omega})\,dx' \\ &\quad + \int_{x_0}^{x}\tau(x',x)\,\sigma(x')\int_\Omega f(x',\vec{\omega}',\vec{\omega})\,L(x',\vec{\omega}')\,d\omega'\,dx' \\ &\quad + \tau(x_0,x)\,L(x_0,\vec{\omega})\,, \qquad (4)\end{aligned}$$

where $\tau(x',x)$ is the transmittance along the line segment from x' to x,

$$\tau(x',x) = e^{-\int_{x'}^{x}\kappa(\xi)\,d\xi}\,,$$

and similar for $\tau(x_0,x)$.

Equation (4) can be simplified if the medium is homogeneous or the scattering is isotropic, but not in the general case which we consider here.

3 The photon map method for surfaces

The photon map method was originally developed for global illumination simulation in scenes without participating media [13]. It is a two-pass method where the first pass is the construction of two view-independent photon maps and the second pass is optimized rendering using these photon maps.

The photon maps are generated by emitting photons from the light sources and tracing these through the scene using photon tracing. Every time a photon hits a nonspecular surface it is stored in the photon map(s). The result is a large number of photons distributed on the surfaces within the scene.

The first photon map is a high quality caustics photon map which consists of all photons that have been traced from the light source through a number of specular reflections or transmissions before intersecting a diffuse surface. This path can be expressed as LS^+D using the notation introduced in [12]. The second photon map is a global photon map which is less accurate than the caustics photon map. It contains all photons representing indirect illumination on a nonspecular surface, $L(S|D)^+D$.

A key aspect of the method is the use of a balanced kd-tree [2, 15] to efficiently and compactly handle these photons. This structure makes it possible to represent each photon using only 20 bytes [13].

The rendering pass is a distribution ray tracer optimized in several ways using the two photon maps. The caustics photon map is used to render caustics directly. This is a significant optimization since caustics are nearly impossible to compute using ray tracing from the eye [37]. The global photon map is used to limit the number of reflections traced by the distribution ray tracer and to sample indirect illumination more efficiently.

The photon map can be used to estimate radiance at any given surface position x using the information about the flux $\Delta\Phi_p$ carried by each photon p in direction $\vec{\omega}'_p$. By locating the n photons with the shortest distance to x it is possible to estimate the photon density around x. The estimate of the reflected radiance from a surface is then computed as [13]

$$L_r(x,\vec{\omega}) \approx \sum_{p=1}^{n} f_r(x,\vec{\omega}'_p,\vec{\omega}) \frac{\Delta\Phi_p(x,\vec{\omega}'_p)}{\pi r^2}, \qquad (5)$$

where f_r is the bidirectional reflectance distribution function and r is the distance to the nth nearest photon. This approach can be seen as expanding a sphere centered at x until it contains n photons. Then r is the radius of the sphere and the denominator in the formula is the projected area of this sphere as illustrated in figure 1(a).

The photon map method has the full flexibility of bidirectional Monte Carlo path tracing [18, 34]: all possible light paths can be simulated, including caustics. Tessellating the geometry is not necessary, and the aliasing problems in methods using texture maps [1, 12] or simple geometric primitives [36] to represent the illumination are significantly reduced. Furthermore, since the photon map radiance estimate is inherently a low-pass filter, the high-frequency, grainy noise present in pure Monte Carlo methods disappears. At the same time, the Monte Carlo method's ability to compute precise illumination details is maintained since the photon density is high where the illumination is intense. The photon map radiance estimate is valid as long as the surface is locally flat, and the estimate converges to the correct value as more photons are used.

The Metropolis light transport algorithm [35], a recent improvement over bidirectional path tracing, focuses its computations on the light paths that contribute most to the rendered image. This gives improved performance for scenes containing for example underwater caustics seen through a wavy water surface. The photon map method also handles such scenes efficiently: in the first pass, photons are refracted by the water surface, hit the diffuse pool bottom, and are stored in the caustics photon map; in the second pass, rays from the eye are refracted to the appropriate position on the pool bottom, and photon map lookups return the intensities of the caustics. For soft indirect illumination, the Metropolis algorithm tends to use the same amount of computation time as brute force bidirectional Monte Carlo path tracing. The illumination in participating media is mostly soft due to the continuous scattering taking place everywhere in the medium. This makes a possible extension of the Metropolis algorithm less attractive for this application.

4 Extending the photon map to participating media

Simulating light transport in participating media requires extension of several aspects of the photon map method. Photons can be scattered and absorbed by the media. To efficiently render the medium, it is necessary to store information about these scattering events. For this purpose several strategies have been considered.

4.1 Algorithmic considerations

One might consider discretizing the media into a finite set of volume elements for which the volume radiance is computed. This strategy would, however, result in unnecessary restrictions and aliasing problems. Another alternative would be to store photons on the surfaces of the media. In this way, the surface radiance estimate in equation (5) could be used to extract information about the radiance leaving or entering a medium. The drawback of this approach is that it would fail to efficiently capture illumination details such as volume caustics inside the medium. Also, it would not work in situations where the viewpoint is within the medium. A better strategy is to store the photons explicitly in the volume. Since the photon map is based on a three-dimensional data structure, this can be done without changing the underlying algorithms. Storing the photons explicitly in the volume has several advantages: the photons can be concentrated where necessary to represent intense illumination, the media do not have to be discretized (it is possible to directly render implicitly defined volumes), and anisotropic scattering can be handled by storing the incoming direction of each photon.

The photons in volumes and on surfaces must be separated when the photon map is queried for information about the incoming flux. This is necessary because the relationship between the density of the photons and the illumination is different on surfaces and in volumes (as described in section 4.3). The separation of the two types of photons could be done by tagging the volume photons and storing them in the global photon map. This would, however, result in more computationally expensive lookups due to the larger size of the kd-tree and the need to separate the photons. Instead we introduce a separate volume photon map for the photons that are scattered in participating media. The volume photon map is used to compute the illumination inside a participating medium, and the global photon map is used — as before — to compute the illumination on surfaces.

The direct illumination of participating media is easy to compute using traditional ray tracing techniques. Therefore we only use the photon map to represent indirect illumination. That is, we only store photons that have been reflected or transmitted by surfaces before interacting with the media, and photons that have been scattered at least once in the media. This is a tradeoff between memory and speed since the photon map is perfectly capable of computing the direct illumination at the expense of using more photons to obtain the desired accuracy.

4.2 Photon tracing

In the first pass, the photon maps are built using photon tracing. A photon traced within a participating medium can either pass unaffected through the medium, or it can interact with it (be scattered or absorbed). If the photon interacts with the medium, and does not come directly from a light source, it is stored in the photon map. The cumulative probability density function, $F(x)$, expressing the probability of a photon interacting with a participating medium at position x is

$$F(x) \;=\; 1-\tau(x_s,x) \;=\; 1-e^{-\int_{x_s}^{x}\kappa(\xi)\,d\xi},$$

where x_s is the point at which the photon enters the medium. The transmittance $\tau(x_s,x)$ is computed using ray marching.

If a photon interacts with the medium, Russian roulette decides whether the photon is scattered or absorbed. The probability of a photon being scattered is given by the scattering albedo $\sigma(x)/\kappa(x)$. The new direction of a scattered photon is chosen using importance sampling based on the phase function at x.

4.3 Estimating radiance

The photons stored in a volume photon map can be used to compute an estimate of the in-scattered radiance $L_i(x,\vec{\omega})$. This illumination is determined by the photons closest to x, as is the case with the technique used for surfaces. However, we cannot directly use the radiance estimate for surfaces since it is based on the definition of radiance for surfaces. Instead, we can utilize the relationship between scattered flux Φ and radiance L in a participating medium [32],

$$L(x,\vec{\omega}) \;=\; \frac{d^2\Phi(x,\vec{\omega})}{\sigma(x)\,d\omega\,dV}\,.$$

By combining this relation with equation (3) we get

$$\begin{aligned}
L_i(x,\vec{\omega}) \;&=\; \int_\Omega f(x,\vec{\omega}',\vec{\omega})\,L(x,\vec{\omega}')\,d\omega' \\
&=\; \int_\Omega f(x,\vec{\omega}',\vec{\omega})\,\frac{d^2\Phi(x,\vec{\omega}')}{\sigma(x)\,d\omega'\,dV}\,d\omega' \\
&=\; \frac{1}{\sigma(x)}\int_\Omega f(x,\vec{\omega}',\vec{\omega})\,\frac{d^2\Phi(x,\vec{\omega}')}{dV} \\
&\approx\; \frac{1}{\sigma(x)}\sum_{p=1}^{n} f(x,\vec{\omega}_p',\vec{\omega})\,\frac{\Delta\Phi_p(x,\vec{\omega}_p')}{\frac{4}{3}\pi r^3}\,, \qquad (6)
\end{aligned}$$

where Φ is the in-scattered flux and dV is the differential volume containing the photons. This differential volume is approximated by $\frac{4}{3}\pi r^3$, corresponding to the smallest sphere containing the n nearest photons (this is effectively a nth nearest neighbor density estimate). Notice the relationship between this formula and formula (5) for estimating radiance on surfaces. The main difference is that the density on a surface is computed using the projected area, whereas the density in a medium is computed using the full volume — as shown in figure 1.

Using formula (6) we can compute a radiance estimate at any given point inside a participating medium. Since we know the incoming direction $\vec{\omega}_p'$ of each photon, we can handle anisotropic phase functions as well as isotropic phase functions.

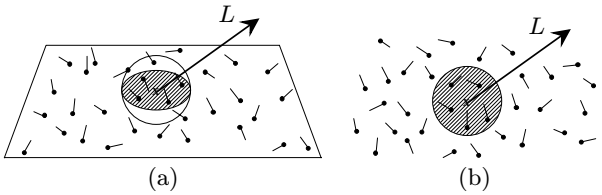


Figure 1: Radiance estimate for (a) surfaces and (b) volumes.

As already mentioned, we split the computation of in-scattered radiance into two parts: direct $L_{i,d}$ and indirect $L_{i,i}$. The direct part $L_{i,d}$ is computed by sampling of the light sources using ray marching. The indirect part $L_{i,i}$ is estimated using the volume radiance estimate from equation (6),

$$L_{i,i}(x,\vec{\omega}) \;\approx\; \frac{1}{\sigma(x)}\sum_{p=1}^{n} f(x,\vec{\omega}_p',\vec{\omega})\,\frac{\Delta\Phi_{p,i}(x,\vec{\omega}_p')}{\frac{4}{3}\pi r^3}\,,$$

where $\Delta\Phi_{p,i}$ is the flux carried by the photons that correspond to the indirect illumination.

Since we store and use all photons that interact with a medium (even photons that are subsequently absorbed), $L_{i,i}$ has to be multiplied by the scattering albedo $\sigma(x)/\kappa(x)$. This leads to the following formula for computing the in-scattered radiance:

$$L_i(x,\vec{\omega}) \;=\; L_{i,d}(x,\vec{\omega}) + \frac{\sigma(x)}{\kappa(x)}\,L_{i,i}(x,\vec{\omega})\,.$$

4.4 Rendering

In the second pass, the image is rendered using the photon maps. We render the surfaces in the scene using the same approach as described in [13]. To incorporate participating media into this method we need to consider those rays that pass through a medium.

The radiance of a ray traversing a participating medium is computed with an adaptive ray marching algorithm [9] that iteratively computes radiance at points along the ray. In each step, the radiance from the previous point is attenuated and the contribution from emission and in-scattering within the step is added, corresponding to equation (4). The emitted and in-scattered radiance is approximated as being constant within each step. With this approximation, the radiance at points x_k along a ray in direction $\vec{\omega}$ is computed iteratively as

$$\begin{aligned}
L(x_k,\vec{\omega}) \;&=\; \alpha(x_k)\,L_e(x_k,\vec{\omega})\,\Delta x_k \\
&+\; \sigma(x_k)\,L_i(x_k,\vec{\omega})\,\Delta x_k \\
&+\; e^{-\kappa(x_k)\,\Delta x_k}\,L(x_{k-1},\vec{\omega})\,,
\end{aligned}$$

where $\Delta x_k = |x_k - x_{k-1}|$ is the step size and x_0 is the nearest intersection point of the ray with a surface (or the back side of the volume).

The step size is recursively halved if the currently computed radiance differs too much from the radiance in the previous point. This adaptation to illumination makes the ray marcher capable of rendering media efficiently while still capturing small illumination details. The size of each step is also varied using jittering to eliminate the aliasing problems that can occur with a fixed step size.

5 Results

We have implemented the presented method as a part of mental ray, a commercial rendering program that supports parallel ray tracing and has a flexible shader interface [6]. We use a parallelized photon map algorithm as described in [14], where one photon map is shared between all processors. The implementation supports nonhomogeneous media and anisotropic scattering. We use Schlick's two-lobed phase function [4] to approximate Mie scattering [23]. Other phase functions such as the Rayleigh or Henyey-Greenstein phase functions [10] could be used as well. We have not implemented emitting media, but this could easily be added.

The images in figures 2–5 have been rendered on an HP S-class computer with sixteen 180 MHz PA-8000 processors, while the images in figures 6–8 have been rendered on an SGI Origin 2000 computer with sixteen 195 MHz MIPS R10000 processors. Each image is 1024 pixels wide and rendered using up to 16 samples per pixel.

Figure 2 shows a volume caustic created as light is focused by a glass sphere in an isotropic (diffuse) homogeneous medium. The light is emitted from a point light source (a big bright halo around the light source is visible in the upper left corner of the image). The volume photon map contains 100,000 photons, and 500 photons were used in the radiance estimate. Building the volume photon map took 5 seconds, and rendering the image took 41 seconds.

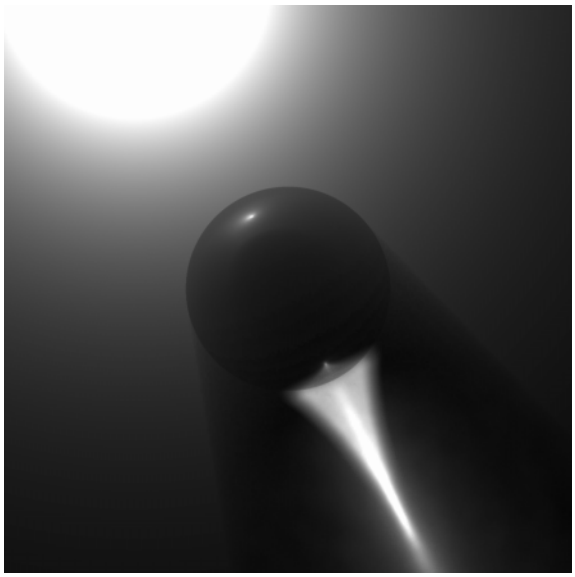


Figure 2: A volume caustic.

Figure 3 illustrates the effect of multiple scattering in a nonhomogeneous, anisotropic medium. The cloud is modeled using an implicitly defined function consisting of 10 blobs combined with turbulent noise [8]. The anisotropic scattering is modeled using an approximation of hazy Mie scattering. In figure 3(a) only single scattering has been simulated (no photons), whereas figure 3(b) demonstrates multiple scattering (using just 10,000 photons in the photon map and 60 photons in the radiance estimate). Despite the low number of photons, the cloud in 3(b) is brighter and looks more realistic. Figure 3(a) took 61 seconds to render. The photon map for image 3(b) was generated in 8 seconds and the image was rendered in 92 seconds.

Our next test case, a variation of the "Cornell box", is

(a)

(b)

Figure 3: The cloud is an anisotropic, nonhomogeneous participating medium: (a) direct illumination (single scattering), (b) global illumination (multiple scattering).

shown in figure 4. The box has diffuse reflection on the walls, floor, and ceiling. It contains a glass sphere, a mirror sphere, and a participating medium. To make it easier to see the participating medium, the box has no front or back wall. The geometry is represented using approximately 25,000 triangles, and the scene is illuminated by a square-shaped area light source.

Figure 4(a) shows direct illumination on surfaces and in the volume (an isotropic, homogeneous medium). It took 1 minute 49 seconds to render this image. Because of the area light source, the shadow of the sphere is sharp close to the sphere and gets blurry further away.

Figure 4(b) has full global illumination on the surfaces and in the isotropic, homogeneous participating medium. Notice that the caustic created by the glass sphere becomes visible in the fog, that the halo around the light source gets slightly bigger because of multiple volume scattering, and that the fog gets red and blue tints near the colored walls of the box. The volume caustic under the glass sphere is wider than in figure 2 since the light source is an area light source. The image was computed using a total of 200,000 photons in the photon maps, out of which 65,000 were in the volume photon map. The radiance estimate in the volume used 100 photons. It took 4 seconds to generate the photon maps, and 3 minutes 32 seconds to render the image.

In figure 4(c) the medium has anisotropic scattering. The scattering is mainly forward; it is modeled using Schlick's scattering model with a single lobe of eccentricity $k = 0.8$. As a result of the forward scattering, the halo around the light source changes shape. Also, the volume caustic gets dimmer since most of the light in it gets scattered in a direction perpendicular to the eye. Photon map building took 4 seconds and rendering took 4 minutes 3 seconds.

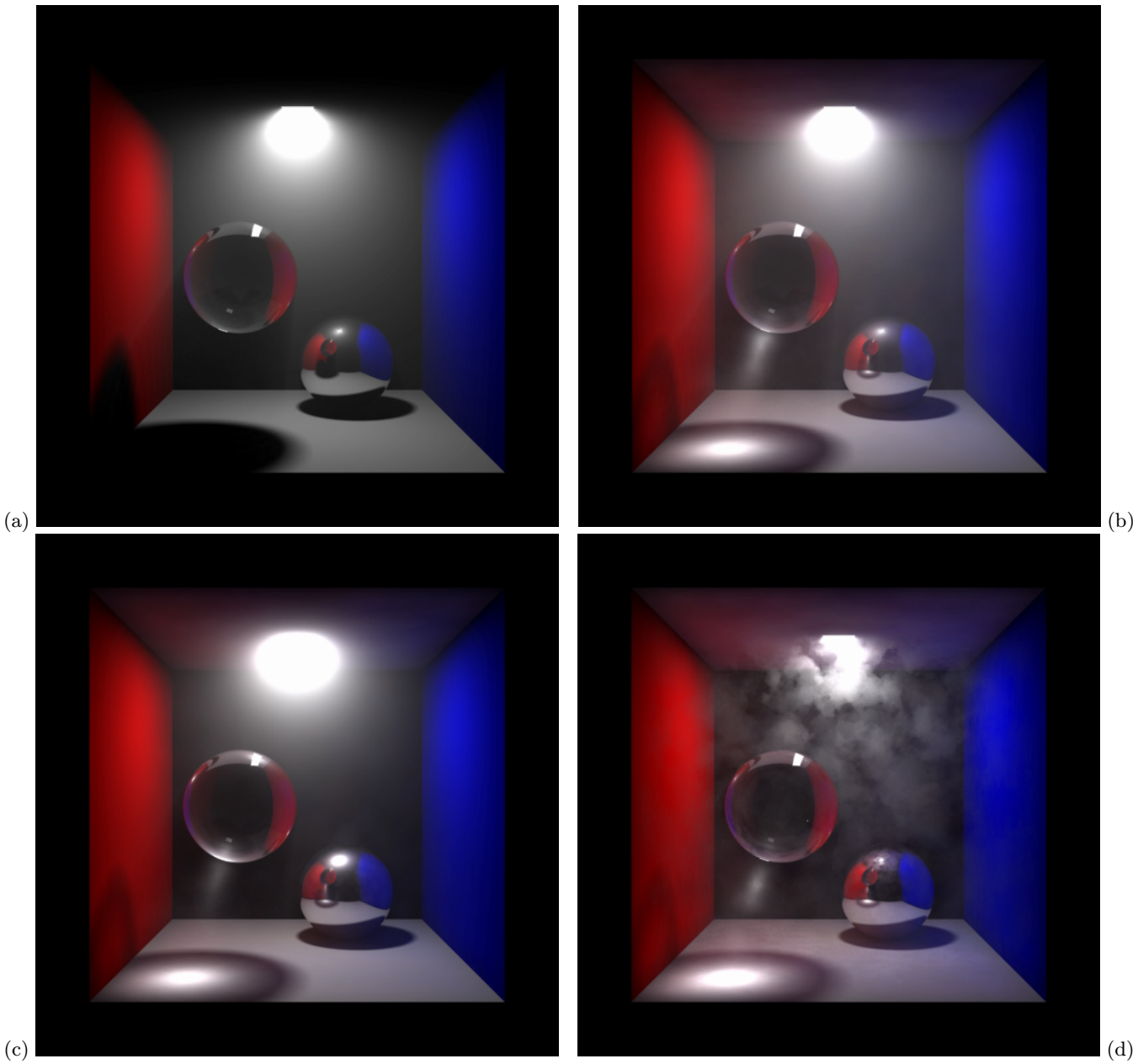


Figure 4: "Cornell boxes": (a) direct illumination; isotropic, homogeneous medium. (b) global illumination; isotropic, homogeneous medium. (c) global illumination; anisotropic, homogeneous medium. (d) global illumination; isotropic, nonhomogeneous medium.

In figure 4(d) the participating medium is nonhomogeneous. A 3D turbulence function [28] was used to model the spatial variation of scattering and extinction coefficients. Generating the photon maps with 200,000 photons took 6 seconds, and rendering took 7 minutes 54 seconds. The radiance estimate in the volume used 50 photons. The higher rendering time is caused by the evaluation of the turbulence function and the fine illumination details which must be resolved by the ray marcher.

Figure 5 shows scattering of sunlight shining through a stained glass window into a dusty room. Some of the photons that pass through the stained glass window are scattered by the dust in the air (using Schlick's approximation of murky Mie scattering). This is particularly visible near the cognac glass on the table top. An important optimization in this scene is that the bright beam of light shining through the window is rendered as direct illumination. This is a reasonable approximation considering the fact that the window glass is thin. The optimization allows the scattering effects in the dusty air to be represented using just 80,000 photons. The illumination of the surfaces in the room is represented using 220,000 photons. The walls, the floor and the ceiling are all displacement mapped and the geometry is approximated using 2.3 million triangles. The photon maps were generated in 27 seconds and the image was rendered in 5 minutes 27 seconds.

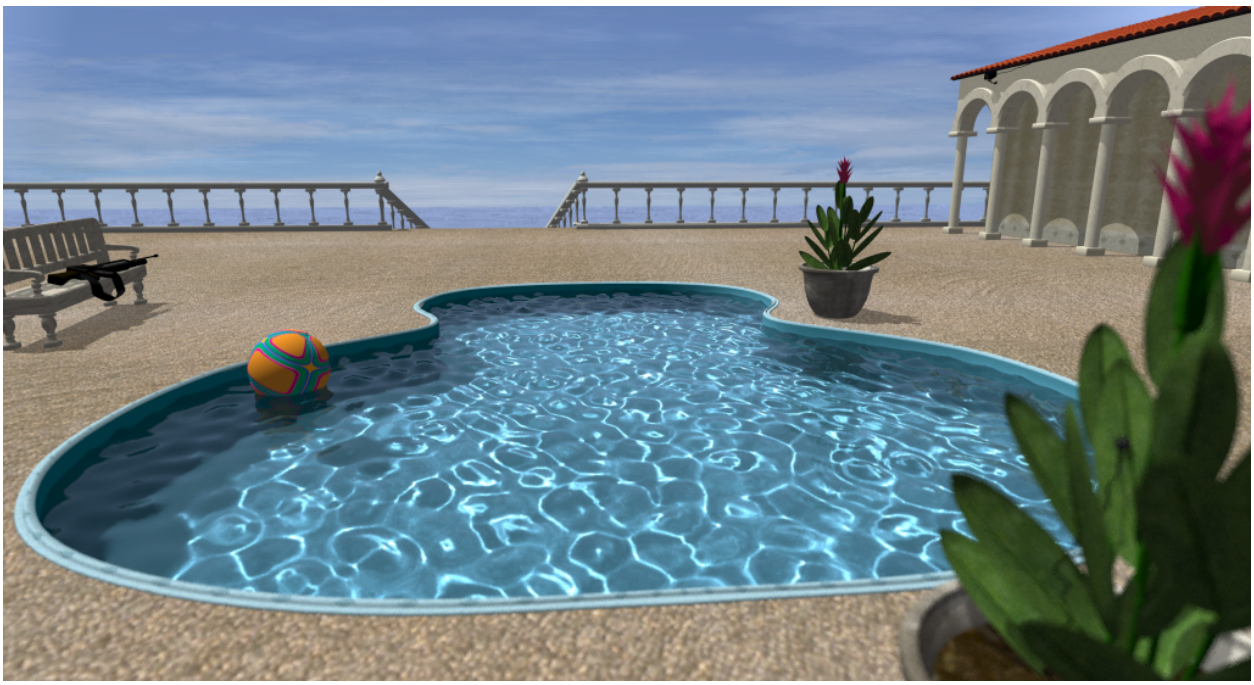


Figure 6: Caustics in a swimming pool seen through a wavy water surface.

Figure 5: Dusty room illuminated by sunlight through a stained glass window.

Figure 6 illustrates a swimming pool with caustics created as light is focused by a wavy water surface. In this scene, the photon maps are used only to simulate the caustics (including volume caustics) resulting from the water in the swimming pool. This is achieved by ignoring all photons which have not intersected the water surface upon the interaction with a diffuse surface. The water surface is displacement mapped and the scene is represented using a total of 2.0 million triangles. Since the water is clear and its extinction coefficient is low, few photons interact with the water before hitting the pool bottom. Therefore only 25,000 photons are stored in the volume photon map. The caustics photon map representing the caustics on the pool bottom and sides contains 475,000 photons. The caustics in the pool are crisp even through the number of photons in the caustics photon map is less than the number of pixels in the image. This is possible since the photons inherently are concentrated where light is focused and intense. The photon maps for this scene were generated in 19 seconds and rendering with a lens simulation to achieve depth-of-field took 1 minute 56 seconds.

Figure 7 shows a simulation of surface and volume caustics in an an underwater scene. The scene is modeled using 1.5 million triangles — these triangles are used primarily for the plants, the pufferfish and the displacement mapped water and sand surfaces. The scene also contains reaction-diffusion textures on the humphead wrasse and glossy reflection on the lowrider crab. The water is modeled as a participating medium to capture the effect of silt particles and plankton, and has highly directional forward scattering. To simulate the caustics and the beams of light in the water, we used 3.0 million photons of which 2.0 million were used in the volume photon map. In order to get sufficiently many photons in the volume photon map, we artificially increased the probability of a photon interacting with the medium (similar to the forced interaction technique used in [26]). This technique can be used to increase the density of photons in the parts of the scene where they contribute most to the overall quality of the illumination representation. The photon maps were generated in 4 minutes 20 seconds and the image was rendered in 24 minutes.

The two images in figure 8 illustrate another simulation of volume caustics in an underwater environment. The two images are from an animation sequence demonstrating gently moving volume caustics. The volume caustics are seen as bright beams of light formed as sunlight is focused due to refraction by the water surface. This particular view of the sun is often seen in pictures of underwater environments;

Figure 7: Underwater scene with volume caustics.

striking examples can be found in [7]. We used 3.0 million photons in the volume photon map for both pictures and a few thousand photons in the caustic photon map to represent the caustics on the fish. For the image in figure 8(a), the photon maps were constructed in 28 seconds and the rendering time was 31 minutes. The image in figure 8(b) required 37 seconds for construction of the photon maps and the rendering time was 32 minutes. The longer rendering time for these images are caused by the fact that the ray marcher must perform a very dense sampling of the medium to adequately capture the crisp beams of light.

The underwater scene in figure 7 was the most memory consuming scene that we rendered (mainly due to the geometry and textures). For the simulation on sixteen processors it reached a maximum resident memory size of 660 MB. By contrast the cloud used 7 MB on one processor of which 4 MB was used for the frame-buffer.

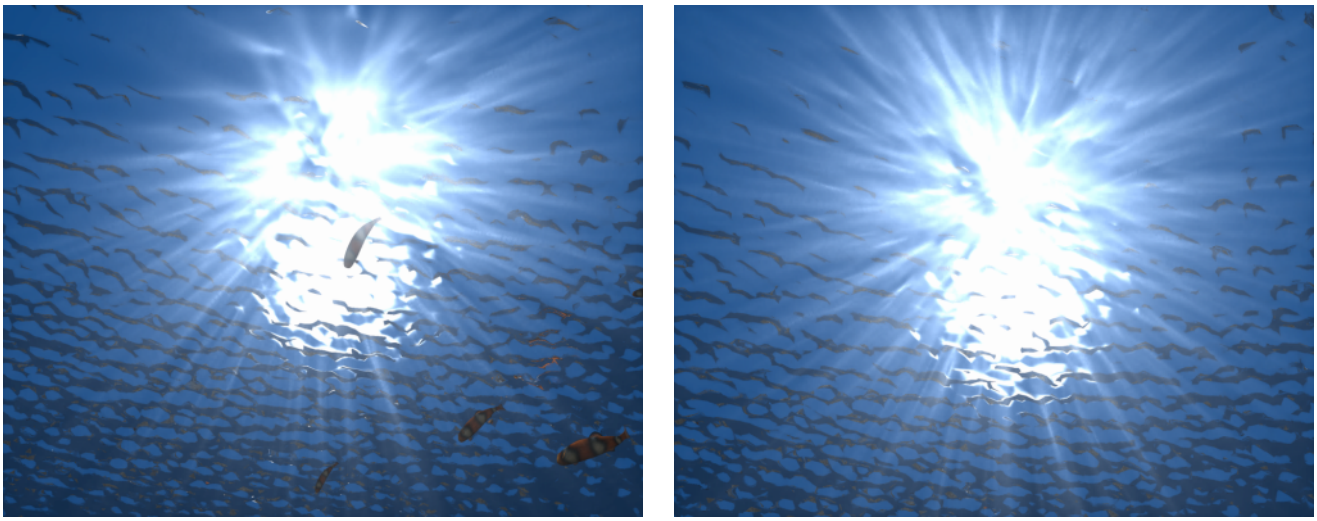
We have tested the method presented here with both Monte Carlo and strictly deterministic quasi-Monte Carlo [24] techniques. In our experience, quasi-Monte Carlo sampling outperforms classical Monte Carlo sampling by distributing the photons more evenly. This leads to an improved photon map radiance estimate with less noise.

We have measured the speedups of the parallel implementation, and for our scenes we achieved speedups between 14.1 and 15.2 on the HP, and speedups between 12.0 and 14.2 on the SGI.

6 Discussion and future work

As our results indicate, the extension of the photon map method to scenes with participating media works very well. The storing of photons in volumes provides a flexible framework and enables this method to simulate complicated media. Our test scenes demonstrate that, in some cases, even a low number of photons in the volume photon map can give very good results. If the lighting situation is more complex, a higher number of photons is required. For the underwater scenes, which contain multiple volume caustics and have large extents, we had to use up to 3.0 million photons in the volume photon map in order to adequately capture the volume caustics. Fortunately, the use of a balanced kd-tree provides an efficient and compact way of handling such a large number of photons.

The radiance estimate for participating media works surprisingly well even with a low number of photons. We compute multiple estimates along each ray traced through a medium, which gives good results using only 50–100 photons per estimate. We found that the combination of ray traced direct illumination and photon map based indirect illumination gives good results. We avoid the costly use of a distribution ray tracer (ie., a final gathering step) to compute the directly visible in-scattered radiance in the participating media. One potential problem with this approach is that the radiance estimate can be incorrect close to the boundary of



(a) (b)

Figure 8: Underwater sunbeams.

a volume if there are too few photons in the photon map. In this situation, the sphere containing the nearest volume photons extends outside of the medium. The volume is still computed as the full sphere and the estimated density of the photons will therefore be too low. Similarly, false photons might be included in the estimate. Most of these problems can be avoided by looking at the incoming direction of the photons, and in our test scenes we have not experienced any visible artifacts due to this.

The memory usage of the photon map method is very favorable in complex scenes. We were able to render high-quality solutions of complex scenes using less than one photon per triangle. The photon map method does not require any additional links between geometry as in for example hierarchical radiosity [11]. It does not use any more memory than what is required for just the photons. This decoupling from the geometry is the primary reason why it can render very complex scenes. It also makes it possible to handle scene geometry represented using for example instancing or implicitly defined functions.

Although the scenes in the result section have only one light source, it would not be a problem to simulate scenes with many light sources. Each light source then contributes relatively less to the illumination in the scene, and therefore less photons should be emitted from each light source.

Even though our results are very encouraging, we believe there is room for further improvements. In particular one might consider the use of visual importance to guide the photons towards the parts of the scene which contribute most to the image. This could be used to reduce the number of photons required in for example the underwater scene. One way of implementing this could be to use an importance map (along the lines of the photon map) which stores importance-carrying particles emitted from the camera. Visual importance might also help address the difficult problem of automatically determining the number of photons needed in the photon maps to obtain an image of a given quality.

Another interesting possibility would be to implement emitting media such as flames. For complex emitting media the sampling of the direct illumination might be too costly. In this case, one could consider storing the photons corresponding to the direct illumination from the media.

7 Conclusion

We have extended the photon map method to simulate global illumination in scenes with participating media by introducing a volume photon map in which photons representing indirect illumination are stored. Furthermore, we have derived a new technique for estimating radiance based on photons stored in volumes. Since the photon map is detached from the geometry, the method is capable of simulating global illumination in complex scenes. For the same reason, the method can use implicit volumetric representations directly. Even though the method is simple, it provides a general framework for simulating light transport in various types of participating media. It can efficiently handle both homogeneous and nonhomogeneous media with isotropic or anisotropic scattering. The method can simulate global illumination effects such as volume caustics, multiple scattering, and color bleeding between surfaces and volumes.

The results demonstrate that the method can handle complex illumination and geometry, has a low level of noise, and is applicable to animations. While simulation of global illumination in scenes with participating media is still computationally expensive, it is now within reach of high-end animation production.

Acknowledgements

Many thanks to Charlotte Manning for modeling the scenes in figures 5–8 and for use of the lowrider crab. Steffen Volz animated the fish in figure 8. Modeling and animation was done using Softimage|3D. Hoang-My Christensen and Rolf Herken proofread several drafts of this paper. Eric Lafortune, Alexander Keller, and the SIGGRAPH reviewers provided helpful comments. Finally, we would like to thank our colleagues at mental images for providing a creative environment and for discussions about this paper. The research and development described herein is funded in part by the European Commission in ESPRIT project 22765 (DESIRE II).

References

[1] James R. Arvo. Backward ray tracing. *ACM SIGGRAPH 86 Course Notes — Developments in Ray Tracing*, 12, 1986.

[2] Jon L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[3] N. Bhate and A. Tokuta. Photorealistic volume rendering of media with directional scattering. *Proceedings of the 3rd Eurographics Workshop on Rendering*, pages 227–245, 1992.

[4] Philippe Blasi, Bertrand Le Saëc, and Christophe Schlick. A rendering algorithm for discrete volume density objects. *Computer Graphics Forum (Proceedings of Eurographics '93)*, 12(3):201–210, 1993.

[5] James F. Blinn. Light reflection functions for simulation of clouds and dusty surfaces. *Proceedings of ACM SIGGRAPH 82*, pages 21–29, 1982.

[6] Per H. Christensen. Global illumination for professional 3D animation, visualization, and special effects. *Rendering Techniques '97 (Proceedings of the 8th Eurographics Workshop on Rendering)*, pages 321–326, 1997.

[7] David Doubilet. *Light in the Sea*. National Geographic, 1989.

[8] David S. Ebert. Volumetric modeling with implicit functions (A cloud is born). *Visual Proceedings of ACM SIGGRAPH 97*, page 147, 1997. Technical Sketch.

[9] David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. *Texturing and Modeling: A Procedural Approach*. AP Professional, 1994.

[10] Andrew S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann, San Francisco, CA, 1995.

[11] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *Proceedings of ACM SIGGRAPH 91*, pages 197–206, 1991.

[12] Paul Heckbert. Adaptive radiosity textures for bidirectional ray tracing. *Proceedings of ACM SIGGRAPH 90*, pages 145–154, 1990.

[13] Henrik Wann Jensen. Global illumination using photon maps. *Rendering Techniques '96 (Proceedings of the 7th Eurographics Workshop on Rendering)*, pages 21–30, 1996.

[14] Henrik Wann Jensen. *The Photon Map in Global Illumination*. PhD thesis, Technical University of Denmark, Lyngby, Denmark, 1996.

[15] Henrik Wann Jensen. Rendering caustics on non-Lambertian surfaces. *Proceedings of Graphics Interface '96*, pages 116–121, 1996.

[16] James T. Kajiya and Brian P. von Herzen. Ray tracing volume densities. *Proceedings of ACM SIGGRAPH 84*, pages 165–174, 1984.

[17] R. Victor Klassen. Modeling the effect of the atmosphere on light. *ACM Transactions on Graphics*, 6(3):215–237, 1987.

[18] Eric P. Lafortune and Yves D. Willems. Bi-directional path tracing. *Proceedings of Compugraphics '93*, pages 145–153, 1993.

[19] Eric P. Lafortune and Yves D. Willems. Rendering participating media with bidirectional path tracing. *Rendering Techniques '96 (Proceedings of the 7th Eurographics Workshop on Rendering)*, pages 92–101, 1996.

[20] Eric Languénou, Kadi Bouatouch, and Michelle Chelle. Global illumination in presence of participating media with general properties. *Proceedings of the 5th Eurographics Workshop on Rendering*, pages 69–85, 1994.

[21] Nelson L. Max. Light diffusion through clouds and haze. *Computer Vision, Graphics, and Image Processing*, 33(3):280–292, March 1986.

[22] Nelson L. Max. Efficient light propagation for multiple anisotropic volume scattering. *Proceedings of the 5th Eurographics Workshop on Rendering*, pages 87–104, 1994.

[23] Gustav Mie. Beiträge zur optik trüber medien, speziell kolloidaler metallösungen. *Annalen der Physik*, 25(3):377–445, 1908.

[24] Harald Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*, volume 63 of *Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pennsylvania, 1992.

[25] Tomoyuki Nishita, Yoshinori Dobashi, and Eihachiro Nakamae. Display of clouds taking into account multiple anisotropic scattering and sky light. *Proceedings of ACM SIGGRAPH 96*, pages 379–386, 1996.

[26] S. N. Pattanaik and S. P. Mudur. Computation of global illumination in a participating medium by Monte Carlo simulation. *Journal on Visualization and Computer Animation*, 4(3):133–152, 1993.

[27] Frederic Pérez, Xavier Pueyo, and François X. Sillion. Global illumination techniques for the simulation of participating media. *Rendering Techniques '97 (Proceedings of the 8th Eurographics Workshop on Rendering)*, pages 309–320, 1997.

[28] Ken Perlin. An image synthesizer. *Proceedings of ACM SIGGRAPH 85*, pages 287–296, 1985.

[29] Holly E. Rushmeier. *Realistic Image Synthesis for Scenes with Radiatively Participating Media*. PhD thesis, Cornell University, Ithaca, New York, 1988.

[30] Holly E. Rushmeier. Rendering participating media: Problems and solutions from application areas. *Proceedings of the 5th Eurographics Workshop on Rendering*, pages 35–56, 1994.

[31] Holly E. Rushmeier and Kenneth E. Torrance. The zonal method for calculating light intensities in the presence of a participating medium. *Proceedings of ACM SIGGRAPH 87*, pages 293–302, 1987.

[32] Robert Siegel and John R. Howell. *Thermal Radiation Heat Transfer, 3rd Edition*. Hemisphere Publishing Corporation, New York, 1992.

[33] Jos Stam. Multiple scattering as a diffusion process. *Rendering Techniques '95 (Proceedings of the 6th Eurographics Workshop on Rendering)*, pages 41–50, 1995.

[34] Eric Veach and Leonidas Guibas. Bidirectional estimators for light transport. *Proceedings of the 5th Eurographics Workshop on Rendering*, pages 147–162, 1994.

[35] Eric Veach and Leonidas J. Guibas. Metropolis light transport. *Proceedings of ACM SIGGRAPH 97*, pages 65–76, 1997.

[36] Bruce Walter, Philip M. Hubbard, Peter Shirley, and Donald P. Greenberg. Global illumination using local linear density estimation. *ACM Transactions on Graphics*, 16(3):217–259, 1997.

[37] Gregory J. Ward. The RADIANCE lighting simulation and rendering system. *Proceedings of ACM SIGGRAPH 94*, pages 459–472, 1994.

[38] Mark Watt. Light-water interaction using backward beam tracing. *Proceedings of ACM SIGGRAPH 90*, pages 377–385, 1990.