Volume Graphics (2007)
H. - C. Hege, R. Machiraju (Editors)

Accelerated Light Propagation Through Participating Media

Richard Lee and Carol O'Sullivan

Graphics, Vision and Visualisation Group, Trinity College Dublin, Ireland

Abstract

Monte Carlo path tracing is a simple and effective way to solve the volume rendering equation. However, propagating light paths through participating media can be very costly because of the need to simulate potentially many scattering events. This paper presents a simple technique to accelerate path tracing of homogeneous participating media. We use a traditional path tracer for scattering near the surface but switch to a new approach for handling paths that penetrate far enough inside the medium. These paths are determined by sampling from a set of precomputed probability distributions, which avoids the need to simulate individual scattering events or perform ray intersection tests with the environment. We demonstrate cases where our approach leads to accurate and more efficient rendering of participating media, including subsurface scattering in translucent materials.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Raytracing

1. Introduction

Many everyday translucent materials such as skin, milk and fruit are still challenging to render in computer graphics. Light which hits the surface of these materials is not simply reflected or absorbed. Instead, some of the light is transported beneath the surface where it may interact with the underlying medium. Modelling this complex subsurface scattering and absorption of light is, in many cases, crucial for realistic image synthesis.

Past research resulted in a number of solutions to this complex light transport problem. Many of these solutions only compute direct illumination of the volume and ignore the contribution due to light that scatters multiple times. Perhaps the simplest and most versatile solution to computing the full global illumination solution within media is the Monte Carlo method. This estimates illumination by tracing a random collection of photons through the environment and simulating what happens to each of them. If a photon enters a participating medium, such as a translucent object, it may bounce internally many times before exiting again. The further the photon penetrates the object, the harder it will be for it to find its way back to the surface and hence, the longer it will take to process these photons. This issue is typically handled in renderers by prematurely terminating the path after a certain number of bounces. However, doing so can lead to biased results or noisy images that are slow to converge.

In this paper, we present a new path sampling technique for photons propagating through a homogeneous participating medium, which results in improved rendering times for scattering media. Instead of simulating the path of a photon as it undergoes a series of scattering events, our technique is able to rapidly transport photons over arbitrarily large distances inside a volume, thus reducing the likelihood that a photon will get trapped scattering inside an object. Our approach enables photons that have penetrated an object to traverse space without requiring phase function sampling to determine new directions for scattered photons, or costly ray intersection tests to determine when photons exit the surface.

2. Related Work

Considering the importance of subsurface scattering to the appearance of many types of material, it is not surprising that a number of approximations have been developed for light transport within translucent materials. One of the first developments was the work of Hanrahan and Krueger [HK93] in which they present an analytical solution for single scattering in stacked layers of different homogeneous media, and an accurate Monte Carlo algorithm to compute the multiple scattering contribution. Lafortune and Willems [LW96] extend bidirectional path tracing to render participating media, and Pauly et al. [PKK00] present a generalization of the Metropolis light transport method to include volume scattering. Premoze et al. [PAS03] use path integral theory to

approximate radiative transfer in forward-scattering, inhomogeneous media.

Jensen and Christensen [JC98] extend the two-pass photon mapping algorithm to scenes with participating media. In the first pass they trace photons through the scene, computing their interaction with participating media and storing the results in a volume photon map. In the second pass they compute direct and indirect illumination within media by using ray marching to compute the single scattering term and the volume photon map to estimate radiance due to multiple scattering. Although their method is practical and efficient in many cases, it is generally considered intractable for highly scattering media due to exorbitant memory requirements [DJ07].

More recently, Jensen et al. [JMLH01, JB02] introduced a fast approximation for subsurface light transport which combines an accurate single scattering computation with a dipole point source diffusion approximation for multiple scattering. Although their method generally produces excellent results for optically thick and highly scattering materials, the underlying assumptions cause inaccuracies for many geometrically complex objects, in particular, those with optically thin features. Donner et al. [DJ07] overcome these limitations by tracing photons inside the medium to capture inter-scattering between surfaces, and applying a quadpole diffusion approximation to improve accuracy.

Chen et al. [CTW*04] makes use of Jensen's work to render inhomogeneous objects by splitting the volume into an inhomogeneous outer layer and a homogeneous inner core. The reflectance field of the voxelized surface layer is precomputed using volume photon mapping, while the homogeneous region approximates the inner core of an inhomogeneous object. This is modelled using the dipole approximation based on the assumption that the light reaching this inner core has diffused due to scattering. A similar approximation by using a homogeneous inner core was employed by Li et al. [LPT05]. They propose a hybrid Monte Carlo method for subsurface scattering that combines path tracing to model scattering events occurring near the surface, with the dipole diffusion approximation to handle paths that penetrate into an isotropic core region. Their approach produces results comparable to a full Monte Carlo simulation at a fraction of the cost.

As well as techniques to accurately model light transport in volumes, many recent developments have focussed on rendering volumetric materials at interactive rates. Mertens et al. [MKB*03] introduce an importance sampling scheme to accelerate subsurface scattering. Dachsbacher and Stamminger [DS03] render subsurface scattering effects by extending the concept of the shadow map to handle translucency. Hegeman et al. [HAP05] apply a fast lighting model to the more general case of inhomogeneous participating media, while Kniss et al. [KPHE02] present an interactive shading model that captures volumetric light attenuation and shadows.

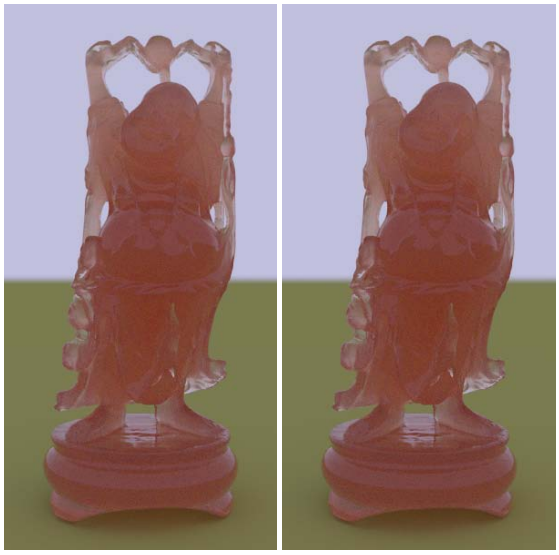

Figure 1: *The left image was rendered in 484 minutes using pure Monte Carlo path tracing. The right image was rendered in 257 minutes using accelerated photon propagation. There is no visible difference between the images indicating that our acceleration technique does not introduce perceptible error into the calculation.*

3. Monte Carlo Path Tracing

We produce images by tracing adjoint photons from the eye, through the scene, until they are absorbed, or hit a light source. Motivated by Morley et al. [MBJ*06], we do not explicitly compute direct lighting and instead rely on importance sampling for variance reduction. This avoids tracing shadow rays, including those within participating media. In other words, we do not perform any ray marching within volumes. Instead, we explicitly simulate the path each photon takes within a participating medium. This generalized approach handles all types of scattering effects within media, including volumetric shadows and caustics.

In our framework, a homogeneous participating medium is defined by a closed mesh which contains the medium. The volume material is described by a set of macroscopic, wavelength-dependent parameters: index of refraction, η, absorption coefficient, σ_a, scattering coefficient, σ_s, and phase function, $p(\theta)$. For $p(\theta)$, we use the empirical Henyey-Greenstein formula, which depends on another wavelength-dependent parameter, g, the mean cosine of the scattering angle.

For translucent materials such as liquids or glass, photons are either reflected or refracted at the boundary based on

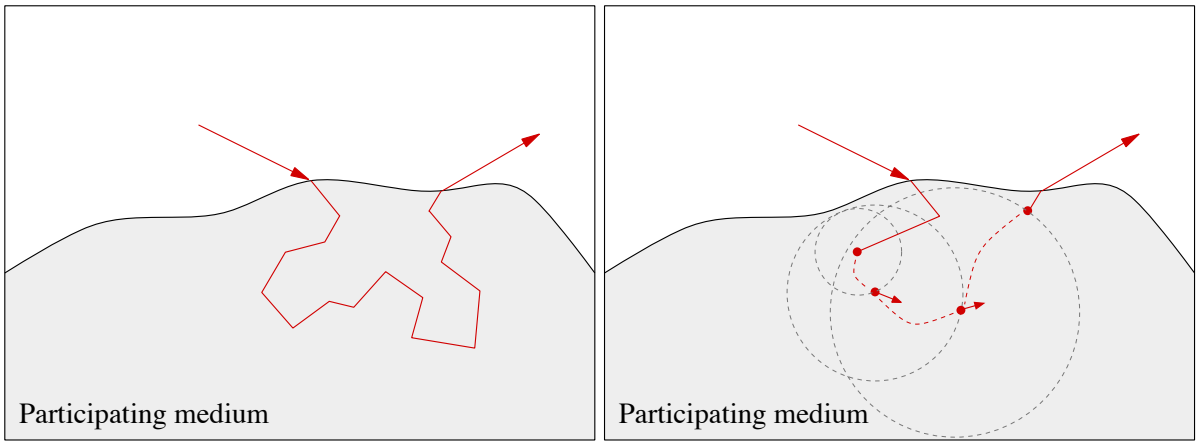

Figure 2: *On the left, a photon enters a participating medium and undergoes a series of scatters before exiting; on the right, when the photon reaches a certain distance within the medium, it is "teleported" between spheres of different radii until it is back within reach of the surface. Less computation is required to compute the path of the photon on the right.*

the Fresnel equations. For other types of participating media such as clouds or smoke, no interaction occurs with the boundary. If a photon enters the medium, we probabilistically decide if an interaction occurs before the photon exits again. To do this we first compute the time at which the next absorption and scattering events will occur:

$$t_a = -\log(\xi_1)/\sigma_a$$
$$t_s = -\log(\xi_2)/\sigma_s$$

where ξ_1, ξ_2 are uniform random numbers in $[0,1]$. We then compare these values to the time at which the photon will exit the medium. If the exit time is smaller than t_a and t_s then the photon passed through the medium unaffected; if $t_a < t_s$, the photon is absorbed; otherwise the photon is scattered by sampling the phase function to determine its new direction.

For optically thick, high albedo materials where the probability of a scattering event is much greater than an absorption event, computation can easily become dominated by simulating the paths of photons as they scatter through an object. Our objective therefore is to reduce the amount of computation required to simulate the propagation of a photon through a volume.

4. Accelerated Scattering

4.1. Overview

Consider a photon travelling through a participating medium. At some instant in time, the photon has position $\mathbf{p}$ and direction $\mathbf{d}$. Now consider a sphere whose radius is equal to the distance from $\mathbf{p}$ to the closest point on the boundary of the medium. Assuming the photon is not absorbed, it may undergo a series of scatters within the sphere. At some future time, the photon will cross the surface of the sphere at some random position, say $\mathbf{p}'$, with random outgoing direction $\mathbf{d}'$.

Our approach provides a way to "teleport" the photon from position $\mathbf{p}$ with direction $\mathbf{d}$, to a new position $\mathbf{p}'$ with new direction $\mathbf{d}'$, without having to simulate the individual scattering events of the photon within the sphere. The information that determines the likelihood of a photon leaving the sphere at some point in some direction is precomputed and sampled at render time to determine new random positions and directions for photons.

Note that the new position $\mathbf{p}'$ of the photon will not necessarily be any closer to the surface of the medium than $\mathbf{p}$. However, by repeating this process of teleporting the photon between the surfaces of spheres of different radii, where the spheres are chosen to be as large as possible while still being fully contained within the medium, we avoid the processing which would otherwise be required to simulate the scattering of the photon within each of these spheres. The idea here is that, if the radii of the spheres are many times greater than the photon's mean free path length, then the photon will traverse space with far fewer computations, as illustrated in Figure 2.

The following sections will discuss how we precompute the probability distributions that allow us to predict where, and in what direction, a photon will exit a sphere, how this data is used to choose new outgoing positions and directions on a sphere, and how this sampling scheme is used in the context of our path tracer to accelerate photon tracing within participating media.

4.2. Precomputation

The previous section discussed transporting a photon with position $\mathbf{p}$ and direction $\mathbf{d}$ to a new outgoing position $\mathbf{p}'$ on the surface of a sphere of some arbitrary radius, with new direction $\mathbf{d}'$. This should be done in such a way that the prob-

ability of choosing a particular outgoing position and direction is equal to the probability that the original photon would have exited the sphere with that position and direction. In other words, we need to importance sample the sphere to determine the new position and direction of the photon. To be able to perform this importance sampling we first need to compute a numerical probability density function (PDF).

The PDF we are interested in describes the probability of choosing a particular outgoing position and direction from the sphere, given an initial position and direction. This is a 9D function. Fortunately, we can reduce the dimensionality of this function to 3D, without loss of generality, by first defining a translation which maps **p** to the origin, and a rotation which maps **d** to the unit vector pointing straight up. Note that the inverse of this transformation can be easily applied, as we will see in the next section, to map the PDF back to the original coordinate system given by **p** and **d**. Defining the PDF in this new coordinate system reduces it to a 4D function of outgoing position and direction, since the initial position and direction are now fixed. For a sphere of a given radius r, we write the outgoing position in spherical coordinates as (r,α,β), the outgoing direction as $(1,\theta,\phi)$, and the PDF as $p(\alpha,\beta,\theta,\phi)$. Due to radial symmetry about the polar axis, the PDF is separable into $p(\beta) = 1/2\pi$ and $p(\alpha,\theta,\phi)$.

Using Monte Carlo simulation, we construct a 3D piecewise constant representation of $p(\alpha,\theta,\phi)$, and also compute the probability that a photon will be absorbed before exiting the sphere. Algorithm 1 shows how to simulate a single photon. This procedure is run for a large number of photons, e.g., one million. The probability of absorption is computed simply as the fraction of photons absorbed. To compute $p(\alpha,\theta,\phi)$, we subdivide each of these three angles over their range to form a set of bins, i.e., each bin corresponds to a unique set of outgoing positions and directions. The angles θ and ϕ are subdivided such that each bin covers an equal solid angle, whereas α is uniformly subdivided to provide more resolution at the poles. This is beneficial, considering that many of the materials we are interested in are highly forward scattering.

We associate a scalar value with each bin, equal to the probability that a photon will exit the sphere through that bin. For each photon, we need to map its exit position and direction from Cartesian coordinates to α,θ,ϕ coordinates. Converting the position and direction into spherical angles gives four values but, as discussed above, we know that the azimuthal angle of the position has a constant PDF of $1/2\pi$. This implies that we can disregard that angle if we first rotate the position and direction of each photon to make its azimuthal angle zero. From here we can easily map the (α,θ,ϕ) value to its corresponding bin and hence determine the fraction of photons that exited the sphere through that bin, in order to compute its probability.

Algorithm 1: Simulating a photon in a sphere.

```
pos ← (0,0,0)
dir ← (0,1,0) // up
state ← SCATTER
repeat
    t_exit ← IntersectSphere((0,0,0), radius, pos, dir)
    t_a ← −log(GetRandom())/σ_a
    t_s ← −log(GetRandom())/σ_s
    if min(t_a, t_s) >= t_exit then
        exit_pos ← pos + dir × t_exit
        exit_dir ← dir
        state = EXIT
    else
        if t_a < t_s then
            state = ABSORB
        else
            // scatter
            pos ← pos + dir × t_s
            dir ← ImportanceSamplePhaseFunc(dir)
        end
    end
until state ≠ SCATTER
```

4.3. Sampling

Given an initial position and direction for a photon, which is located at the centre of a sphere of a certain radius, we first check if the photon gets absorbed before reaching the surface of the sphere. This is done by comparing a uniform random variable to the probability of absorption for the sphere. If the photon is not absorbed then we need to compute its new direction and position on the surface of the sphere. The multidimensional, piecewise constant probability density, $p(\alpha,\theta,\phi)$, described in the previous section, provides us with the information we need to randomly sample according to the correct distribution. We begin by describing how to convert random numbers over $[0,1]^3$ to samples drawn from the 3D (α,θ,ϕ) distribution.

The inversion method [PH04] can be used to efficiently draw samples from a 1D PDF by computing a 1D cumulative distribution function (CDF), $P(x)$, and evaluating its inverse $P^{-1}(\xi)$ at uniformly distributed random locations, $\xi \in [0,1]$. Samples can also be drawn from multidimensional distributions by considering a "cascading set" of 1D CDFs [LRR05]. In this case, each dimension is sampled in turn, based on the values chosen for previous dimensions. In the case of $p(\alpha,\theta,\phi)$, we need to compute CDFs for $P(\phi|(\alpha,\theta))$, $P(\theta|\alpha)$ and $P(\alpha)$. Given these, we can then draw an (α,θ,ϕ) sample by first drawing an α sample from $P(\alpha)$. Using this, we sample θ from $P(\theta|\alpha)$. And finally, given (α,θ), we draw a ϕ sample from $P(\phi|(\alpha,\theta))$. Algorithm 2 shows how this translates into pseudocode. Note that the *Sample()* function draws random samples from 1D CDFs. Its output is the index of the bin from which a sam-

ple was draw, and the probability value in $[0,1]$. Probability values in $[0,1]^3$ can be converted to (α,θ,ϕ) values by multiplying by $(\pi,\pi,2\pi)$.

The next step is to compute β, the azimuthal angle describing the outgoing position. As discussed earlier, $p(\beta)$ is constant, so we can compute its value by drawing a sample uniformly from $[0,2\pi]$. This angle is then used to rotate the outgoing direction (by adding β to ϕ), so that it will point in the correct direction relative to the photon's outgoing position. We now need to convert $(\alpha,\beta,\theta,\phi)$ into the outgoing position and direction by mapping from spherical to Cartesian coordinates. However, remember that $p(\alpha,\theta,\phi)$ assumed a photon located at the origin, directed upwards. Since we are really interested in photons located at different positions with arbitrary directions, the final step is to transform the photon's new position and direction so that it is correct relative to the old position and direction, as shown in Algorithm 2.

Algorithm 2: Sampling a photon's new state.

input : **pos**, **dir**
output: **exit_pos**, **exit_dir**
if $GetRandom() >$ absorption_prob **then**
  $(bin_\alpha, t_\alpha) \leftarrow cdf_\alpha.Sample()$
  $(bin_\theta, t_\theta) \leftarrow cdf_\theta[bin_\alpha].Sample()$
  $(bin_\phi, t_\phi) \leftarrow cdf_\phi[bin_\alpha][bin_\theta].Sample()$
  $\alpha \leftarrow \pi t_\alpha$
  $\beta \leftarrow 2\pi \times GetRandom()$
  $\theta \leftarrow \pi t_\theta$
  $\phi \leftarrow \beta + 2\pi t_\phi$
  $(\mathbf{u},\mathbf{v},\mathbf{w}) \leftarrow MakeOrthonormalBasis(\mathbf{dir})$
  exit_pos $\leftarrow$ **pos** $+ r\cos\beta\sin\alpha\,\mathbf{u} + r\sin\beta\sin\alpha\,\mathbf{v} + r\cos\alpha\,\mathbf{w}$
  exit_dir $\leftarrow \cos\phi\sin\theta\,\mathbf{u} + \sin\phi\sin\theta\,\mathbf{v} + \cos\theta\,\mathbf{w}$
end

4.4. Rendering

So far we have described how to transport a photon from the centre of a sphere of a given radius to a point on its surface. In this section, we will discuss how this is used in the context of our path tracer.

When a photon enters a participating medium, we evaluate its distance, d, to the boundary of the volume. If d is less than a threshold τ, we use the traditional approach described in Section 3 to propagate the photon. Otherwise we do the following: First, we use a binary search to select a radius, r, from a finite set of radii, for which the PDF described earlier has been precomputed. We choose the largest r possible that is less than $(d-\varepsilon)$ (ε is a small value that ensures that the photon stays within the volume). We now consider the photon to be located at the centre of a sphere of radius r. Since we have the PDF for this sphere, we use Algorithm 2 to determine if the photon is absorbed and, if not, where the photon moves to on the sphere's surface. With the photon at its new position and direction, we reevaluate d and repeat the process until the photon exits the volume.

We choose τ to equal the mean free path length for scattering, $1/\sigma_s$. This is also the minimum radius for which we precompute a PDF. The radius increment and maximum radius that define the rest of the PDFs depend on the scattering coefficients and the scale of the volumes we wish to render. However, but we found that ten or more integral multiples of the mean free path length worked well. Also, since the PDF is dependent on the material properties of the volume, which themselves are wavelength-dependent, we need to compute separate PDFs for sets of radii corresponding to red, green and blue wavelengths. At render time, we sample from whichever of these is closest to the wavelength of the photon in question.

To evaluate the distance d to the closest point on the bounding mesh of the volume, we use a signed distance function, sampled on a uniform grid. We can compute a quick approximation to this function by exploiting our raytracer's ability to perform fast ray intersection tests. At each grid point for which we want to calculate signed distance, we uniformly sample directions over the sphere and shoot rays in these directions to determine the distance to the closest intersection point. The sign is given by the dot product of the ray direction and the normal at the intersection point. Although this method is not robust, it is easy to implement and works well for the scenes we have tested. We can also avoid both the trilinear interpolation required to evaluate signed distances, and the search required to find the best radius, by precomputing a uniform grid where each cell stores either an index into the list of radii or -1 if the cell is too close to the boundary. By mapping the photon's position to its corresponding cell, we can retrieve the appropriate radius in a single look-up, without having to evaluate the distance to the boundary at run-time.

4.5. Discussion

We emphasize that the only bias our acceleration scheme introduces is due to sampling from discrete instead of continuous PDFs. Photons only begin teleporting when they are determined to lie within the medium, and teleporting a photon does not imply that an interaction with the medium occurred. For example, if the medium is non-scattering then a photon will propagate through it along an approximately straight line (depending on the the bin resolution), even as it teleports between points.

Also note that teleporting itself does not have an effect on the probabilities of interactions occurring. Consider, for example, the probability, $p(A)$, that a photon will be absorbed along a path of length d: $p(A) = 1 - e^{-\sigma_a d}$. Now consider that the photon is teleported by distance t, somewhere between its initial position and d. The photon must

now traverse two subpaths, of length t and $d-t$ (analogous to spheres of radii t and $d-t$). Let $p(B)$ and $p(C)$ be the probabilities of absorption along each of these subpaths: $p(B) = 1 - e^{-\sigma_a t}$, $p(C) = 1 - e^{-\sigma_a (d-t)}$. Since the photon is tested for absorption along both subpaths, the probability that the photon is absorbed along the full path is $p(A \vee B) = p(A) + p(B) - p(A)\,p(B) = 1 - e^{-\sigma_a d} = p(C)$, which shows that teleporting the photon does not affect the probability of absorption. The same argument holds for scattering events.

5. Results

We have used our Monte Carlo path tracer to render comparison images with and without the acceleration technique described in this paper. The images in Figure 1 were rendered at 300x600 resolution with 2,500 samples per pixel, while those in Figure 3 were rendered at 500x500 with 1,600 samples per pixel. All renderings were performed on a Xeon 3.6GHz PC.

The Buddha model in Figure 1 is rendered with a translucent appearance by enclosing the participating medium within a dielectric boundary mesh with an index of refraction of 1.5. The medium properties were chosen such that the average scattering albedo is approximately 0.9, and the average cosine of the scattering angle is 0.85. These values describe a highly scattering medium that is similar to the properties of many natural materials such as skin. Note that the Buddha model we render is optically thin in many places due to its scale. Our approach performs significantly better for optically thick configurations due to the increased number of photon bounces within the volume, but even in this case, we achieve an approximately $2\times$ performance increase.

The number of scattering events is reduced by 61% due to teleportation. Overall, 2.6 billion scatters were replaced by 0.7 billion teleports, due to the fact that each teleport aggregates many scattering events. In addition, with the accelerated algorithm, 57% of absorption events are detected using the conditional at the top of Algorithm 2. This test is much faster to perform than computing the time of absorption and comparing that to the time the photon exits the boundary, as described in Section 3.

Figure 3 shows a horse model rendered as a high albedo participating medium. Again, the speedup is approximately $2\times$. In this case, 59% of the scattering events are avoided by teleportation. This means that the remaining 41% of scatters occur very close to the boundary of the medium and are not accelerated. For the scenes we have tested, this seems to be the limiting factor in achieving greater than $2\times$ speedup. Note that, for this scene, the average number of scattering events per pixel sample is approximately 5, i.e., each teleport does not avoid a substantial number of scatters. Frisvad et al. [FCJ07] point out that, for materials such as milk, the average photon experiences several thousand scattering events within the medium. We have not investigated how our algorithm performs for various scattering parameters but we expect substantially greater savings for highly scattering materials such as milk.

For the Buddha scene, the time to precompute the 24 PDFs (a set of 8 radii for each of red, green and blue material parameters) was approximately 3 minutes. This preprocess only needs to be performed once for each material. Each PDF was quantized into $16\times16\times16$ bins and computed using one million photons. Although we have not examined sensitivity of results to the number of bins, we have not noticed any artifacts due to tabular sampling of the PDFs. Finally, the storage space required for the data is a very moderate 400 KB.

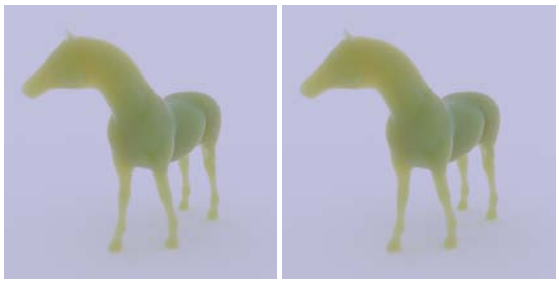

Figure 3: *The image on the left shows a participating medium rendered in 135 minutes using pure Monte Carlo path tracing. The image on the right shows the same scene rendered in 71 minutes using accelerated photon propagation. Again, there is no perceptible difference between the images.*

6. Conclusions and Future Work

In this paper we have presented a new method for accelerating the propagation of photons in scattering media, which does not compromise the accuracy of a full Monte Carlo solution. Our technique performs best for highly forward scattering, optically thick media, in which photons are expected to scatter multiple times before being absorbed or exiting the medium.

In the future we plan to more thoroughly investigate the conditions under which our acceleration technique leads to improved performance. We also wish to determine how important it is for image quality to allow photons to scatter deep within media, instead of performing early truncation of scattering paths as is usually done. We also note that, instead of importance sampling the expected paths (defined by the phase function) as we have done, it would be easy to modify our method so that alternate directions for scattered light could be specified, e.g., towards light sources or towards the closest boundary point in a volume. Such importance sampling would require calculating the probability of photons choosing these alternate, predetermined paths. It

would be interesting to investigate if the reduction in variance is enough to offset the extra cost associated with evaluating these PDFs.

References

[CTW*04] CHEN Y., TONG X., WANG J., LIN S., GUO B., SHUM H.-Y.: Shell texture functions. *ACM Trans. Graph. 23*, 3 (2004), 343–353.

[DJ07] DONNER C., JENSEN H. W.: Rendering translucent materials using photon diffusion. In *Eurographics Symposium on Rendering* (2007).

[DS03] DACHSBACHER C., STAMMINGER M.: Translucent shadow maps. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 197–201.

[FCJ07] FRISVAD J. R., CHRISTENSEN N. J., JENSEN H. W.: Computing the scattering properties of participating media using lorenz-mie theory. In *SIGGRAPH* (2007).

[HAP05] HEGEMAN K., ASHIKHMIN M., PREMOŽE S.: A lighting model for general participating media. In *I3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2005), ACM Press, pp. 117–124.

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

[JB02] JENSEN H. W., BUHLER J.: A rapid hierarchical rendering technique for translucent materials. In *SIGGRAPH* (2002), pp. 576–581.

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

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

[KPHE02] KNISS J., PREMOZE S., HANSEN C., EBERT D.: Interactive translucent volume rendering and procedural modeling. In *VIS '02: Proceedings of the conference on Visualization '02* (Washington, DC, USA, 2002), IEEE Computer Society, pp. 109–116.

[LPT05] LI H., PELLACINI F., TORRANCE K. E.: A hybrid monte carlo method for accurate and efficient subsurface scattering. In *Rendering Techniques* (2005), pp. 283–290.

[LRR05] LAWRENCE J., RUSINKIEWICZ S., RAMAMOORTHI R.: Adaptive numerical cumulative distribution functions for efficient importance sampling. In *Rendering Techniques* (2005), pp. 11–20.

[LW96] LAFORTUNE E. P., WILLEMS Y. D.: Rendering participating media with bidirectional path tracing. In *Proceedings of the Eurographics workshop on Rendering techniques '96* (London, UK, 1996), Springer-Verlag, pp. 91–100.

[MBJ*06] MORLEY R. K., BOULOS S., JOHNSON J., EDWARDS D., SHIRLEY P., ASHIKHMIN M., PREMOZE S.: Image synthesis using adjoint photons. In *Graphics Interface* (2006), pp. 179–186.

[MKB*03] MERTENS T., KAUTZ J., BEKAERT P., REETH F. V., SEIDEL H.-P.: Efficient rendering of local subsurface scattering. In *PG '03: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications* (Washington, DC, USA, 2003), IEEE Computer Society, p. 51.

[PAS03] PREMOZE S., ASHIKHMIN M., SHIRLEY P.: Path integration for light transport in volumes. In *Rendering Techniques* (2003), pp. 52–63.

[PH04] PHARR M., HUMPHREYS G.: *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, 2004.

[PKK00] PAULY M., KOLLIG T., KELLER A.: Metropolis light transport for participating media. In *Rendering Techniques* (2000), pp. 11–22.