A Practical Analytic Single Scattering Model for Real Time Rendering*

Bo Sun
Columbia University

Ravi Ramamoorthi
Columbia University

Srinivasa G. Narasimhan
Carnegie Mellon University

Shree K. Nayar
Columbia University

Abstract

We consider real-time rendering of scenes in participating media, capturing the effects of light scattering in fog, mist and haze. While a number of sophisticated approaches based on Monte Carlo and finite element simulation have been developed, those methods do not work at interactive rates. The most common real-time methods are essentially simple variants of the OpenGL fog model. While easy to use and specify, that model excludes many important qualitative effects like glows around light sources, the impact of volumetric scattering on the appearance of surfaces such as the diffusing of glossy highlights, and the appearance under complex lighting such as environment maps. In this paper, we present an alternative physically based approach that captures these effects while maintaining real-time performance and the ease-of-use of the OpenGL fog model. Our method is based on an explicit analytic integration of the single scattering light transport equations for an isotropic point light source in a homogeneous participating medium. We can implement the model in modern programmable graphics hardware using a few small numerical lookup tables stored as texture maps. Our model can also be easily adapted to generate the appearances of materials with arbitrary BRDFs, environment map lighting, and precomputed radiance transfer methods, in the presence of participating media. Hence, our techniques can be widely used in real-time rendering.

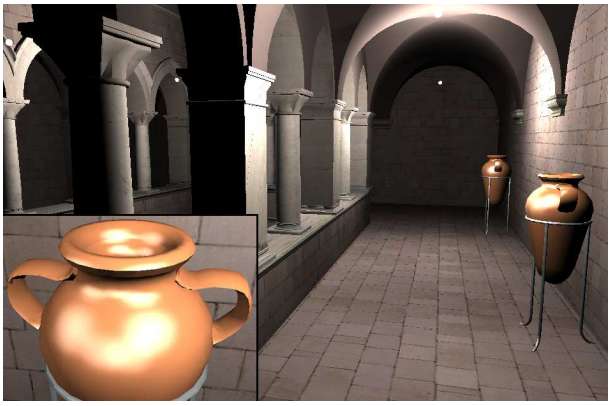
1 Introduction

Many real-time rendering applications like games or interactive simulations seek to incorporate atmospheric effects such as mist, fog and haze. These participating media lead to a number of qualitative effects not present in clear-day conditions (compare figure 1a with our result in figure 1c). For instance, there are often glows around light sources because of scattering. The shading on objects is also softer, with specular highlights diffused out, dark regions brightened and shadows softer. It is critical to capture these effects to create realistic renderings of scenes in participating media.

In computer graphics, the approaches for capturing these effects represent two ends in the spectrum of speed and quality. For high-quality rendering, a number of Monte Carlo and finite element techniques have been proposed. These methods can model very general volumetric phenomena and scattering effects. However, they are slow, usually taking hours to render a single image. Significant gains in efficiency can generally be obtained only by substantial precomputation, and specializing to very specific types of scenes.

At the other extreme, perhaps the most common approach for interactive rendering is to use the OpenGL fog model, which simply blends the fog color with the object color, based on the distance of the viewer (figure 1b). The fog model captures the attenuation of surface radiance with distance in participating media. This model is also popular because of its simplicity—implementation requires almost no modification to the scene description, and the user need only specify one parameter, β, corresponding to the scattering coefficient of the medium (density of fog). However, many qualitative

*e-mail: {bosun,ravir,nayar}@cs.columbia.edu; srinivas@cs.cmu.edu



(a) Clear day

(b) OpenGL fog

(c) Our model

Figure 1: *Rendered images of a scene with 66,454 texture-mapped triangles and 4 point lights. The insets show an image for another view of the vase, with highlights from all 4 sources, to amplify shading differences. (a) Standard OpenGL rendering (without fog), (b) OpenGL fog which captures attenuation with distance and blending with fog color, and (c) Our real-time model, that includes the glows around light sources, and changes to surface shading such as dimming of diffuse radiance (floor and wall), brightening of dark regions (back side of pillars and vases) and dimming and diffusing of specular highlights (inset). All the visual effects in this complex scene are rendered by our method at about 20 frames per second.*

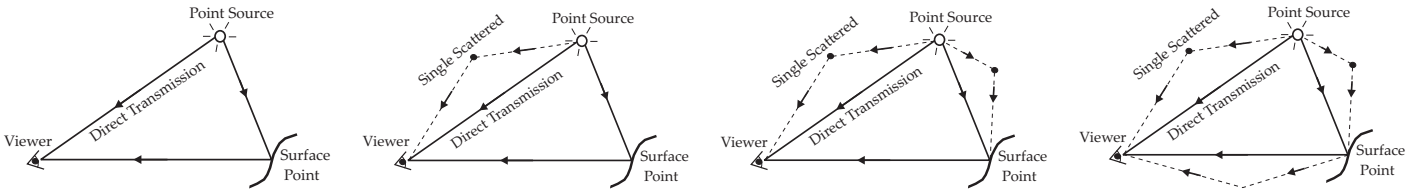


Figure 2: *Diagrams showing three cases of how light travels to the viewer through the participating medium. In (a) light travels in a straight line and directly reaches the surface and the viewer. This is essentially what previous interactive models such as OpenGL fog compute. In (b), in addition to what happens in (a), airlight scatters to the viewer and produces effects like glows around the light source. In (c), in addition to what happens in (b), airlight also scatters to the surface and gets reflected, leading to effects such as the diffusing out of specular highlights and brightening of darker regions. In image (d), reflected rays from the surface also scatter to the viewer.*

effects are missing, such as the glows around light sources, the effect of scattering on object shading, and the ability to incorporate complex lighting effects like environment maps.

In this paper, we take a significant step towards improving the realism of rendered images with participating media (figure 1c), while maintaining the real-time performance and ease of use of the OpenGL fog model. Our model can be implemented as a simple vertex or pixel shader (pseudocode in figure 13), allowing it to be easily added to almost any interactive application. The method can also be applied with complex lighting, allowing environment mapping and precomputed radiance transfer to be used interactively with participating media for the first time (figures 15 and 16).

Figure 2 illustrates three important visual effects due to light transport in scattering media. In this discussion, and this paper, we assume single scattering (i.e. that light scatters at most once in the medium), which is a common approximation in volumetric scattering and can be shown to be accurate in many common situations such as thin fog. Figure 2a corresponds to *direct transmission* of light from the source or surfaces to the viewer. We can simply attenuate the clear-day radiance values based on the distance (optical thickness). This simple approach is essentially what interactive models like OpenGL fog implement. Figure 2b also includes the glows around light sources, commonly referred to as *airlight* [Koschmeider 1924]. Glows occur because light reaches the viewer from different directions due to scattering in the atmosphere. Figure 2c further includes the effect of airlight on the outgoing *surface radiance*, leading to effects such as the spreading out of specular highlights and softening of shadows. These are important effects, usually neglected in previous interactive methods. Our model renders all of the effects in figure 2c in real-time.

Figure 2d illustrates the case where the surface radiance is single scattered in addition to being attenuated, before reaching the view point. On one hand, the attenuation decreases the brightness of the radiance at the surface according to the distance of the surface from the viewer. On the other hand, the single scattering results in slight brightening and blurring of this surface radiance. Implementing the latter effect requires a depth-dependent convolution. In this paper, we will only consider attenuation of surface radiance, and we will set aside a more thorough investigation of the latter effect for future work[1]. The specific technical contributions of this paper are:

Explicit Compact Formula for Single Scattering: The common approach to using single scattering is to numerically integrate brightness contributions while marching along the viewing ray. However, this approach is too slow for interactive applications, which require an explicit formula such as the OpenGL fog model. One of the main contributions of this paper is the derivation of an explicit compact formula for the single scattering from an isotropic point source in a homogeneous participating medium, by analytically integrating the single scattering equations. This *airlight model* (section 3) allows us to simulate effects like the glows around light sources (figure 2b). We can also use the model to calculate the effects of scattering on the surface shading (figure 2c). These calculations are very expensive even for numerical integration, because we must consider incident airlight from the entire visible hemisphere. However, they can be directly implemented using our explicit *surface radiance model* (section 4).

Implementation on Programmable Graphics Hardware: We speculate that an explicit formula for the single scattering integrals has previously not been derived because of the complexity of the calculations involved. In this paper, we reduce these difficult integrals to a combination of analytic functions that depend only on the physical parameters of the problem, and a few lookups of tabulated 2D functions, that have no simple analytic form, but are smooth and purely numerical—independent of the physical parameters. The numerical functions can be precomputed and stored as 2D texture maps, and the entire analytic computation and table lookups can be implemented in simple pixel or vertex shaders in modern programmable graphics hardware (section 5).

Extensions to Complex Lighting and BRDFs: It is also possible to extend our airlight and surface radiance models to incorporate more complex illumination models and material properties (section 6). Mathematically, we derive a point-spread function (PSF) to represent the glow around a light source. We can convolve an environment map with this PSF to get the appearance of a foggy scene under natural lighting. We can also use a frequency domain spherical harmonic representation to enable rendering with arbitrary BRDFs, and add in shadows and interreflections with precomputed radiance transfer methods. This approach enables methods such as environment mapping and precomputed radiance transfer to be used with volumetric scattering effects for the first time.

Our goal is to achieve interactive rendering of participating media. To enable this, and derive an explicit compact expression that can be implemented in real-time, we make a number of assumptions—isotropic point light sources, homogeneous media, the single scattering approximation, and no cast or volumetric shadows (shadows can, however, be added using precomputed light transport methods). More complex and general scattering effects are certainly desirable in many situations, but are not possible to obtain at real-time rates for general scenes. On the other hand, our method captures most of the important visual effects of scattering, while being very simple to add to any interactive application.

2 Related Work

The literature on simulating volumetric effects is large, going back to [Blinn 1982], and we only discuss important representative papers. Most techniques are based on numerical or analytic approximations to the radiative transfer equation [Chandrasekhar 1960]. Monte Carlo ray tracing methods were adapted by computer graphics researchers to render impressive effects including multiple scattering and non-homogeneous media [Kajiya and Herzen 1984; Max 1994; Jensen 2001]. However, such methods can take hours to render a single image. To speed up rendering, numerical methods that only simulate single scattering have also been proposed [Pattanaik and Mudur 1993; Nakamae et al. 1990; Sakas 1990; Rushmeier and Torrance 1987]. However, they still require significant running times, and are not suitable for interactive applications.

[1] Single scattering from different surface points in the scene can partially compensate for the loss of brightness due to attenuation. Neglecting this can produce consistently darker images, especially for indoor scenes.

Hardware-accelerated numerical methods: A number of recent hardware-accelerated techniques can significantly decrease the running times of numerical simulations, although they are still usually not fast enough for many interactive applications such as games. Dobashi et al. [2002] describe a multi-pass rendering technique that numerically integrates the single scattering equations, using graphics hardware to accumulate the results at a number of planes in the scene, similar to volume rendering. Harris and Lastra [2001] render clouds by including a forward scattering term in addition to single scattering. Note that their method is geared toward the case when the viewer is far from the clouds, and they apply a different and slower approach when the viewer and scene are immersed inside the medium, as is the scenario in our work.

These methods are intended to apply to specific phenomena like the sky or clouds [Dobashi et al. 2002; Riley et al. 2004; Harris and Lastra 2001]. This allows them to make use of complex tabular volume specifications, precomputed lighting solutions or multipass rendering techniques to produce effects including inhomogeneous media and simple heuristics for multiple scattering. They allow for viewpoint, and in a few cases interactive lighting variation, but usually fix the medium properties and scene specification.

In contrast, our technique, while focusing on homogeneous media and single scattering, can be encapsulated in a simple shader for general scenes, and allows for real time variation of the viewpoint, lighting, scattering properties of the medium, and even scene geometry and reflectance. Another major benefit of our method is that it addresses the effects of scattering on surface shading (figure 2c) and complex lighting like environment maps. These effects are not included in previous methods because they are difficult to numerically simulate efficiently, requiring an integration over all incident scattered lighting directions at each surface point.

Analytically based methods: The diffusion approximation for optically thick media was applied to subsurface scattering [Stam 1995; Jensen et al. 2001]. An analytic form for the single scattering term was also derived by Hanrahan and Krueger [1993]. However, the problem we are solving is very different from that of subsurface scattering, where the light sources and viewer are outside the medium. In our case, both the sources and viewer are immersed *inside* the medium. Also, unlike in the case of diffusion, we are interested in strongly directional effects like glows around sources.

Analytic expressions for airlight with directional light sources, based on the derivation by Koschmeider [1924], are used frequently for rendering skies [Preetham et al. 1999; Hoffman and Preetham 2003; Narasimhan and Nayar 2002]. However, our focus is different. We wish to derive an analytic model with "near-field" point sources, which is a significantly more complex lighting situation as compared to distant lighting (collimated beams).

Analytic expressions for the glows around point light sources *inside* homogeneous media have also been derived [Max. 1986; Biri et al. 2004; Narasimhan and Nayar 2003]. Therefore, those methods could be used to render glows in real time. However, it is not clear how to extend them to a complete real-time rendering system that also considers the effects of airlight on surface shading, or handles complex environment map lighting. Furthermore, their derivations involve approximations that are not feasible in several common rendering scenarios. For instance, the model derived by Max [1986] does not take into account attenuation. Biri et al. [2004] use a polynomial approximation to single scattering which results in inaccurate glows along viewing directions near the source. The multiple scattering model in [Narasimhan and Nayar 2003] is not strictly valid when objects are present in the medium, especially near the sources (as is generally true in most common scenes), or for optically thin media. Further, the integration required for surface radiance cannot be computed analytically or simulated numerically at interactive rates.

3 The Airlight Model

In this section, we will derive an explicit model for the single scattered radiance at a viewer, due to an isotropic point light source, assuming that both the viewer and the source are immersed in a homogeneous scattering medium. Consider the scenario illustrated in figure 4 (the notations used are indicated in figure 3). The point light source has a radiant intensity I_0 and is at a distance D_{sv} from the view point, making an angle γ with the viewing direction. The radiance, L, is composed of the direct transmission, L_d, and the single scattered radiance or airlight, L_a,

s, v, p	Subscripts for Source, Viewer, surface Point
γ	Angle between light source and viewing ray
D_{sv}	Distance between source and viewer
D_{vp}	Distance between viewer and closest surface point
D_{sp}	Distance between source and surface point
T_{sv}	Optical thickness between source, viewer (βD_{sv})
T_{vp}	Optical thickness between viewer, surface point (βD_{vp})
T_{sp}	Optical thickness between source, surface point (βD_{sp})
β	Scattering coefficient of the participating medium
α	Angle of scattering
x	Distance along the ray from viewer (integration variable)
d	Distance of single scattering from light source
I_0	Radiant intensity of point light source
f_r	BRDF of surface

Figure 3: *Notation used in our derivations.*

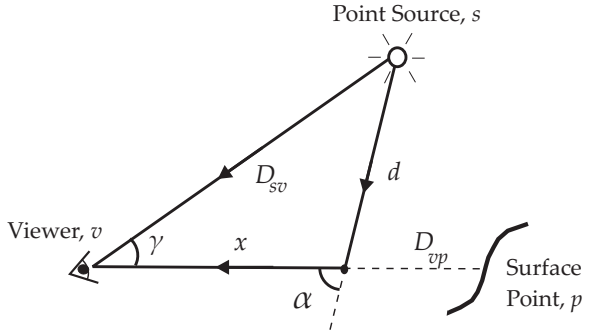


Figure 4: *Diagram showing how light is scattered once and travels from a point light source to the viewer.*

$$L = L_d + L_a. \tag{1}$$

The direct term L_d simply attenuates the incident radiance from a point source (I_0/D_{sv}^2) by an exponential corresponding to the distance between source and viewer, and the scattering coefficient[2] β,

$$L_d(\gamma, D_{sv}, \beta) = \frac{I_0}{D_{sv}^2} e^{-\beta D_{sv}} \cdot \delta(\gamma), \tag{2}$$

where the delta function indicates that for direct transmission, we receive radiance only from the direction of the source (no glows).

3.1 The Airlight Integral

We focus most of our attention on the airlight L_a. The standard expression [Nishita and Nakamae 1987] is given by an integral along the viewing direction,

$$L_a(\gamma, D_{sv}, D_{vp}, \beta) = \int_0^{D_{vp}} \beta k(\alpha) \cdot \frac{I_0 \cdot e^{-\beta d}}{d^2} \cdot e^{-\beta x} dx, \tag{3}$$

where D_{vp} is the distance to the closest surface point along the viewing ray or infinity if there are no objects in that direction, and $k(\alpha)$ is the particle phase function. The exponential attenuation corresponds to the total path length traveled, $d + x$. The two parameters d and angle α in the integrand depend on x. In particular, d is

[2] When there is light absorption in addition to scattering, β is called the extinction coefficient and is given by the sum of the scattering and absorption coefficients. In this paper, we simply refer to β as the scattering coefficient, and it is straightforward to include absorption in our models.

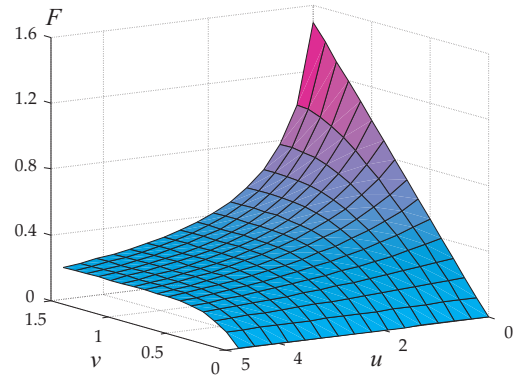


Figure 5: *3D plot of special function $F(u,v)$ in the range of $0 \le u \le 10$ and $0 \le v \le \frac{\pi}{2}$. The plot shows that the function is well-behaved and smooth and can therefore be precomputed as a 2D table. As expected from the definition in equation 10, the function decreases as u increases, and increases as v increases. The maximum value in the plot above therefore occurs at $(u = 0, v = \frac{\pi}{2})$. Also note from equation 10, that for $u = 0$, there is no attenuation so the function is linear in v.*

given by the cosine rule as

$$d = \sqrt{D_{sv}^2 + x^2 - 2xD_{sv}\cos\gamma}. \tag{4}$$

Let us now substitute equation 4 into equation 3. For now, we also assume the phase function $k(\alpha)$ is isotropic and normalized to $1/4\pi$ (our approach can also be generalized to arbitrary phase functions—see appendix D on CDROM). In this case,

$$L_a(\gamma, D_{sv}, D_{vp}, \beta) = \frac{\beta I_0}{4\pi} \int_0^{D_{vp}} \frac{e^{-\beta\sqrt{D_{sv}^2+x^2-2xD_{sv}\cos\gamma}}}{D_{sv}^2 + x^2 - 2xD_{sv}\cos\gamma} \cdot e^{-\beta x} dx. \tag{5}$$

We refer to this equation as the *airlight single scattering integral* and next focus on simplifying it further to derive an explicit form.

3.2 Solution to the Airlight Integral

We take a hybrid approach to solve equation 5. The key result is that this integral can be factorized into two expressions—(a) an **analytic expression** that depends on the physical parameters of the scene and (b) a two-dimensional numerically tabulated function that is **independent** of the physical parameters. Essentially, this factorization enables us to evaluate the integral in equation 5 analytically. A high-level sketch of the derivation is given below and detailed simplifications are included in appendix A.

STEP 1. Reducing the dimensions of the integral: Since the integral in equation 5 depends on 4 parameters, our first step is to apply a series of substitutions that reduce the dependency of the integrand to only one parameter. For this, we first write the expressions in terms of optical thicknesses $T_* = \beta D_*$ and $t = \beta x$. In most cases, this eliminates the separate dependence on both β and the distance parameters, somewhat reducing the complexity, and giving us a simpler intuition regarding the expression's behavior. Then, we combine the dependence on T_{sv} and γ by making the substitution $z = t - T_{sv}\cos\gamma$, to obtain

$$L_a(\gamma, T_{sv}, T_{vp}, \beta) = \frac{\beta^2 I_0}{4\pi} e^{-T_{sv}\cos\gamma} \int_{-T_{sv}\cos\gamma}^{T_{vp}-T_{sv}\cos\gamma} \frac{e^{-z-\sqrt{z^2+T_{sv}^2\sin^2\gamma}}}{T_{sv}^2\sin^2\gamma + z^2} dz. \tag{6}$$

Now, the integrand really depends on only one physical parameter $T_{sv}\sin\gamma$, beginning to make the computation tractable.

It is possible to further simplify equation 6, as described in appendix A. To encapsulate the dependence on the physical parameters of the problem, we define the following two auxiliary expressions, corresponding respectively to the normalization term outside the integrand, and the single physical parameter in the integrand, $T_{sv}\sin\gamma$:

$$A_0(T_{sv}, \gamma, \beta) = \frac{\beta^2 I_0 e^{-T_{sv}\cos\gamma}}{2\pi T_{sv}\sin\gamma} \tag{7}$$

$$A_1(T_{sv}, \gamma) = T_{sv}\sin\gamma. \tag{8}$$

It is then possible to derive, as shown in appendix A, that

$$L_a = A_0(T_{sv}, \gamma, \beta) \int_{\gamma/2}^{\frac{\pi}{4}+\frac{1}{2}\arctan\frac{T_{vp}-T_{sv}\cos\gamma}{T_{sv}\sin\gamma}} \exp[-A_1(T_{sv}, \gamma)\tan\xi]\, d\xi. \tag{9}$$

Although equation 9 might seem complicated, it is really in a simplified form. We already have simple analytic expressions for A_0 and A_1. Further, the function A_1 is a numerical constant as far as the integration is concerned.

STEP 2. Evaluating the integral using a special function: To encapsulate the key concepts in the integrand of equation 9, we define the special function,

$$F(u,v) = \int_0^v \exp[-u\tan\xi]\, d\xi. \tag{10}$$

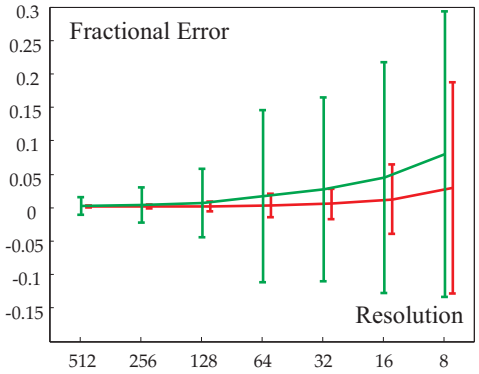


Figure 6: *Accuracy of the airlight model. The plots show the error (versus numerically integrating equation 5) as a function of the resolution for the 2D tables for $F(u,v)$. We report the fractional error, normalizing by the total airlight over the hemisphere. The error for each resolution is averaged over 40000 parameter values of β, D_{sv}, D_{vp} and γ. Bilinear (red) and nearest neighbor (green) interpolation is used to interpolate $F(u,v)$ at non-grid locations of the indices (u,v). The plots clearly indicate the high accuracy of our compact formula, and that a 64×64 table for $F(u,v)$ suffices for a maximum error of less than 2%.*

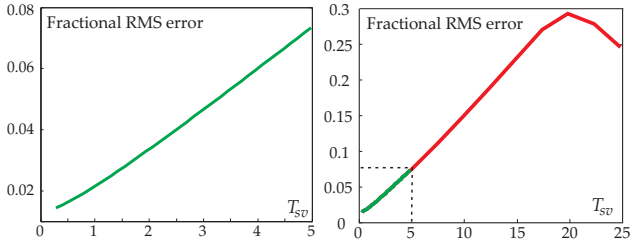


Figure 7: *Comparison of the airlight model with a standard Monte Carlo simulation that includes multiple scattering. The plots show the relative RMS error between the two methods for the case of isotropic phase function. [Left] The low RMS errors show that our model is physically accurate (less than 4% error) for optically thin media ($T_{sv} \le 2$). [Right] From this plot, it is evident that multiple scattering becomes more important as optical thickness increases. However, the actual errors grow slowly and are still low for a wide range of optical thicknesses ($T_{sv} < 10$). It is also interesting to note that for very high optical thicknesses ($T_{sv} > 20$), attenuation dominates over scattering and once again the RMS errors decrease.*

Figure 8: *The images show glows around three identical point light sources (street lamps) at different distances from the viewer. From left to right, we show three different values of the scattering coefficient β ($\beta = 0, 0.01, 0.04$). Larger values of β correspond to larger optical thicknesses T_{sv}. We clearly see the effect of greater glows for larger β. Also, the radiance from farther light sources is attenuated more in each individual image, resulting in smaller glows for the farther lights. In the fourth (rightmost) image, we show a different view with $\beta = 0.04$, where all the light sources are approximately equidistant, with the result that they have similar glows. (The shading on the surfaces is computed using the surface radiance model in section 4.)*

Unfortunately, there exists no simple analytic expression for $F(u,v)$. However, the function is a well behaved 2D function as shown in figure 5. Therefore, we can simply store it numerically as a 2D table. This is really no different from defining functions like sines and cosines in terms of lookup tables. In practice, we will use texture mapping in graphics hardware to access this 2D table. Note that $F(u,v)$ is purely numerical (independent of the physical parameters of the problem), and thus needs to be precomputed only once.

Finally, we can obtain for $L_a(\gamma, T_{sv}, T_{vp}, \beta)$,

$$L_a = A_0 \Big[F(A_1, \frac{\pi}{4} + \frac{1}{2} \arctan \frac{T_{vp} - T_{sv}\cos\gamma}{T_{sv}\sin\gamma}) - F(A_1, \frac{\gamma}{2}) \Big], \quad (11)$$

where we have omitted the parameters for L_a, A_0 and A_1 for brevity.

In the important special case of $T_{vp} = \infty$, corresponding to no objects along the viewing ray, we get $L_a(\gamma, T_{sv}, \infty, \beta)$ as

$$L_a = A_0(T_{sv}, \gamma, \beta) \Big[F(A_1(T_{sv}, \gamma), \frac{\pi}{2}) - F(A_1(T_{sv}, \gamma), \frac{\gamma}{2}) \Big]. \quad (12)$$

In summary, we have reduced the computation of a seemingly complex single scattering integral in equation 5 into a combination of an analytic function computation that depends on the physical parameters of the problem and a lookup of a precomputed 2D smooth function that is independent of the physical parameters of the problem. In the rest of the paper, we will demonstrate several extensions and applications of our model.

3.3 Accuracy of the Airlight Model

We first investigate the accuracy of our analytic model as compared to numerically integrating equation 5. Figure 6 shows plots of the mean error in L_a as a function of the resolution of the 2D numerical table for the special function $F(u,v)$. We use interpolation to evaluate $F(u,v)$ at non-grid locations for the indices (u,v) (bilinear and nearest neighbor interpolations are shown in figure 6). For each resolution, the error computed is averaged over 40000 sets of parameter values for β, D_{sv}, D_{vp}, γ. The error bars in the figure show the standard deviation of the errors. The plots indicate that even a low resolution 64×64 table suffices to compute $F(u,v)$ accurately, with a maximum error of less than 2%. As expected, bilinear interpolation performs better, but, for faster rendering, one can use nearest neighbor interpolation with only a small loss in accuracy.

We also validate the accuracy of the single scattering assumption in our airlight model. Figure 7 shows the relative RMS errors between glows around light sources computed using our model and a standard volumetric Monte Carlo approach that takes into account multiple scattering as well. The Monte Carlo simulation took approximately two hours to compute each glow, whereas our explicit model runs in real-time. The comparison was conducted for optical thicknesses over a wide range $T_{sv} \in (0.25, 25)$ and $T_{vp} \in (0.5, 50)$, which covers almost all real situations. As expected, for optically thin media ($T_{sv} \leq 2$), our model is very accurate (less than 4% relative RMS error). Interestingly, even for greater optical thicknesses ($T_{sv} > 2$), the error only increases slowly. Thus, our single scattering model may be used as a viable approximation for most common real-time rendering scenarios, such as games.

3.4 Visual Effects of the Airlight Model

The dependence of the model on the viewing direction γ and the distance of the source from the observer D_{sv}, predicts visual effects like the glows around light sources and the fading of distant objects. As discussed above, these effects are physically accurate for thin fog (low β and T), and qualitatively reasonable in other cases. In figure 8, we also see how these glows change as a function of the medium properties (the scattering coefficient β) and distance to the sources. As β increases, we go from no glow ($\beta = T = 0$) to a significant glow due to scattering. The differences in the 3 light sources should also be observed. The farther lights are attenuated more, and we perceive this effect in the form of reduced glows around more distant sources. The final (rightmost) image in figure 8 shows a different viewpoint, where the sources are at approximately the same distance, and the glows therefore look the same.

4 The Surface Radiance Model

In this section, we discuss the effects of airlight on the outgoing surface radiance. Consider the illustration in figure 9, where an isotropic point light source s illuminates a surface point p. We will calculate the reflected radiance at the surface. To get the actual appearance at the viewer, we need to attenuate by $\exp[-T_{vp}]$ as usual, where T_{vp} is the optical thickness between viewer and surface point.

The reflected radiance L_p is the sum of contributions, $L_{p,d}$ and $L_{p,a}$, due to direct transmission from the source, and single scattered airlight from the source respectively,

$$L_p = L_{p,d} + L_{p,a}. \quad (13)$$

The direct transmission corresponds to the standard surface reflectance equation, only with an attenuation of $\exp[-T_{sp}]$ added because of the medium, where T_{sp} is the optical thickness between the source and the surface point:

$$L_{p,d} = \frac{I_0 e^{-T_{sp}}}{D_{sp}^2} f_r(\theta_s, \phi_s, \theta_v, \phi_v) \cos\theta_s, \quad (14)$$

where f_r is the BRDF, (θ_s, ϕ_s) is the direction to the source, and therefore also the incident direction, and (θ_v, ϕ_v) is the viewing direction. All angles are measured with respect to the surface normal, in the local coordinate frame of the surface.

4.1 The Surface Radiance Integral

On the other hand, the single-scattered radiance $L_{p,a}$ is more complicated, involving an integral of the airlight (L_a from equation 12) over all incident directions,

$$L_{p,a} = \int_{\Omega_{2\pi}} L_a(\gamma'(\theta_s, \omega_i), T_{sp}, \infty, \beta) f_r(\theta_i, \phi_i, \theta_v, \phi_v) \cos\theta_i \, d\omega_i. \quad (15)$$

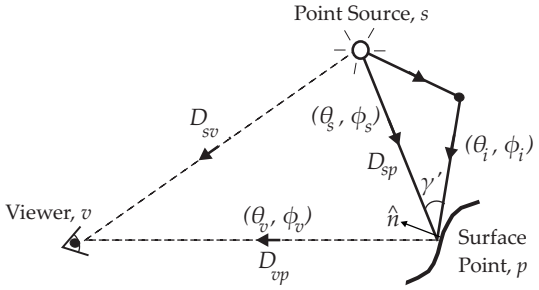


Figure 9: *Diagram showing how light travels from a point light source to a surface point and gets reflected towards the viewer by the surface point.*

Consider the parameters of L_a in the integrand of the above equation. The angle γ' in this case is the angle[3] between the incident direction ω_i and the source direction (θ_s, ϕ_s). Note that for isotropic BRDFs, we can always rotate the coordinate system so $\phi_s = 0$, allowing us to write $\gamma'(\theta_s, \omega_i)$. Finally, L_a also depends on the optical thickness between the source and the surface point T_{sp} (instead of between source and viewer in equation 12).

We refer to equation 15 as the *surface radiance single scattering integral*, analogous to the airlight single scattering integral in equation 5, and next focus on deriving an explicit compact form.

4.2 Solution to the Surface Radiance Integral for Lambertian and Phong BRDFs

First consider the Lambertian case, so the BRDF is a constant k_d. The integral will then depend only on the parameters of L_a, i.e. γ', T_{sp} and β. Of these, the dependency on β is primarily a normalization factor and does not affect the integrand. The angle γ' is a function of the source direction θ_s, and the integration variable ω_i. Hence, the integrand depends on only two physical variables, T_{sp} and θ_s. Thus, as in the previous section, we can define a special two-dimensional numerical function $G_0(T_{sp}, \theta_s)$.

For the Phong BRDF, we employ the reparameterization method in [Ramamoorthi and Hanrahan 2002], measuring angles from the reflection of the viewing direction about the surface normal, rather than the surface normal itself. To indicate this, we denote by θ_s' the angle the source makes with respect to this reflected direction. Upon making this transformation, it can be shown that the Phong BRDF is mathematically analogous to the Lambertian case. To allow for the Phong exponent n, we define the 2D function G_n instead of G_0. These functions are well-defined and smooth as shown by the plots in figure 10. The details of these calculations are in appendix B, and the formula for G_n is

$$G_n(T_{sp}, \theta_s') = \int_{\Omega_{2\pi}} \frac{e^{-T_{sp}\cos\gamma'}}{\sin\gamma'} \Big[F(A_1, \frac{\pi}{2}) - F(A_1, \frac{\gamma'}{2})\Big] \cos^n\theta_i \, d\omega_i, \tag{16}$$

where γ' and $A_1(T_{sp}, \gamma')$ are functions of θ_s' and ω_i, i.e. $\gamma'(\theta_s', \omega_i)$.

The final shading formula, considering both direct transmission and single scattering is then given by (derivation in appendix B):

$$L_p = I_0 k_d \Big[\frac{e^{-T_{sp}}}{D_{sp}^2}\cos\theta_s + \beta^2 \frac{G_0(T_{sp}, \theta_s)}{2\pi T_{sp}}\Big] + \tag{17}$$

$$I_0 k_s \Big[\frac{e^{-T_{sp}}}{D_{sp}^2}\cos^n\theta_s' + \beta^2 \frac{G_n(T_{sp}, \theta_s')}{2\pi T_{sp}}\Big]. \tag{18}$$

As in the airlight model derviation, we have reduced the computation of surface radiance due to single scattering to a few analytic function evaluations and a few 2D table lookups.

[3] We use the prime on γ' to make a technical distinction from the angle γ between the source and viewer used in the previous section. An explicit trigonometric formula for γ' is given in appendix B.

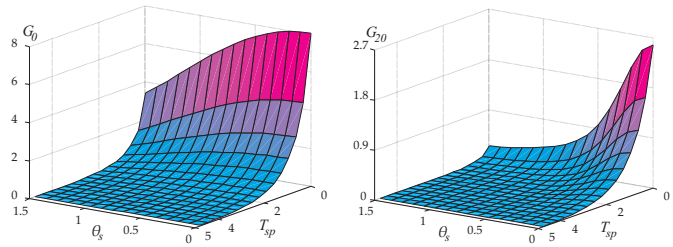
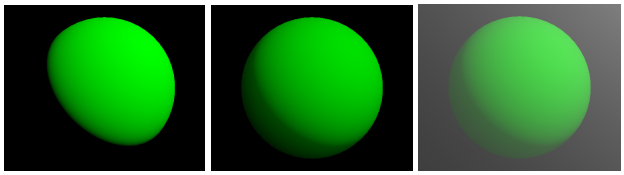
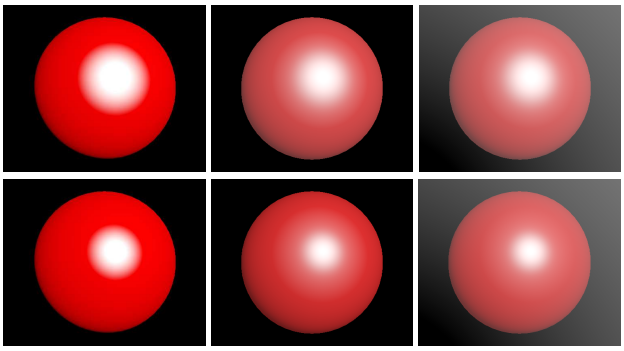


Figure 10: *3D plots of functions G_0 and G_n for $n = 20$ in the range of $0 \le T_{sp} \le 10$ and $0 \le \theta_s \le \frac{\pi}{2}$. The plots show that both functions are well-defined and smooth and can therefore be precomputed as 2D tables. The functions reach their peak values for $\theta_s = T_{sp} = 0$, decaying with increases in both parameters. The decay is faster for the peakier G_{20} on the right.*



(a) (b) (c)

Figure 11: *Influence of scattering on Lambertian surface radiance. In the foggy image (b), created using our surface radiance model, we see a dimming due to attenuation and diffusing of shading (note the brightening of darker areas compared to the clear day image in (a)). These effects are perhaps more apparent in (c), where we also include airlight from the source.*



(a) (b) (c)

Figure 12: *Influence of scattering on specular surface radiance (top row has Phong exponent 10, bottom has Phong exponent 20). In the foggy images (b), we see a dimming and diffusing of the specular highlight compared to the clear-day image in (a). Note also the overall loss in color saturation and contrast, especially in (c). These are important visual effects, usually missing in previous interactive techniques.*

4.3 Visual Effects of the Surface Radiance Model

To illustrate the different qualitative effects we see on surfaces due to scattering, we rendered spheres with Lambertian (figure 11) and Phong BRDFs (figure 12) using our model above. The columns are from left to right (a) no scattering, (b) the effects of scattering on surface shading, and (c) combining this with the airlight effects directly from the source. For the Lambertian sphere in figure 11, we see a dimming due to attenuation of light through the scattering medium, and the diffusing of shading leading to a brightening of darker shadow regions. In the specular case, we see a dimming and diffusing out of the specular highlight due to scattering, combined with an overall reduction in color saturation and contrast. These are important qualitative shading effects that add to the realism of scene appearance in scattering media.

```
frag2app fmain(
    float4 objPos : TEXCOORD3,                    // 2D texture coords
    ...
    uniform samplerRECT F,                        // 2D special functions
    uniform samplerRECT G0,
    uniform samplerRECT Gn)
{
    frag2app OUT;                                 // output radiance
    // Set up and calculate Tsv, γ, Dsv, Tvp, θs and θs'

    /********** Compute La from equation 11 ******/
    A0 = (β * I0 * exp[-Tsv * cos γ])/(2π * Dsv * sin γ); // equation 7
    A1 = Tsv * sin γ;                             // equation 8
    v = π/4 + (1/2) arctan [(Tvp - Tsv * cos γ)/(Tsv * sin γ)];
                                                  // v is one of texture coords
    f1 = texRECT(F, float2(A1, v));               // 2D texture lookup
    f2 = texRECT(F, float2(A1, γ/2));
    airlight = A0 * (f1 - f2);                    // equation 11

    /********** Diffuse surface radiance from equation 17 ******/
    d1 = kd * exp[-Tsp] * cos θs * I0/(Dsp * Dsp);
    d2 = (kd * I0 * β * β)/(2π * Tsp) * texRECT(G0, float2(Tsp, θs));
    diffuse = d1 + d2;

    /********** Specular surface radiance from equation 18 ******/
    s1 = ks * exp[-Tsp] * cos^n θs' * I0/(Dsp * Dsp);
    s2 = (ks * I0 * β * β)/(2π * Tsp) * texRECT(Gn, float2(Tsp, θs'));
    specular = s1 + s2;

    /********** Final Color (equation 19) ******/
    OUT.color = airlight + (diffuse + specular) * exp[-Tvp];
    return  OUT;
}
```

Figure 13: *Pseudocode for the Cg fragment shader that implements our combined model for airlight and surface radiance.*

5 The Complete Model and its Hardware Implementation

While the mathematical derivations in the previous two sections are somewhat involved, the actual implementation is straightforward. Our model provides an explicit form that can be easily implemented in modern programmable graphics hardware. This requires minimal changes to both the original rendering code and scene description, and can therefore be easily integrated into other existing real-time rendering methods. Indeed, the user need only specify the coefficient β of the medium, as in standard OpenGL fog, and use the shader corresponding to our model (pseudocode is in figure 13).

To compute the final appearance, we sum up the attenuated reflected radiance from the surface and the airlight from the source,

$$L = e^{-T_{vp}} L_p + L_a. \tag{19}$$

L_a is the airlight and is given by equation 11. L_p is the exitant radiance at the surface and is given by equations 17 and 18. **We only need to compute a few simple analytic terms and do 4 texture lookups for each vertex or pixel, two for special function F, and one each for G_0 and G_n** (these texture lookups correspond to the texRECT function call in the pseudocode of figure 13). Clearly, these computations can be done by modern programable graphics cards interactively in a single rendering pass.

In practice, we implement the model using Cg in the fragment shader of an NVidia Geforce 6800 graphics card. The special functions F, G_0 and G_n are precomputed and tabulated as 64×64 floating point textures. Since these textures are precomputed only once, we minimize frequent data transfer between the graphics card and main memory.

The rendering speed depends on a variety of variables, and rendering time is linear in the number of light sources. As shown in the video, we are able to achieve real-time rates even for fairly complex scenes with several light sources. As an example, we rendered the scene shown in figure 1, with 39,999 vertices and 66,454 triangles. We simulated the scattering effects from 4 light sources and achieved about 20 fps using the graphics hardware mentioned above. The model for the scene was obtained from http://hdri.cgtechniques.com.

6 Complex BRDFs and Lighting

So far, we have considered arbitrarily located point lights, and simple Lambertian and Phong BRDFs, showing how an explicit expression can be derived and implemented. Rendering time is linear in the number of lights. In this section, we show how these ideas can be extended to efficiently handle complex BRDFs and environment map lighting using convolution, if we are willing to make particular simplifying assumptions. We first introduce the notion of a point-spread function (PSF) for the radiance or glow from a point source due to single scattering. This is similar in spirit to the PSFs derived by Narasimhan and Nayar [2003] and Premoze et al. [2004] in the context of multiple scattering for offline rendering. We will then discuss a number of applications including

- Rendering arbitrary BRDFs with point light sources, by convolving the BRDF with this PSF, as shown in figure 14. This approach can be used if we are willing to precompute a tabular BRDF representation, instead of using a simple explicit formula, as for Lambertian and Phong BRDFs.

- Convolving an environment map with the PSF to efficiently handle very complex lighting (with possibly thousands of lights, corresponding to the pixels of an environment map). This convolution is possible if we assume that all light sources are equally far away, as in a distant environment map. This enables us to obtain the characteristic glows and blurriness around light sources on foggy days, as shown in figure 15.

- Integrating volumetric scattering into precomputed radiance transfer methods that include complex lighting, realistic materials, cast shadows and interreflections (figure 16). The idea of convolving with the point-spread function can be applied to almost any technique that uses environment maps, enabling environment mapping and precomputed radiance transfer to be used in participating media for the first time.

Throughout the section, we apply the signal-processing results of Ramamoorthi and Hanrahan [2001] and Basri and Jacobs [2003] to efficiently compute the convolutions in the frequency domain using spherical harmonics.

6.1 Airlight Point Spread Function (PSF)

In section 3, we determined the radiance $L_a(\gamma, D_{sv}, D_{vp}, \beta)$ from a point source reaching a viewer, due to single scattering. If we fix the distance to the source D_{sv}, the integrating distance D_{vp}, and the scattering coefficient β of the medium, the radiance becomes a function only of the angle γ. We normalize this function by I_0/D_{sv}^2 to account for the intensity of the source, and define the PSF as

$$\mathrm{PSF}(\gamma)_{D_{sv}, D_{vp}, \beta} = \frac{D_{sv}^2 L(\gamma, D_{sv}, D_{vp}, \beta)}{I_0}. \tag{20}$$

Since the PSF is mostly applied for surface shading, we will generally set $D_{vp} = \infty$, as in section 4.

6.2 Empirical PSF factorization for Speedup

The PSF defined above still depends on the parameters of the medium such as the coefficient β. So, changing these parameters changes the PSF and requires us to redo any convolutions. However, we have observed empirically that the PSF above can be factored into **a purely angular component that is independent of the medium parameters and an amplitude component that depends on the medium parameters. This factorization enables us**

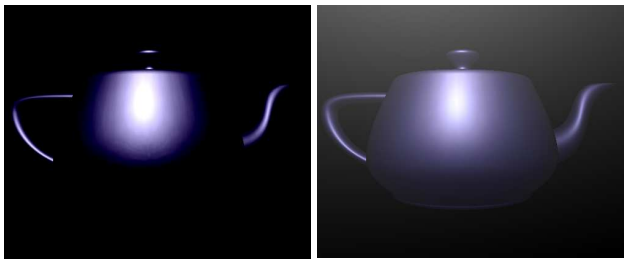


Figure 14: *[Left] A teapot rendered using the measured blue metallic BRDF. [Right] The teapot as it appears in a scattering medium. The brightening of darker regions, and softening of shading, is clearly visible.*

to change the medium parameters interactively without having to re-compute the PSF or redo any convolutions. Specifically,

$$\mathrm{NPSF}(\gamma) = \frac{F(\sin\gamma, \frac{\pi}{2}) - F(\sin\gamma, \frac{\gamma}{2})}{2\pi \sin\gamma \cdot e^{(\cos\gamma - 1)}} \quad (21)$$

$$\mathrm{PSF}(\gamma)_{D_{sv},\infty,\beta} = T_{sv} e^{-T_{sv}} \cdot \mathrm{NPSF}(\gamma), \quad (22)$$

where NPSF has only angular dependence, independent of other physical parameters. In appendix C, we derive and validate this approximation, showing plots that indicate there is little noticeable numerical error.

6.3 Rendering with arbitrary BRDFs

We can use convolution with the PSF to render with arbitrary tabulated BRDFs, such as measured reflectance. For each outgoing direction, we tabulate the BRDF as a function over the sphere of incident directions. A new effective BRDF can be obtained for that outgoing direction by convolving this function with the PSF.

Mathematically, we first write the (isotropic) BRDF in terms of spherical harmonic coefficients for each outgoing angle as

$$\rho^{\mathrm{orig}}(\theta_i, \theta_o, \phi) = \sum_{l,m} \rho^{\mathrm{orig}}_{lm}(\theta_o) Y_{lm}(\theta_i, \phi), \quad (23)$$

where $\rho^{\mathrm{orig}}_{lm}$ are the coefficients, and Y_{lm} is the spherical harmonic. To perform the convolution [Ramamoorthi and Hanrahan 2001], we multiply the coefficients $\rho^{\mathrm{orig}}_{lm}$ of the original BRDF by the corresponding coefficients of the point-spread function[4] PSF_l,

$$\rho^{\mathrm{eff}}_{lm}(\theta_o) = \sqrt{\frac{4\pi}{2l+1}} \mathrm{PSF}_l \rho^{\mathrm{orig}}_{lm}(\theta_o). \quad (24)$$

Then, we can use the effective BRDF to compute the reflected radiance due to airlight, and the original BRDF for the reflected radiance due to direct transmission. Thus, standard rendering algorithms can be executed with only slight modification and at virtually no additional cost. Note however, that while our previous formulae for Lambertian and Phong models required no precomputations, the convolution approach requires precomputation of the spherical harmonic coefficients for a collection of outgoing angles.

Figure 14 shows images rendered with the Blue metallic BRDF measured by Matusik et al. [2003]. In the left image, we simply render a tabular description of the BRDF without scattering. In the right image, we use the formula above to compute a new effective tabulated BRDF, including the effects of airlight. The brightening of darker regions owing to scattering is clearly visible on the right.

6.4 Rendering with Environment Maps

Our point spread function can be applied directly to environment maps, with the effects of scattering obtained by convolving the environment map with the PSF. To use a single PSF for all sources, we must assume that the lighting is made of small equidistant light sources (fixed D_{sv}). This is a good approximation when the size of the objects is small compared to the distance to the environment[5].

[4] Since the PSF is radially symmetric, depending only on γ, only spherical harmonic coefficients with m = 0 are nonzero.

Figure 15: *[Top] Grace cathedral environment map with no scattering. [Middle] The environment map is convolved with the airlight single scattering PSF to create a foggy/misty appearance. Notice the glows around the light sources, the blurring of the sources and the brightening of dark areas. [Bottom] A scene illuminated by the environment map without scattering (left) and with scattering (right). Notice the spreading of the highlights and brightening of the spheres due to scattering.*

We first consider simply looking at the environment, where we would like to see the glows around the bright light sources, to create the effects of foggy or misty appearance. To achieve this effect, we simply need to convolve the environment map with the PSF,

$$L^{convolved}_{lm} = \sqrt{\frac{4\pi}{2l+1}} PSF_l L^{original}_{lm}. \quad (25)$$

Furthermore, similar to equation 1, we can simply use a combination of the original attenuated environment map $L^{attenuated}$ (for direct transmission, and corresponds to L_d in equation 1) and the convolved version $L^{convolved}$ above (for airlight, and corresponds to L_a in equation 1) to compute the surface shading,

$$L^{final} = L^{attenuated} + L^{convolved} \quad (26)$$

$$L^{attenuated} = L^{original} e^{-T_{sv}}. \quad (27)$$

Figure 15 shows results obtained by convolving the Grace Cathedral environment map [Debevec 1998] with the single scattering PSF. The blurring of light sources and the overall increase in brightness in dark regions can be immediately seen. Below that, we compare the appearances of spheres rendered illuminated by this environment map with and without scattering. Notice the spreading of highlights and the brightening of the objects.

[5] Note that while this assumption is similar to standard environment mapping, our PSF requires us to specify a finite (but possibly large) D_{sv}.

Figure 16: *[Left] A scene rendered using precomputed radiance transport, to capture the complex shadows and specular surface BRDFs. [Middle] The same scene rendered as though it was immersed in a scattering medium, with the observer close to the object. Notice the blurring and fogging of the environment in the background. In the insets, we see a number of shading changes, such as the brightening of dark regions in the face because of the scattering of light, the attenuation due to dimming and diffusing of specular highlights on the base, and the softening of shadows on the plane. [Right] The same scene including effects of airlight between viewer and object (as if the viewer were far away, seeing the Buddha through fog). Note the loss in contrast and saturation.*

6.5 Precomputed Radiance Transfer

The traditional environment map rendering techniques do not take shadows or interreflections into account. Precomputed radiance transport methods [Sloan et al. 2002] compute the visibility in an off-line manner, followed by interactive rendering. To add participating media, we only need to convolve the lighting (environment map) with our PSF and use the result as input to the existing precomputed radiance transfer techniques. To demonstrate this, we used the technique of [Wang et al. 2004; Liu et al. 2004], which handles non-diffuse objects under all-frequency environment illumination using a separable BRDF approximation.

We show the result of the Happy Buddha model rendered with the Ashikhmin-Shirley BRDF [2000] in figure 16. The left image is the standard result with no scattering. In the middle image, we show a view of the Buddha, where we include the effect of airlight from the environment on surface appearance, but there is no attenuation or scattering between the viewer and object itself (as if the observer were very close to the Buddha). We clearly see the foggy appearance of the background and the glows or airlight due to the light sources. On the face of the Buddha, we see a brightening of darker regions, along with a dimming and diffusing of specular highlights. A similar effect is seen on the base, where the dimming and diffusing of highlights reduces saturation and contrast. Finally, the shadows on the plane are blurred out, with a considerable softening and loss of detail. In the right image, there is also scattering or airlight between the object and the viewer (as if the observer were far away and seeing the Buddha through fog). This leads to a further loss of detail and contrast, so that the original glossy appearance of the object is essentially lost.

7 Conclusions and Future Work

We have presented a simple method to add the effects of participating media to interactive applications. Our approach can be easily implemented in programmable graphics hardware and leads to a number of new effects in the real-time domain, such as interactive rendering with glows around light sources, the effects of scattering on surface shading, environment maps, and precomputed light transport. The key insight is a new analytic model for integrating the light transport equations assuming single scattering, which can also be extended to predict the impact of scattering or airlight on the inherent appearance of surfaces.

More broadly, this paper indicates the power of using explicit formulae to simulate difficult effects like volumetric scattering, speeding up such a process by many orders of magnitude. We do sacrifice some generality, considering only isotropic point light sources, single scattering, homogeneous media, and excluding most cast and volumetric shadowing, but believe this is a worthwhile tradeoff to enable a simple technique that achieves real-time rates.

Future work can follow many avenues. For instance, we can attempt to extend our theoretical model to consider non-isotropic light sources (like spotlights) and inhomogeneous media. Our preliminary work in this area indicates that some of these generalizations, while relatively simple in standard surface calculations, are rather non-trivial for volumetric media. However, we believe that many common cases can be addressed by extensions of the basic methods proposed in this paper. In general, we believe that analytic models of difficult to simulate volumetric phenomena are critical to achieving efficient renderings for real-time applications.

Acknowledgements: We thank Rui Wang, John Tran and David Luebke for the precomputed radiance transfer code, and Simon Premoze for the Monte Carlo simulation code. This research was supported by a Columbia University Presidential Fellowhip, an ONR Grant #N00014-05-1-0188 (A Physical Approach to Underwater Optical Imaging), an NSF Grant #0305322 (Real-Time Rendering and Visualization of Complex Scenes), an NSF CAREER award #0446916 (Mathematical and Computational Fundamentals of Visual Appearance), and equipment donations from Intel and NVIDIA.

References

ASHIKHMIN, M., AND SHIRLEY, P. 2000. An anisotropic phong model. *Journal of Graphics Tools 5*, 2, 25–32.

BASRI, R., AND JACOBS, D. W. 2003. Lambertian reflectance and linear subspaces. *IEEE Trans. Pattern Anal. Mach. Intell. 25*, 2, 218–233.

BIRI, V., MICHELIN, S., AND ARQUES, D. 2004. Real-time single scattering with shadows. In *In review http://igm.univ-mlv.fr/˜biri/indexCA_en.html*.

BLINN, J. 1982. Light reflection functions for simulation of clouds and dusty surfaces. In *SIGGRAPH 82*, 21–29.

CHANDRASEKHAR, S. 1960. *Radiative Transfer*. Oxford Univ. Press.

DEBEVEC, P. 1998. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *SIGGRAPH 98*, 189–198.

DOBASHI, Y., YAMAMOTO, T., AND NISHITA, T. 2002. Interactive rendering of atmospheric scattering effects using graphics hardware. In *Graphics Hardware Workshop 02*, 99–109.

HANRAHAN, P., AND KRUEGER, W. 1993. Reflection from layered surfaces due to subsurface scattering. In *SIGGRAPH 93*, 165–174.

HARRIS, M., AND LASTRA, A. 2001. Real-time cloud rendering. In *Eurographics 2001*, 76–84.

HOFFMAN, N., AND PREETHAM, A. J. 2003. Real-time light-atmosphere interactions for outdoor scenes. *Graphics programming methods*, 337–352.

JENSEN, H., MARSCHNER, S., LEVOY, M., AND HANRAHAN, P. 2001. A practical model for subsurface light transport. In *SIGGRAPH 01*, 511–518.

JENSEN, H. W. 2001. *Realistic Image Synthesis Using Photon Mapping*. AK Peters.

KAJIYA, J., AND HERZEN, B. 1984. Ray tracing volume densities. In *SIGGRAPH 84*, 165–174.

KOSCHMEIDER, H. 1924. Theorie der horizontalen sichtweite. *Beitr. Phys. freien Atm., 12*.

LIU, X., SLOAN, P.-P. J., SHUM, H.-Y., AND SNYDER, J. 2004. All-frequency precomputed radiance transfer for glossy objects. In *EuroGraphics Symposium on Rendering 04*, 337–344.

MATUSIK, W., PFISTER, H., BRAND, M., AND MCMILLAN, L. 2003. A data-driven reflectance model. *ACM Transactions on Graphics (SIGGRAPH 03) 22*, 3, 759–769.

MAX., N. L. 1986. Atmospheric illumination and shadows. In *SIGGRAPH 86*, 117–124.

MAX, N. 1994. Efficient light propagation for multiple anisotropic volume scattering. In *Eurographics Rendering Workshop 94*, 87–104.

NAKAMAE, E., KANEDA, K., OKAMOTO, T., AND NISHITA, T. 1990. A lighting model aiming at drive simulators. In *SIGGRAPH 90*, 395–404.

NARASIMHAN, S., AND NAYAR, S. 2002. Vision and the atmosphere. *IJCV 48*, 3 (August), 233–254.

NARASIMHAN, S., AND NAYAR, S. 2003. Shedding light on the weather. In *CVPR 03*, 665–672.

NISHITA, T., AND NAKAMAE, E. 1987. A shading model for atmospheric scattering considering luminous intensity distribution of light sources. In *SIGGRAPH*, 303 – 310.

PATTANAIK, S., AND MUDUR, S. 1993. Computation of global illumination in a participating medium by monte carlo simulation. *Journal of Visualization and Computer Animation 4*, 3, 133–152.

PREETHAM, A. J., SHIRLEY, P., AND SMITS, B. 1999. A practical analytic model for daylight. In *SIGGRAPH*, 91–100.

PREMOZE, S., ASHIKHMIN, M., TESENDORF, J., RAMAMOORTHI, R., AND NAYAR, S. 2004. Practical rendering of multiple scattering effects in participating media. In *EuroGraphics Symposium on Rendering 04*, 363–374.

RAMAMOORTHI, R., AND HANRAHAN, P. 2001. A signal-processing framework for inverse rendering. In *SIGGRAPH 01*, 117–128.

RAMAMOORTHI, R., AND HANRAHAN, P. 2002. Frequency space environment map rendering. *ACM Transactions on Graphics (SIGGRAPH 02) 21*, 3, 517–526.

RILEY, K., EBERT, D., KRAUS, M., TESSENDORF, J., AND HANSEN, C. 2004. Efficient rendering of atmospheric phenomena. In *EuroGraphics Symposium on Rendering 2004*, 375–386.

RUSHMEIER, H., AND TORRANCE, K. 1987. The zonal method for calculating light intensities in the presence of a participating medium. In *SIGGRAPH 87*, 293–302.

SAKAS, G. 1990. Fast rendering of arbitrary distributed volume densities. In *Eurographics 90*, 519–530.

SLOAN, P.-P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics (SIGGRAPH 02) 21*, 3, 527–536.

STAM, J. 1995. Multiple scattering as a diffusion process. In *Eurographics Rendering Workshop 95*, 41–50.

WANG, R., TRAN, J., AND LUEBKE, D. 2004. All-frequency relighting of non-diffuse objects using separable BRDF approximation. In *EuroGraphics Symposium on Rendering 2004*, 345–354.

Appendix A:Solution to Airlight Integral

We start the derivation from equation 5.

$$L_a = \frac{\beta I_0}{4\pi}\int_0^{D_{vp}} \frac{e^{-\beta\sqrt{D_{sv}^2+x^2-2xD_{sv}\cos\gamma}}}{D_{sv}^2+x^2-2xD_{sv}\cos\gamma}\cdot e^{-\beta x}dx \tag{28}$$

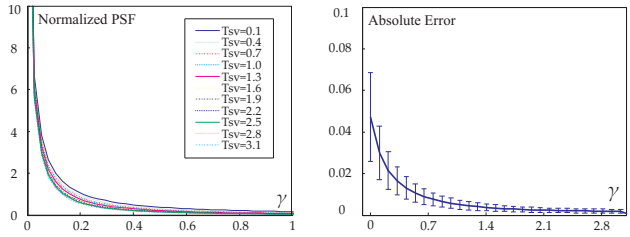


Figure 17: *[Left] Plot of PSF$(\gamma)_{D_{sv},\infty,\beta}$ normalized by $T_{sv}e^{-T_{sv}}$ for different optical thicknesses T_{sv} ranging from 0.1 to 3.1. After normalization, the PSF depends onγ and is largely independent of T_{sv}. This implies that we can factor it into a purely angular component and an amplitude component depending on T_{sv}. [Right] The average and standard deviation of the absolute error of the empirical PSF. While the error increases for smaller angles γ, it remains well below 0.05.*

$$\begin{aligned}
&\text{——>substitute } T_* = \beta D_* \text{ and } t = \beta x \\
&= \frac{\beta^2 I_0}{4\pi}\int_0^{T_{vp}} \frac{e^{-\sqrt{T_{sv}^2+t^2-2tT_{sv}\cos\gamma}}}{T_{sv}^2+t^2-2tT_{sv}\cos\gamma}\cdot e^{-t}dt \qquad (29)\\
&\text{——>substitute } z = t - T_{sv}\cos\gamma \\
&= \frac{\beta^2 I_0 e^{-T_{sv}\cos\gamma}}{4\pi}\int_{-T_{sv}\cos\gamma}^{T_{vp}-T_{sv}\cos\gamma} \frac{e^{-\sqrt{z^2+T_{sv}^2\sin^2\gamma}}}{z^2+T_{sv}^2\sin^2\gamma}\cdot e^{-z}dz \qquad (30)\\
&\text{——>substitute } z = T_{sv}\sin\gamma\tan\eta \\
&= \frac{\beta^2 I_0 e^{-T_{sv}\cos\gamma}}{4\pi T_{sv}\sin\gamma}\int_{\gamma-\frac{\pi}{2}}^{\arctan\frac{T_{vp}-T_{sv}\cos\gamma}{T_{sv}\sin\gamma}} e^{-T_{sv}\sin\gamma\frac{1+\sin\eta}{\cos\eta}}d\eta \qquad (31)\\
&\text{——>substitute } \eta = 2\xi - \tfrac{\pi}{2} \\
&= \frac{\beta^2 I_0 e^{-T_{sv}\cos\gamma}}{2\pi T_{sv}\sin\gamma}\int_{\gamma/2}^{\frac{\pi}{4}+\frac{1}{2}\arctan\frac{T_{vp}-T_{sv}\cos\gamma}{T_{sv}\sin\gamma}} \exp[-T_{sv}\sin\gamma\tan\xi]d\xi, \qquad (32)
\end{aligned}$$

from which we obtain equation 9.

Appendix B: Formula for Lambertian and Phong BRDFs

Here, we derive the expression for Lambertian and Phong BRDFs. We first consider the Lambertian BRDF, beginning with equation 15, Note that in the derivation below, γ' is given from trigonometry by $\gamma'(\theta_s,\omega_i) = \cos\theta_i\cos\theta_s + \sin\theta_i\sin\theta_s\cos\phi_i$.

$$\begin{aligned}
L_{p,a} &= \int_{\Omega_{2\pi}} L_a(\gamma'(\theta_s,\omega_i),T_{sp},\infty,\beta) f_r(\theta_i,\phi_i,\theta_v,\phi_v)\cos\theta_i\, d\omega_i \qquad (33)\\
&\text{——>substitute equation 12 for } L_a \text{ and a constant } k_d \text{ for } f_r \\
&= \int_{\Omega_{2\pi}} A_0(T_{sp},\gamma,\beta)\Big[F(A_1(T_{sp},\gamma'),\frac{\pi}{2}) - F(A_1(T_{sp},\gamma'),\frac{\gamma'}{2})\Big]k_d\cos\theta_i\, d\omega_i \qquad (34)\\
&\text{——>substitute equation 7 for } A_0 \text{ and take constants out of integration} \\
&= \frac{\beta^2 I_0 k_d}{2\pi T_{sp}}\int_{\Omega_{2\pi}} \frac{e^{-T_{sp}\cos\gamma'}}{\sin\gamma'}\Big[F(A_1(T_{sp},\gamma'),\frac{\pi}{2}) - F(A_1(T_{sp},\gamma'),\frac{\gamma'}{2})\Big]\cos\theta_i\, d\omega_i \\
&= \frac{\beta^2 I_0 k_d}{2\pi T_{sp}} G_0(T_{sp},\theta_s). \qquad (35)
\end{aligned}$$

For the Phong BRDF after reparameterization, instead of $k_d\cos\theta_i$, we will obtain $k_s\cos^n\theta_i$, where n is the Phong exponent. This can be handled exactly as above, simply replacing G_0 with G_n.

Appendix C:Empirical PSF factorization

The empirical PSF factorization is inspired by the observation that after being normalized by $T_{sv}e^{-Tsv}$, the PSF becomes essentially independent of the medium physical parameters (optical thickness) and largely depends on angle γ as shown in figure 17 (left). This implies we can factor the PSF into a purely angular component and an amplitude component that depends on the medium parameters. We define the angular component NPSF(γ) as the PSF$(\gamma)_{T_{sv}=1}$ normalized by $T_{sv}e^{-Tsv}$ and define the amplitude component as the normalization factor $T_{sv}e^{-T_{sv}}$. Then, the PSF can be expressed using these two terms as in equation 22. The absolute approximation error is plotted in figure 17 (right) for 11 different optical thickness ranging from 0.1 to 3.1.

Appendix D: Extension to General Phase Functions It is well known that phase functions $k(\alpha)$ of most media can be written as polynomials of $\cos\alpha$ [**?**; Chandrasekhar 1960],

$$k(\alpha) = \sum_{k=0}^{n} W_k \cos^k \alpha\,, \tag{36}$$

where W_k's are coefficients of the polynomial. Then, we get a sum of integrations, each of which resembles equation 5:

$$L_a(\gamma, D_{sv}, D_{vp}, \beta) \quad = \quad \sum_{k=0}^{n} W_k \int_0^{D_{vp}} \beta \cos^k \alpha \cdot \frac{I_0 \cdot e^{-\beta d}}{d^2} \cdot e^{-\beta x} dx\,, \tag{37}$$

where, the angle α is given from trigonometry by $\alpha = \pi - \cos^{-1} \frac{x - D_{sv}\cos\gamma}{\sqrt{D_{sv}^2 + x^2 - 2xD_{sv}\cos\gamma}}$. Thus, for each term $L_{a,n}$, we apply essentially the same steps as in Appendix A through equations 28 to 32:

$$L_{a,n} \quad = \quad \frac{(-1)^k \beta I_0}{4\pi} \int_0^{D_{vp}} \cos^k \alpha \frac{e^{-\beta\sqrt{D_{sv}^2 + x^2 - 2xD_{sv}\cos\gamma}}}{D_{sv}^2 + x^2 - 2xD_{sv}\cos\gamma} \cdot e^{-\beta x} dx \tag{38}$$

——>substitute for α

$$= \quad \frac{(-1)^k \beta I_0}{4\pi} \int_0^{D_{vp}} \frac{(x - D_{sv}\cos\gamma)^k e^{-\beta\sqrt{D_{sv}^2 + x^2 - 2xD_{sv}\cos\gamma}}}{(D_{sv}^2 + x^2 - 2xD_{sv}\cos\gamma)^{1+k/2}} \cdot e^{-\beta x} dx \tag{39}$$

——>substitute $T_* = \beta D_*$ and $t = \beta x$

$$= \quad \frac{(-1)^k \beta^2 I_0}{4\pi} \int_0^{T_{vp}} \frac{(t - T_{sv}\cos\gamma)^k e^{-\sqrt{T_{sv}^2 + t^2 - 2tT_{sv}\cos\gamma}}}{(T_{sv}^2 + t^2 - 2tT_{sv}\cos\gamma)^{1+k/2}} \cdot e^{-t} dt \tag{40}$$

——>replace $t - T_{sv}\cos\gamma$ with z

$$= \quad \frac{(-1)^k \beta^2 I_0 e^{-T_{sv}\cos\gamma}}{4\pi} \int_{-T_{sv}\cos\gamma}^{T_{vp} - T_{sv}\cos\gamma} \frac{z^k e^{-\sqrt{z^2 + T_{sv}^2 \sin^2\gamma}}}{(z^2 + T_{sv}^2 \sin^2\gamma)^{1+k/2}} \cdot e^{-z} dz \tag{41}$$

——>substitute $z = T_{sv}\sin\gamma\tan\eta$

$$= \quad \frac{(-1)^k \beta^2 I_0 e^{-T_{sv}\cos\gamma}}{4\pi T_{sv}\sin\gamma} \int_{\gamma - \frac{\pi}{2}}^{\arctan\frac{T_{vp} - T_{sv}\cos\gamma}{T_{sv}\sin\gamma}} \sin^k \eta\, e^{-T_{sv}\sin\gamma\frac{1+\sin\eta}{\cos\eta}}\, d\eta \tag{42}$$

——>substitute $\eta = 2\xi - \frac{\pi}{2}$

$$= \quad \frac{\beta^2 I_0 e^{-T_{sv}\cos\gamma}}{2\pi T_{sv}\sin\gamma} \int_{\gamma/2}^{\frac{\pi}{4} + \frac{1}{2}\arctan\frac{T_{vp} - T_{sv}\cos\gamma}{T_{sv}\sin\gamma}} \cos^k 2\xi \cdot \exp[-T_{sv}\sin\gamma\tan\xi] d\xi\,. \tag{43}$$

Then, as before, the expression for general phase functions can be written using the special function $F^k(u,v)$ (analogous to $F(u,v)$),

$$F^k(u,v) = \int_0^v \cos^k 2\xi \exp[-u\tan\xi]\, d\xi\,. \tag{44}$$

Thus, if n terms of the phase function are used, we need to store n 2D tables. In practice, since a small value for n (typically $n \leq 3$) typically suffices to represent most phase functions, only a few 2D tables need to be stored.