

Efficient Rendering of Atmospheric Phenomena

Kirk Riley[†], David S. Ebert[†], Martin Kraus[†], Jerry Tessendorf[‡], and Charles Hansen[§]

Abstract

Rendering of atmospheric bodies involves modeling the complex interaction of light throughout the highly scattering medium of water and air particles. Scattering by these particles creates many well-known atmospheric optical phenomena including rainbows, halos, the corona, and the glory. Unfortunately, most radiative transport approximations in computer graphics are ill-suited to render complex angularly dependent effects in the presence of multiple scattering at reasonable frame rates.

Therefore, this paper introduces a multiple-model lighting system that efficiently captures these essential atmospheric effects. We have solved the rendering of fine angularly dependent effects in the presence of multiple scattering by designing a lighting approximation based upon multiple scattering phase functions. This model captures gradual blurring of chromatic atmospheric optical phenomena by handling the gradual angular spreading of the sunlight as it experiences multiple scattering events with anisotropic scattering particles. It has been designed to take advantage of modern graphics hardware; thus, it is capable of rendering these effects at near interactive frame rates.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism Color, shading, shadowing, and texture

1. Introduction

Particles in the atmosphere are responsible for a wide range of optical phenomena. Wavelength-dependent effects, such as the arcs of rainbows, cirrus halos, and the glory, are just a few [Min93]. These effects add beauty and realism to outdoor scenes; thus, the capability to render these effects accurately is an important component of an outdoor renderer.

Unfortunately, physically accurate radiative transfer is a notoriously difficult problem because the number of possible paths photons may take through a cloud volume grows exponentially with the expected number of scattering events. Multiple scattering in a highly scattering medium, such as a cloud, has long been recognized [KVH84] as an essential lighting component. Multiple scattering blurs source light both angularly and spatially. Thus, the process of multiple scattering gradually blurs out fine angularly dependent effects from the phase function, as the light distribution becomes more isotropic traversing the cloud. Modeling the angular distribution of light as it interacts with anisotropic scat-

tering particles is a difficult problem and, therefore, it is often over-simplified, or even ignored, in most graphics applications.

An accurate lighting model in clouds must produce both single-scattering effects, as well as diffusive illumination. Unfortunately, lighting models either cannot handle both, or require an arbitrarily long time to do so. We have solved this problem through the use of multiple scattering phase functions to model the gradual loss of angular coherence in the light beam. Our algorithm takes full advantage of the power and flexibility offered by modern programmable graphics hardware. Therefore, this system efficiently renders single and multiple scattering atmospheric effects with aerial perspective, while maintaining reasonable rendering times.

In Section 2, we provide the background for rendering atmospheric data and skylight. In Section 3, we describe the atmospheric particle data we are rendering. Next, in Section 4, we discuss the rendering and lighting model used in our system. Some implementation details are given in Section 5. Then, in Section 6, we present our results. Finally, conclusions and future work are given in Section 7 and Section 8 respectively.

[†] Purdue University

[‡] Rhythm and Hues

[§] University of Utah

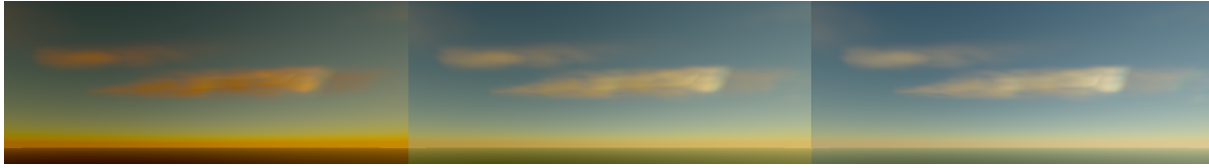


Figure 1: *Cirrus sunrise time sequence*

2. Previous Work

Rendering of outdoor and highly diffusive media has been an active area in computer graphics for more than ten years. The single-scattering low-albedo model [Bli82] provides the framework for rendering particle fields. Early work in modeling multiple scattering involved spherical harmonics [KVH84] and volumetric radiosity solutions [RT87]. More recent diffusion systems model light with a first order angular distribution using a multi-grid scheme [Sta95], with multiple order scattering reference patterns [NDN96], and with multiple anisotropic scattering [MSM*04]. The subsurface light transport system [JMLH01] calculates a diffusion approximation by applying the dipole method to meet boundary conditions at the surface. This method is well suited for reasonably flat boundaries between two homogeneous regions but is less suited for a volumetrically varying inhomogeneous volume, such as multiple-particle clouds. Additionally, diffusion approximations tend to blur fine angularly dependent effects quickly, and are less suited to rendering fine angularly dependent effects in volumes, than the technique presented here. This is particularly evident in the case of larger particles, where, for unity albedo, it may take more than 25 scattering events before the distribution is essentially isotropic. Discrete ordinates [Max94], photon mapping [JC98], and path integral methods [PAS03, Tes87] can generate high accuracy solutions but are computationally expensive when fine angularly dependent effects are desired. Volume rendering [DCH88] has been widely applied in computer graphics and has been extended to gaseous volume rendering [SF95, EP90, REHL03].

Hardware-accelerated techniques in atmospheric modeling and rendering include imposter systems [HL01], cloud formation modeling systems [HBSL03], and intuitive modeling systems [SSEH03]. The small angle approximation [KPH*03, KPHE02], which has been applied to atmospheric rendering [REHL03], is efficient but discards the phase effects after the first scattering event. These hardware-accelerated systems have increased the efficiency of atmospheric rendering systems but have not achieved the capability to render fine angularly dependent effects.

Advanced phase function effects are an essential component of an atmospheric renderer. Rendering rainbows with Descartes' ray tracing method [Mus89] for rain drop phase function calculation has been presented. Complex phase functions and rendering advanced atmospheric optical ef-

fects have been achieved for a single-scattering lighting model [JW97]. Halos and parhelia have also been discussed for ray-tracing [JW98]. While advanced phase functions have been considered in computer graphics, they have not been used in an interactive system that models the gradual angular spreading of the source light.

These atmospheric rendering options, while useful, make significant simplifications to the light's angular distribution, the phase function, or both. A common approximation is to estimate scattering phase based on a function of the first moment of the phase function and the dot product of the incident and exit directions [Bli82, SF95, MSM*04, Sta95, NSTN93], such as the Henyey-Greenstein function. This results in an oversimplification of the angular distribution of the light that eliminates fine angular dependent effects. Here, we present a system that is capable of modeling both a complex lighting distribution and a complex phase function that can capture the effects of both the angular spreading and diffusion of light, and the optical effects of advanced phase functions.

Rendering the sky is another essential element of rendering outdoor scenes. The two density layer scheme [Kla87] provided the ground work for rendering the sky. The basis function method [NSTN93] is very efficient but is less suited for the addition of interacting media. An efficient physically-based sky light approximation [PSS99] has been presented for ground observers. A physically accurate system for modeling the skylight at night has also been described [JDS*01]. Numerical integration techniques [EMP*03] also provide accurate solutions to generating aerial perspective and skylight. While these schemes are very useful, we propose a hardware-accelerated method of rendering the sky that allows us to use an analytical expression and, thus, achieve a sky that interacts with the atmospheric data at reasonable rendering speeds.

3. Atmospheric Data

The atmosphere is an inhomogeneous mixture of a wide variety of particles. In order to develop a system for the accurate rendering of these particles, it is essential to first specify the particles' properties as described in Section 3.1. We then translate these properties into the spatially varying scattering probabilities and phase functions required to volumetrically render the particle fields in Section 3.2.

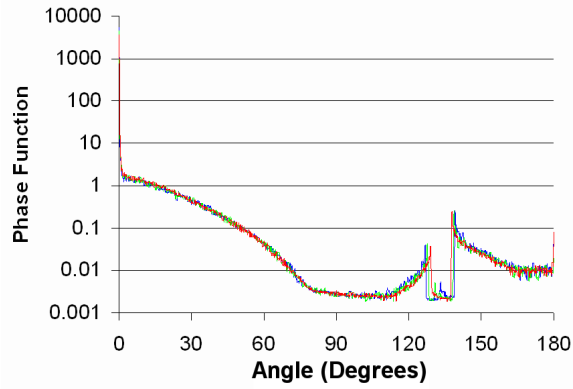


Figure 2: Rain Mie phase function for red, green, and blue wavelengths

3.1. Particle Properties

We assume atmospheric bodies primarily consist of two types of high-albedo particles and that dirt and other particle types have a minimal contribution. We also assume that the renderer will only consider wavelengths in the visible spectrum (between 400 and 700 nm), where these particle properties are accurate.

The first particle type we consider consists of air molecules characterized by highly wavelength-dependent scattering probabilities, yet comparatively isotropic Rayleigh scattering and a wavelength-independent phase function. The second type consists of larger ice and water particles with nearly wavelength-independent scattering probabilities and strongly forward peaked wavelength-dependent phase functions. Specifically, the water particles we consider are classified as cloud (small spherical water), rain (large spherical water), ice (hexagonal ice cylinders) and snow (random ice crystals), but the system does not currently render haze, fog, or more complex ice particles such as graupel. Because scattering behavior is vastly different between the two types of particles, we use different lighting models for each. The scattering from atmospheric particles are characterized by wavelength-dependent parameters including extinction cross-section ($\sigma_{\text{ex}}^\lambda$), the single-scattering albedo (α^λ), and the single-scattering phase function ($P^\lambda(\Omega)$), summarized in Table 1.

As described before, the extinction cross-section is highly wavelength-dependent for air particles and not wavelength-dependent for the larger particles. Therefore, $\sigma_{\text{ex}}^\lambda = \sigma_{\text{ex}}$ for water particles. For large spherical particles, the extinction cross-section is approximately twice the geometric cross-section, $\sigma_{\text{ex}} = 2\pi r^2$ [LLO*91]. The scattering cross-section is $\sigma_{\text{sca}} = \alpha\sigma_{\text{ex}}$, where α is the single-scattering albedo. The extinction cross-sections of the particles used in our system are given in Table 2. Although the system is not limited to

Variable	Definition
α^λ	single-scattering albedo ($\beta_{\text{sca}}^\lambda/\beta_{\text{ex}}^\lambda$)
$A m s_n^\lambda$	multiple scattering angular distribution
$A^\lambda(\vec{s}', \Omega)$	angular distribution function
$\beta_{\text{ex}}^\lambda$	extinction coefficient (m^{-1})
$\beta_{\text{sca}}^\lambda$	scattering coefficient (m^{-1})
Δ	slice spacing (m)
$I^\lambda(\vec{s}')$	total intensity for all directions at \vec{s}'
λ	light wavelength (nm)
$L^\lambda(\vec{s}, \Omega)$	light intensity
\vec{m}_Ω	normalized vector in Ω direction
$n(\vec{s})$	expected number of scattering events
Ω	direction
$\Omega - \Omega'$	scalar angle between directions (radians)
Ω_{lt}	sunlight direction
\vec{p}_Ω	opposite Ω normalized vector
$P^\lambda(\Omega)$	particle single-scattering phase function
$P^\lambda(\Omega - \Omega', \vec{s}')$	combined (spatial) phase function
\vec{s}	3-D point in space
$\vec{s}_{\text{bg}}(\vec{s}, \Omega)$	background point for this ray
$\sigma_{\text{ex}}^\lambda$	extinction cross-section (m^2)
$T^\lambda(\vec{s}, \vec{s}')$	transparency between \vec{s} and \vec{s}'
ξ	User adjustable back-scattering contribution

Table 1: Variable and parameter descriptions

Attribute	$\sigma_{\text{ex}}(\text{m}^2)$
Air at 650 nm	2.28×10^{-31}
Air at 515 nm	5.33×10^{-31}
Air at 440 nm	1.30×10^{-30}
Cloud (0.01 mm sphere)	6.28×10^{-10}
Rain (1 mm sphere)	6.28×10^{-6}
Ice	1.82×10^{-6}
Snow	1.41×10^{-5}

Table 2: Atmospheric particle extinction cross-sections

unity albedo, atmospheric particles have near unity albedos in visible wavelengths [KYBN02, CS92]; therefore, we assume all the particle albedos are one.

The scattering phase function for these particles, $P^\lambda(\Omega)$, is the probability density function describing the angular deflection of light (Ω) after an interaction with the particle. In general, Ω is a vector, but for randomly oriented particles, $P^\lambda(\Omega)$ is a function of the scalar angle between the incoming and outgoing (after scattering) directions. This assumption, valid for a turbulent atmosphere, eliminates some ice crystal optical phenomena, such as parhelia, but simplifies the model substantially. The Rayleigh phase function for air

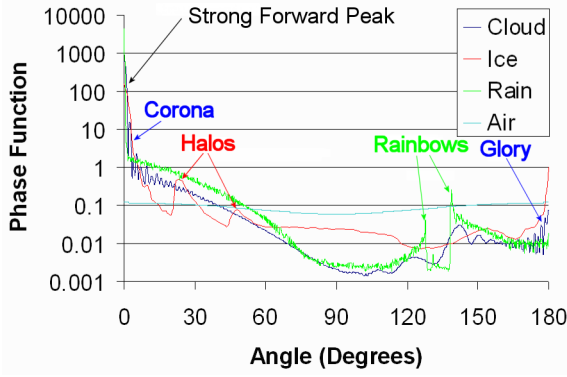


Figure 3: Phase of all particle fields for green light

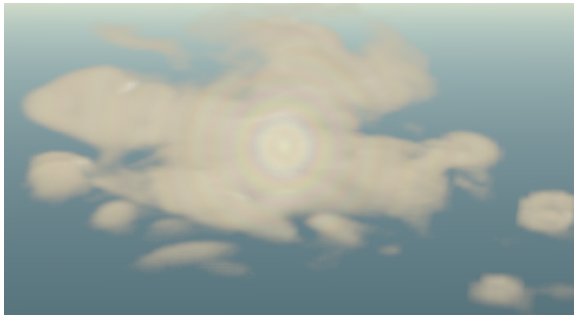


Figure 4: Glory in cumulus cloud

molecules is given in Equation 1.

$$P_{\text{air}}(\Omega) = \frac{3}{16\pi} (1 + \cos^2(\Omega)) \quad (1)$$

Mie scattering dominates for homogeneous spheres much larger than the wavelength of light, and describes rain and cloud particle phase functions. We use Philip Laven’s MiePlot [Lav], an implementation of Bohren and Huffman’s [BH83] Mie code, to calculate the Mie phase function at an angular resolution of 0.1 degrees. Additionally, because rain and clouds are actually a collection of randomly distributed particles, we calculate the phase function using a lognormal distribution for the particle size, with parameters described by Pruppacher and Klett [PK00]. The rain phase function is given in Figure 2. For comparison, Figure 3 is a plot of the phase functions for green light for all particle fields. Cloud water has a similar behavior, with a milder forward peak and more relaxed peaks at rainbow angles as shown in Figure 3. Wavelength dependence for the peaks near 180 degrees in the cloud water phase functions create the glory: tight rings of color reflected backward towards the sun, as shown in Figure 4. Notice the double peaks around 120 and 140 degrees for rain. The wavelength dependence of these peaks causes rainbows, like the one in Figure 5; the lull between them is Alexander’s dark band.

For ice, we assume random orientation such that the phase function is dependent solely on the angle between the eye and light rays. The phase data is taken from Hess and Wiegner’s Cirrus Optical Properties system [HW94]. A plot of the ice phase function for green light is also shown in Figure 3. The peaks of this ice phase function at 22 and 46 degrees create halos of color when the sun penetrates optically thin cirrus clouds, such as in Figure 6. The other wavelengths have skewed values at the peaks at 22 and 46 degrees, making these halos chromatic phenomena.

3.2. Rendering Properties

One quantity of interest for calculating light transport is the extinction coefficient, β_{ex} . This is the probability, per meter, that light intersects a particle. As described by Blinn [Bli82], this can be calculated by multiplying the particle density with the extinction cross-section. At a point in space, \vec{s} , this is calculated as the following:

$$\beta_{\text{ex}}(\vec{s}) = \eta(\vec{s})\sigma_{\text{ex}}. \quad (2)$$

Here η is the particle concentration in particles per cubic meter. For multiple particle fields, the extinction coefficients linearly combine.

$$\beta_{\text{ex}}(\vec{s}) = \sum_{\text{all fields } i} \beta_{\text{ex}}^{(i)}(\vec{s}) \quad (3)$$

The optical depth, τ , along a segment between \vec{s} and \vec{s}' is the integral of the extinction coefficient along the ray.

$$\tau(\vec{s}, \vec{s}') = \int_{\vec{s}}^{\vec{s}'} \beta_{\text{ex}}(\vec{s}'') ds'' \quad (4)$$

The transparency, T , of the medium is then:

$$T(\vec{s}, \vec{s}') = e^{-\tau(\vec{s}, \vec{s}')}. \quad (5)$$

The total phase function at a point in space is calculated by weighting each of the individual particle phase functions by their contribution to the total scattering coefficient.

$$P(\Omega, \vec{s}) = \frac{\sum_{\text{all fields } i} \alpha^{(i)} \beta_{\text{ex}}^{(i)}(\vec{s}) P^{(i)}(\Omega)}{\sum_{\text{all fields } i} \alpha^{(i)} \beta_{\text{ex}}^{(i)}(\vec{s})} \quad (6)$$

Now that we have a system for calculating the spatially varying properties of the particle fields, we shift our attention to a system for rendering these particles that can capture realistic illumination effects.

4. Rendering Atmospheric Data

Our efficient rendering of atmospheric data is accomplished by combining several rendering models. The first is a single-scattering sky-rendering approximation. Single-scattering is used because, for altitude angles greater than about 16 degrees, and with a flat earth, the expected number of air scattering events along a view ray is less than one. This angle



Figure 5: Rainbow in rain field of a storm



Figure 6: Halo in a cirrus cloud

is even smaller for a round earth, as the height will increase faster with distance. Calculation of the sky attenuation and sky lighting terms are given in Section 4.1.

For rendering the larger water and ice particles, we utilize a model based upon multiple scattering phase functions described in Section 4.2. At its core, this is a volume rendering application that renders multi-field atmospheric data on a uniform voxel grid, containing particle density ($\eta(\vec{s})$) values. It applies the half angle slicing scheme for a light buffer and an eye buffer, as presented in Kniss et al. [KPH*03]. This method slices volume data at an angle suitable for both the eye and light perspectives, i.e. at the angle halfway between them. Rendering is done both to an eye (or image) buffer to accumulate light directed into the eye rays, and to a light buffer, which stores the attenuation of light as it traverses the volume.

A diagram showing the flow of the full system is given in Figure 7. For front to back rendering, a slice halfway through the volume that covers the entire image is rendered first, to estimate atmospheric inscattering between the eye and the volume. Then, half-angle slicing is performed on the volume, alternating between compositing in the eye buffer, and attenuating in the light buffer.

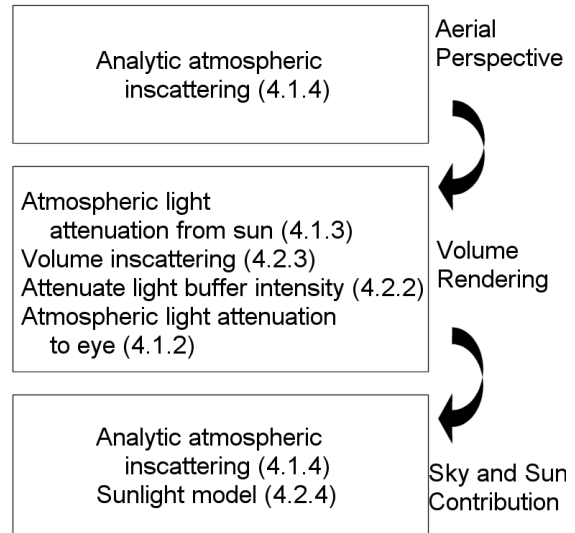


Figure 7: Rendering system flow diagram (front-to-back)

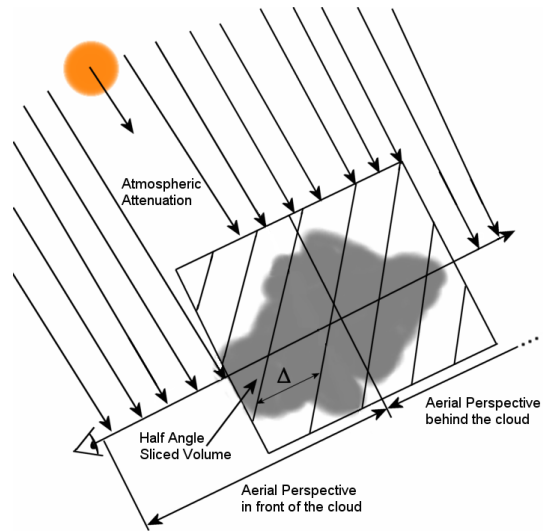


Figure 8: Rendering system overview

Finally, the atmosphere behind the cloud is approximated by calculating all atmospheric inscattering from the plane halfway through the volume onward, and this result is composited into the image. To get an overview of which system components render which model components, an overview image is given in Figure 8.

A general light transport equation, relating the light intensity at \vec{s} in the direction Ω to the light intensity everywhere

else, is given in Equation 7 [Cha60].

$$L^\lambda(\vec{s}, \Omega) = L^\lambda(\vec{s}_{\text{bg}}(\vec{s}, \Omega), \Omega) T^\lambda(\vec{s}_{\text{bg}}(\vec{s}, \Omega), \vec{s}) + \int_{\vec{s}_{\text{bg}}(\vec{s}, \Omega)}^{\vec{s}} T^\lambda(\vec{s}, \vec{s}') \beta_{\text{sca}}^\lambda(\vec{s}') \times \int_{4\pi} P^\lambda(\Omega - \Omega', \vec{s}') L^\lambda(\vec{s}', \Omega') d\Omega' ds' \quad (7)$$

Note that $\vec{s}_{\text{bg}}(\vec{s}, \Omega)$ is the background point found by traversing from \vec{s} in the direction opposite Ω until it intersects with the background, which may be the sun, the earth, or space. Also note that $\Omega - \Omega'$ represents the scalar angle between the direction vectors Ω and Ω' . Equation 7 says that the light at \vec{s} in the direction Ω equals the background intensity defined by \vec{s} and Ω times the transparency through the volume, plus the contribution from inscattering at all points along the ray. The difficult issue is that the light distribution at all points along the ray must be known to calculate the light intensity in a single direction reaching a point along that ray. We now present an analytic aerial perspective model that simplifies this equation with a single-scattering approximation.

4.1. Aerial Perspective

Both aerial perspective and skylight are essential in an atmospheric renderer. Aerial perspective (the blueing of distant dark objects during the day, and the reddening of bright objects at dusk and dawn) is an essential outdoor depth cue, while proper skylight is required for any realistic outdoor scene renderer. It is often desirable to render atmospheric bodies from any perspective and, therefore, we do not limit the observer to be on the ground. Direct sunlight is accounted for separately, in Section 4.2.4. Therefore, the single-scattering analytic model presented here has zero background intensity. We also assume sunlight to be collimated; thus, we need only consider the direction of the sun, Ω_{lt} , in the directional integral over Ω .

Air consists of a consistent mixture of molecules, and, therefore, the phase function of the particles does not change with spatial location. Hence, for atmospheric inscattering, Equation 7 quickly reduces to something more manageable in Equation 8.

$$L^\lambda(\vec{s}, \Omega) = \int_{\vec{s}_{\text{bg}}(\vec{s}, \Omega)}^{\vec{s}} T^\lambda(\vec{s}, \vec{s}') \beta_{\text{sca}}^\lambda(\vec{s}') \times P_{\text{air}}(\Omega - \Omega_{\text{lt}}) L^\lambda(\vec{s}', \Omega_{\text{lt}}) ds' \quad (8)$$

It is clear that P_{air} does not depend on the point in space, and is factored. Therefore, we need only concern ourselves with the three other terms. We first calculate the scattering coefficient. We then consider how to find the transparency from the sample to the eye. Next, we calculate the light value at the sample, assuming the sun is at infinity. Finally, we put it all together into an analytic expression for the light directed toward the eye.

Wavelength (nm)	β_{sl}
680	$5.8 \times 10^{-6} \text{m}^{-1}$
550	$1.35 \times 10^{-5} \text{m}^{-1}$
440	$3.31 \times 10^{-5} \text{m}^{-1}$

Table 3: Sea-level scattering coefficients

4.1.1. Scattering Coefficient

Like Nishita et al. [NSTN93] and Preetham et al. [PSS99], we make the hydrostatic approximation that the atmosphere has an exponentially decreasing particle density from sea-level as shown in Equation 9.

$$\eta(h) = 2.55 \times 10^{25} e^{-\gamma h} \text{m}^{-3} \quad (9)$$

The scattering coefficient is then the scattering cross-section times this particle density. We use a single-scattering model to calculate the aerial perspective because of its analytic simplicity and because we expect few scattering events as sunlight traverses the atmosphere. Since the scattering coefficient is linear in particle concentration, η , the scattering coefficient is an exponentially decreasing function of height.

$$\beta_{\text{sca}}^\lambda(h) = \beta_{\text{sl}}^\lambda e^{-\gamma h} \quad (10)$$

h is the height above sea-level. In this model, the lapse rate is $\gamma = \frac{1}{8000} \text{m}$. The β_{sl} values we use for the three color channels (red, green, and blue) are given in Table 3. Now that we have an expression for the scattering coefficient, we can calculate the transparency of the atmosphere.

4.1.2. Atmospheric Transparency

Air molecules have high albedo and, thus, the probability of extinction (any mechanism for removing light from a ray) is equal to the probability of scattering, $\beta_{\text{ex}} = \beta_{\text{sca}}$. We assume that the earth is flat, so that the height varies linearly along the ray between two points. Let \vec{p}_Ω be the vector in the opposite direction of Ω . Substituting Equation 10 into Equation 5 and performing the integral, we get a solution in terms of the y -component of \vec{p}_Ω , $p_{\Omega y}$, the scattering coefficient at sea-level, $\beta_{\text{sl}}^\lambda$, the height of \vec{s} , s_y , the distance between \vec{s} and \vec{s}' , denoted K , and the lapse rate, γ .

$$T^\lambda(\vec{s}, \vec{s}') = \exp\left(-\frac{\beta_{\text{sl}}^\lambda}{\gamma p_{\Omega y}} e^{-\gamma s_y} \left(1 - e^{-\gamma p_{\Omega y} K}\right)\right) \quad (11)$$

Thus, we have an expression for the transparency of the atmosphere between two points. We use this to calculate the atmospheric attenuation of light to a point, and to find the attenuation from that point to the eye.

4.1.3. Sample Lighting

The last component needed to solve the overall light transport equation is the light value at a point in space, $L(\vec{s}, \Omega_{\text{lt}})$. If

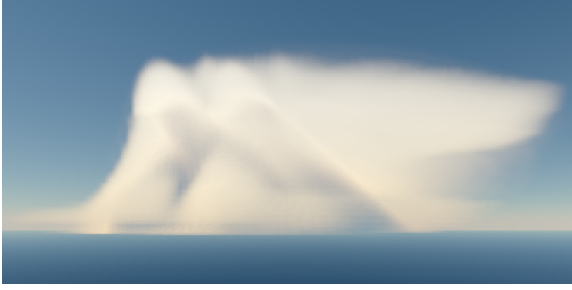


Figure 9: Sky and aerial perspective



Figure 10: Uniform background sky, no aerial perspective

we set the sun intensity outside the atmosphere to be I_s^λ , then we simply use Equation 11 to attenuate this intensity. Let l be the normalized vector pointing towards the sun. Therefore, we are only interested in the initial height of the sample, s'_y , and the y-component of l , l_y . If we assume that l_y is positive for a daylight scene (the light must always be above the horizon) and let the distance go to infinity, the integral converges to Equation 12.

$$L^\lambda(\vec{s}', \Omega_{lt}) = I_s^\lambda \exp\left(-\frac{\beta_{sl}^\lambda}{\gamma l_y} e^{-\gamma s'_y}\right) \quad (12)$$

We now have an expression for all quantities in Equation 8 and can calculate the amount of inscattering along the eye ray between two points.

4.1.4. Analytic Sky Model

Finally, we derive an expression for the amount of atmospheric inscattering of light into a ray along a given distance. To simplify, let β_{eye}^λ be the scattering coefficient at the eye.

$$\beta_{eye}^\lambda = \beta_{sl}^\lambda e^{-\gamma s_y} \quad (13)$$

If we substitute Equations 10, 11, and 12 into Equation 8 we get an equation of the form:

$$L^\lambda(\vec{s}, \Omega) = P_{air}(\Omega - \Omega_{lt}) \times \int_{\vec{s}_{bg}(\vec{s}, \Omega)}^{\vec{s}} \left[\exp\left(-\frac{\beta_{eye}^\lambda}{\gamma p_{\Omega y}} (1 - e^{-\gamma p_{\Omega y} K})\right) \times \beta_{sl}^\lambda e^{-\gamma s'_y} I_s^\lambda \exp\left(-\frac{\beta_{sl}^\lambda}{\gamma l_y} e^{-\gamma s'_y}\right) \right] ds' \quad (14)$$

Now, recognizing that $s'_y = p_{\Omega y} K + s_y$, we solve this integral, and get a final expression in Equation 15 for the skylight reaching the eye from all points along a ray between \vec{s} and a sample that is K away, \vec{s}' .

$$L^\lambda(\vec{s}, \Omega) = I_s^\lambda P_{air}(\Omega - \Omega_{lt}) \frac{l_y}{p_{\Omega y} - l_y} \times \left(\exp\left(-\frac{1}{\gamma} \beta_{eye}^\lambda \left(\frac{1}{l_y} - \frac{1}{p_{\Omega y}}\right) e^{-\gamma p_{\Omega y} K}\right) - \exp\left(-\frac{1}{\gamma} \beta_{eye}^\lambda \left(\frac{1}{l_y} - \frac{1}{p_{\Omega y}}\right)\right) \right) \quad (15)$$

This approximation produces realistic sky results, as shown in Figure 9. For comparison, the same cloud without the sky and aerial perspective is shown in Figure 10. Note that the hue in darker regions of this very large cloud model are more blue in the rendering with aerial perspective than the rendering without. Additionally, this skylight and atmospheric attenuation creates the characteristic reddening of objects at sunrise and sunset. This effect can be seen in the dawn images of cumulus clouds, in Figures 1 and 11, and a cirrus in Figure 12. Now that we have a model for the sky attenuation and source, we present our approximation for multiple scattering in the light volume.

4.2. Multiple Scattering Phase Function Lighting Approximation

We have developed a model for the angular dispersion of light as it experiences its first few scattering events traversing the volume. This model has been designed for efficient rendering in a single pass through the volume, taking advantage of programmable graphics hardware. This technique extends the half-angle slicing technique described by Kniss et al. [KPHE02] to handle the gradual angular spreading of light as it undergoes multiple scattering, rather than only considering unscattered and diffuse light. This is essential for capturing fine angularly dependent effects in the presence of large atmospheric scattering particles, which highly favor forward scattering.

As mentioned previously, the difficulty with Equation 7 is the presence of L^λ on both sides of the equation. To approximate this equation, we first separate $L^\lambda(\vec{s}, \Omega)$ into its

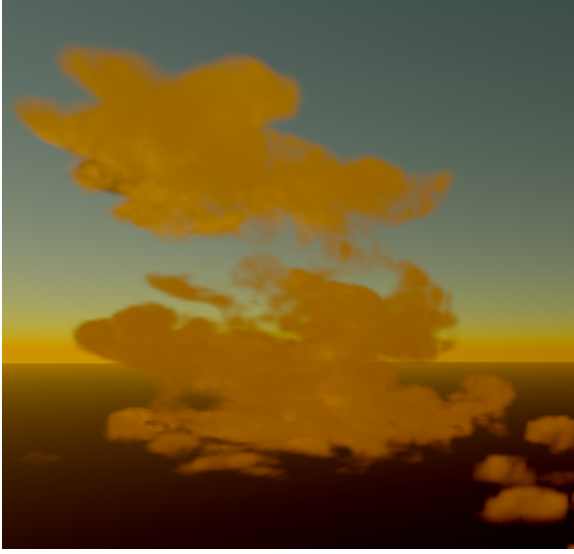


Figure 11: Cumulus cloud at dawn

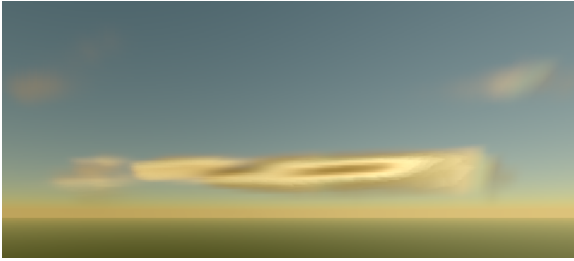


Figure 12: Cirrus cloud with halo at dawn

intensity, $I^\lambda(\vec{s})$, and its angular distribution, $A^\lambda(\vec{s}, \Omega)$.

$$L^\lambda(\vec{s}, \Omega) = I^\lambda(\vec{s})A^\lambda(\vec{s}, \Omega) \quad (16)$$

A^λ , as an angular distribution, is subject to the constraint:

$$\int_{4\pi} A^\lambda(\vec{s}, \Omega) d\Omega = 1, \forall \vec{s}, \lambda \quad (17)$$

The distribution of light, A^λ , is closely tied to the number of scattering events it experiences between the source and the given point in the volume. We make the assumption of a purely collimated initial light source that is then redirected through scattering.

$$L^\lambda(\vec{s}_{bg}(\vec{s}, \Omega), \Omega) = I_s^\lambda \delta(\Omega - \Omega_{lt}) \quad (18)$$

Therefore, all background intensities are equal and have a single direction. In Section 4.2.1, we describe a method for estimating the angular distribution of the light in the volume. Then, in Section 4.2.2, we develop a system for calculating the intensity of light in the volume using this angular distribution.

4.2.1. Estimating Angular Distribution

Since we have separated the light contribution into intensity and distribution, we utilize Equation 18 and rewrite Equation 7 into the form in Equation 19.

$$\underbrace{I^\lambda(\vec{s})A^\lambda(\vec{s}, \Omega)}_{L^\lambda(\vec{s}, \Omega)} = I_s^\lambda \delta(\Omega - \Omega_{lt}) T^\lambda(\vec{s}_{bg}(\vec{s}, \Omega), \vec{s}) + \int_{\vec{s}_{bg}(\vec{s}, \Omega)}^{\vec{s}} I^\lambda(\vec{s}') T^\lambda(\vec{s}, \vec{s}') \beta_{sca}^\lambda(\vec{s}') \times \int_{4\pi} P^\lambda(\Omega - \Omega', \vec{s}') A^\lambda(\vec{s}', \Omega') d\Omega' ds' \quad (19)$$

Clearly, the last integral is a spherical convolution in Ω . After a single scattering event, we expect the scattered light to have the angular distribution of the particle phase function. If the light experiences two scattering events, we expect the twice-scattered light to have an angular distribution of the phase function convolved with itself. Therefore, we define a multiple-scattering angular distribution function, Ams_n^λ , to approximate the distribution of light after n scattering events with Equation 20.

$$Ams_n^\lambda(\Omega) = \left(\delta(\Omega - \Omega_{lt}) * \underbrace{P^\lambda * \dots * P^\lambda}_{n\text{-times}} \right) (\Omega) \quad (20)$$

If we assume that, for a small number of scattering events, the angular distribution of L^λ remains peaked in the initial light direction with gradual angular spreading, then we can approximate the angular distribution with this multiple scattering angular distribution function. The expected number of scattering events for a ray is given in Equation 21.

$$n(\vec{s}) = \int_{\vec{s}_{bg}(\vec{s}, \Omega_{lt})}^{\vec{s}} \alpha^\gamma \beta_{ex}^\gamma(\vec{s}') ds' \quad (21)$$

In terms of transparency from the light source:

$$n(\vec{s}) = -\alpha \ln(T(\vec{s}, \vec{s}_{bg}(\vec{s}, \Omega_{lt}))) \quad (22)$$

We estimate the number of scattering events to be this average number. Note that this is, nominally, a function of the direction and the data along all possible paths, but, for efficiency, we estimate the number of scattering events as this mean. The result is that, by tracking only the transparency of the material traversed from the light source, we can get an approximate expression for the angular distribution of the total intensity at that point. Our system records a single intensity value in a light buffer (see Section 5) and sums the contribution from the previous slice that is directed to the current sample. A preset number of angular distribution functions, $Ams_n^\lambda(\Omega)$, is pre-computed and saved in a floating-point precision texture. These functions are calculated for up to $n = 26$, when the angular distribution function for rain (the worst case with the highest forward peak) becomes isotropic (to within .0001%).

4.2.2. Intensity Calculation

With this approximation for the angular distribution of the light, it is easier to calculate its intensity distribution. This intensity distribution is the quantity stored in the light buffer, and represents the total light intensity reaching the samples. This value, along with its angular distribution is then used to calculate the final gather to the eye, as described in Section 4.2.3. The intensity received at a given sample normally has two components: the unscattered light intensity already directed towards the sample, and the scattered light that has been re-directed into the sample. Because we calculate the scattered angular distribution of light, these are not handled separately.

To model the attenuation of light, from one sampling slice to the next, we discard all backward directed light (relative to the direction of the sun) for each scattering event. This is done for two primary reasons. The first is that the phase functions have only a weak back-scattered component. The second is that this system performs a single slicing pass through the volume and, therefore, has no knowledge of the data on future slices. Note that this is for intensity transport in the light buffer only, and that light may still be backscattered towards the eye in the final gather, described in Section 4.2.3. Therefore, we let F_i^λ be the forward-scattered contribution after i scatters, i.e., the integral over the forward hemisphere of $\text{Ams}_i^\lambda(\Omega)$.

$$F_i^\lambda = \int_{\text{forward hemisphere}} \text{Ams}_i^\lambda(\Omega) d\Omega \quad (23)$$

Let $\vec{m}_{\Omega_{it}}$ be the vector in the direction of Ω . To get the intensity for the current slice ($I(\vec{s} + \vec{m}_{\Omega_{it}}\Delta)$), from the intensity on the previous slice ($I(\vec{s})$), for a discrete number of scattering events, the light is attenuated according to Equation 24.

$$I(\vec{s} + \vec{m}_{\Omega_{it}}\Delta) = I(\vec{s})T(\vec{s}, \vec{s} + \vec{m}_{\Omega_{it}}\Delta) + I(\vec{s})(1 - T(\vec{s}, \vec{s} + \vec{m}_{\Omega_{it}}\Delta)) \prod_{i=n(\vec{s})+1}^{n(\vec{s}) + \lceil \beta_{\text{ex}}^\lambda \Delta \rceil} \alpha(\vec{s}) F_i^\lambda \quad (24)$$

This product is not practical for implementation on graphics hardware. Linear interpolation is not yet available for floating point textures, therefore, we implement intensity loss for the worst case attenuation (highest number of scattering events) in Equation 25 for efficiency.

$$I(\vec{s} + \vec{m}_{\Omega_{it}}\Delta) = I(\vec{s})T(\vec{s}, \vec{s} + \vec{m}_{\Omega_{it}}\Delta) + I(\vec{s})(1 - T(\vec{s}, \vec{s} + \vec{m}_{\Omega_{it}}\Delta)) \left(\alpha(\vec{s}) F_{\lceil n(\vec{s})+1 + \beta_{\text{ex}}^\lambda \Delta \rceil}^\lambda \right)^{\beta_{\text{ex}}^\lambda \Delta} \quad (25)$$

This intensity value is calculated and stored in its own light buffer, as described in Section 5. This approximation introduces errors that increase with the number of scattering events, as the backward directed light contribution becomes large. Thus, our approximation performs best at the optically thin regions of the cloud, and for the first few scattering events, where we wish to capture fine angularly dependent

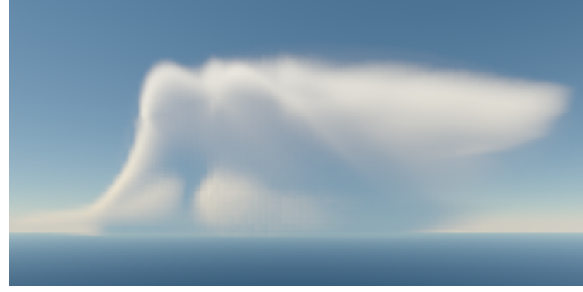


Figure 13: Low albedo rendering

optical effects, and is least accurate in dense inner regions of the cloud.

4.2.3. Rendering with Angular Distribution

To apply this result, we do a single slicing pass through the volume, collecting the inscattered intensity from the volume samples into the eye ray. For light scattered to the eye, we apply the angular distribution function for one more scattering event than the expected number of scattering events reaching this sample, $n(\vec{s}_f + i\vec{m}_{\Omega}\Delta) + 1$, as this scattering event is the one that directs light into the eye ray. Rendering without any backscattered light darkens the region of the cloud where light first strikes it; therefore, a user adjustable term is added to the phase function, ξ . This term models the amount of light the system will receive back from multiple scattering further along the path. If we assume front-to-back compositing, and the beginning of the ray to be \vec{s}_f , then Equation 26 describes the accumulation for the pixel.

$$L(\vec{s}, \Omega) = \sum_{i=\text{front}}^{\text{back}} T(\vec{s}, \vec{s}_f + i\vec{m}_{\Omega}\Delta) I(\vec{s}_f + i\vec{m}_{\Omega}\Delta) \times \left(\text{Ams}_{\lceil n(\vec{s}_f + i\vec{m}_{\Omega}\Delta) + 1 \rceil}^\lambda (\Omega - \Omega_{it}) + \xi \right) \beta_{\text{sca}} \Delta \quad (26)$$

This is a sum over sampling slabs, i , that scatter light contributions towards the eye. This model allows us to capture essential single-scattering effects, as seen in Figures 4, 5, and 6, while not suffering from the premature darkening that limits low-albedo models. Observe the darker cloud for a low-albedo lighting model in Figure 13, versus the more accurate results in Figure 9. Note that the transparency must include both attenuation by the volume and by the atmosphere.

4.2.4. Direct sunlight

The direct contribution of the sun is added in the distant light pass for skylight. We store the total transparency of the volume in the alpha channel of the eye buffer. This value attenuates the sunlight, as does the atmosphere. The dot product of the view and light vectors is then used to determine if the view vector is within the 0.5 degree disk of the sun. If so, this attenuated sunlight term is added.

5. Implementation

We implemented this model on nVidia GeForce FX graphics hardware using per-vertex and per-pixel shading programs, written in nVidia's Cg shading language. We created four fragment programs: one for rendering the sky from the eye to halfway through the volume, one for per-slice rendering the volume to the eye buffer (with Equation 26), a third for per-slice rendering to the light buffer (with Equation 25), and a fourth for rendering the sky from halfway through the volume to the outer atmosphere with sunlight. We consider all atmospheric scattering from halfway through the volume to the eye as in front of the cloud, and all scattering further away as behind the cloud. This produces the desirable aerial perspective effects at a nominal cost in rendering time. As described by Kniss et al. [KPHE02], we composite according to the relationship between the eye and the sun. For acute angles between the eye and light directions, we composite in front-to-back fashion with respect to the eye, and for obtuse angles we composite in back-to-front fashion with respect to the eye. A flow diagram for the system for acute angles is given in Figure 7. The obtuse angle system flow simply reverses the order of the sky and aerial perspective shaders. This system uses floating-point buffers in order to achieve high-dynamic-range results. We use a single, large, floating-point buffer render target with the OpenGL RENDER_TO_TEXTURE extension. This buffer has an eye half, and a light half, which are selected by performing a viewport transformation, as described in Kniss et al. [KPH*03]. By binding this pixel buffer as a render target in its own context, we are able to read from and render to the same pixel buffer. This avoids the undesirable slowdown caused by copying buffers, or context switching between two sets of buffers. According to the specification, rendering to and reading from the same buffer produces undefined behavior. We have found that if small overlapping polygons are rendered and blended in this fashion, rather than the series of large sampling polygons used in this system, the texture cache must be flushed, or erroneous results occur. The pixel buffer texture is used to gather the light sample ($I(\vec{s}_f + i\vec{m}_\Omega\Delta)$ in Equation 26) and the previous eye sample for alpha-blending in the eye fragment program. This texture also provides the previous light sample for attenuation according to Equation 25. The alpha channel of the light buffer stores the number of scattering events for that sample, $n(\vec{s}_f + i\vec{m}_\Omega\Delta)$. We also use floating-point textures to store the high-dynamic range multiple scattering phase functions, $Ams_i^\lambda(\Omega)$ for all the particles, with up to 26 convolutions, at a resolution of about 0.18 degrees. The F factors for up to 26 convolutions are also stored in this texture, as the light distribution is essentially isotropic thereafter.

6. Results

Our system is capable of capturing essential optical effects in the atmosphere. These include rainbows (Figure 5), halos

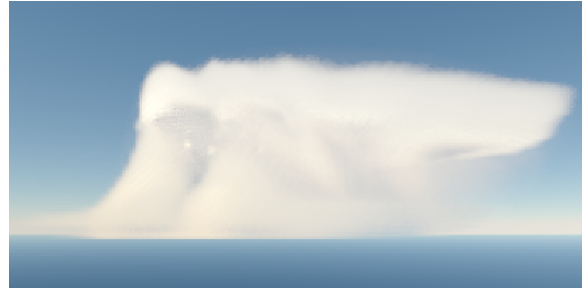


Figure 14: 512×512, with 256 sampling planes, at 3.0 fps

(Figure 6), and the glory (Figure 4). It also captures the aerial perspective of the volume (Figure 9) and sunrise/sunset effects (Figures 1, 11 and 12). Our hardware implementation renders these effects with reasonable performance: all images were rendered at a resolution of 768×768 with between 64 and 512 volume sampling planes in less than 4 seconds using a 2.66 GHz Intel Xeon processor, with 2GB memory, and a 256 MB prototype nVidia GeForce FX 6800 running at 350 MHz, with 1 GHz memory. Interactive rates are achieved, at a marginal cost in quality, by setting the resolution to 512×512 with 256 sampling planes, as shown in Figure 14. We applied our system successfully to large scale storm simulations (Figures 5, 9 and 14), microscale cloud simulations (Figures 4, 11), and cloud modeling software output (Figures 1 and 12).

7. Conclusion

We have developed a new system for the efficient rendering of many atmospheric optical phenomena. By creating a model based on multiple scattering phase functions for the first few scatters, we have developed a rendering system optimized for rendering the many complex optical phenomena found in the phase functions of cloud volumes. This model is supplemental to more conventional light diffusion approximations, and is a significant component in an atmospheric rendering package. Our system is capable of capturing fine angularly dependent chromatic effects, that can trouble other approaches to global illumination. The scattering behavior of water and air particles is vastly different, and, thus, we handle them separately. Aerial perspective is approximated by considering atmospheric inscattering up to halfway through the cloud as in front of the cloud, and the remaining atmospheric inscattering as behind it. We have successfully shown that complex atmospheric optical phenomena may be rendered at interactive frame rates on programmable graphics hardware with this multiple model system.

8. Future Work

While this model can capture angularly dependent chromatic effects efficiently, the errors it introduces, particularly in the dense regions of the cloud, need to be quantified. Through comparison with Monte-Carlo and other rendering techniques we will better establish where this model is best and worst behaved, and likely derive improvements to the system. In the future, we would like to explore the possibility of having the multiple scattering angular distribution rendering results seed another global multiple scattering model to get maximum quality, though non-interactive, renderings. Skylight as a light source on the volume would add realism to this model. Adding parhelia effects to this system is another possible extension.

Acknowledgements

The authors would like to thank nVidia for their support in providing the prototype GeForce FX 6800. We would also like to thank the reviewers for their many insightful and helpful comments. This material is based upon work supported by the National Science Foundation under Grant Nos. 0222675, 0081581, 0121288, 0196351, and 0328984.

References

- [BH83] BOHREN C. F., HUFFMAN D. R.: *Absorption and Scattering of Light by Small Particles*. John Wiley and Sons, 1983. 4
- [Bli82] BLINN J.: Light reflection functions for simulation of clouds and dusty surfaces. *Computer Graphics* 8, 3 (July 1982), 21–28. 2, 4
- [Cha60] CHANDRASEKHAR S.: *Radiative Transfer*. Dover Publications, 1960. 6
- [CS92] CORNETTE W. M., SHANKS J. G.: Physically reasonable analytic expression for the single-scattering phase function. *Applied Optics* 32, 16 (June 1992), 3152–3160. 3
- [DCH88] DREBIN R., CARPENTER L., HANRAHAN P.: Volume rendering. *Computer Graphics (SIGGRAPH '88 Proc.)* 22 (Aug. 1988), 65–74. 2
- [EMP*03] EBERT D., MUSGRAVE K., PEACHEY D., PERLIN K., WORLEY S.: *Texturing and Modeling A Procedural Approach*, 3rd ed. Morgan Kaufman Publishers, 2003. 2
- [EP90] EBERT D. S., PARENT R. E.: Rendering and animation of gaseous phenomena by combining fast volume and scanline a-buffer techniques. *Computer Graphics* 24, 4 (Aug. 1990), 357–366. 2
- [HBSL03] HARRIS M. J., BAXTER W. V., SCHEUERMANN T., LASTRA A.: Simulation of cloud dynamics on graphics hardware. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware* (2003), Eurographics Association, pp. 92–101. 2
- [HL01] HARRIS M. J., LASTRA A.: Real-time cloud rendering. *Eurographics 2001 Proceedings* 20, 3 (Sept. 2001), 76–84. 2
- [HW94] HESS M., WIEGNER M.: Cop: a data library of optical properties of hexagonal ice crystals. *Appl. Opt.* 33 (1994), 7740–7746. 4
- [JC98] JENSEN H., CHRISTENSEN P.: Efficient simulation of light transport in scenes with participating media using photon maps. *Proceedings SIGGRAPH '98, Computer Graphics Proc, Ann Conf. Series* (1998), 311–320. 2
- [JDS*01] JENSEN H. W., DURAND F., STARK M. M., PREMOŽE S., DORSEY J., SHIRLEY P.: A physically-based night sky model. In *Proceedings of ACM SIGGRAPH 2001* (Aug. 2001), Computer Graphics Proceedings, Annual Conference Series, pp. 399–408. 2
- [JMLH01] JENSEN H. W., MARSCHNER S. R., LEVOY M., HANRAHAN P.: A practical model for subsurface light transport. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM Press, pp. 511–518. 2
- [JW97] JACKEL D., WALTER B.: Modeling and rendering of the atmosphere using Mie-scattering. *Computer Graphics Forum*, 4 (1997), 201–210. 2
- [JW98] JACKEL D., WALTER B.: Simulation and visualization of halos. *Proceedings of ANIGRAPH '98* (1998). 2
- [Kla87] KLASSEN V. R.: Modeling the effect of the atmosphere on light. *ACM Transactions on Graphics* 6, 3 (July 1987), 215–237. 2
- [KPH*03] KNISS J., PREMOŽE S., HANSEN C., SHIRLEY P., MCPHERSON A.: A model for volume lighting and modeling. *IEEE Transactions on Visualization and Computer Graphics* 2003 9, 2 (April-June 2003), 109–116. 2, 5, 10
- [KPHE02] KNISS J., PREMOZE S., HANSEN C., EBERT D.: Interactive translucent volume rendering and procedural modeling. In *Proceedings of the conference on Visualization '02* (2002), IEEE Computer Society, pp. 109–116. 2, 7, 10
- [KVH84] KAJIYA J. T., VON HERZEN B. P.: Ray tracing volume densities. *Computer Graphics* 18, 3 (July 1984), 165–174. 1, 2
- [KYBN02] KEY J. R., YANG P., BAUM B. A., NASIRI

- S. L.: Parameterization of shortwave ice cloud optical properties for various particle habits. *Journal of Geophysical Research* 107, D13 (July 2002), 4181. 3
- [Lav] LAVEN P.: Mieplot. <http://www.philiplaven.com>. 4
- [LLO*91] LIU K., LEE J., OU S., FU W., TAKANO Y.: Ice cloud microphysics, radiative transfer and large-scale cloud processes. *Meteorology and Atmospheric Physics* 46 (1991), 41–50. 3
- [Max94] MAX N.: Efficient light propagation for multiple anisotropic volume scattering. *Proc. of the Fifth Eurographics Work-Shop on Rendering* (1994), 87–104. 2
- [Min93] MINNAERT M.: *Light and Color in the Outdoors*. Springer-Verlag New York, 1993. 1
- [MSM*04] MAX N., SCHUSSMAN G., MIYAZAKI R., IWASAKI K., NISHITA T.: Diffusion and multiple anisotropic scattering for global illumination in clouds. *Journal of WSCG* 12, 2 (2004), 277–284. 2
- [Mus89] MUSGRAVE F.: Prisms and rainbows: a dispersion model for computer graphics. *Proceedings of Graphics Interface '89* (1989), 227–234. 2
- [NDN96] NISHITA T., DOBASHI Y., NAKAMAE E.: Display of clouds taking into account multiple anisotropic scattering and sky light. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), ACM Press, pp. 379–386. 2
- [NSTN93] NISHITA T., SIRAI T., TADAMURA K., NAKAMAE E.: Display of the earth taking into account atmospheric scattering. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (1993), ACM Press, pp. 175–182. 2, 6
- [PAS03] PREMOŽE S., ASHIKHMIN M., SHIRLEY P.: Path integration for light transport in volumes. In *Proceedings of the 13th Eurographics workshop on Rendering* (2003), Eurographics Association, pp. 52–63. 2
- [PK00] PRUPPACHER H. R., KLETT J. D.: *Microphysics of Clouds and Precipitation*, 2 ed., vol. 18. Kluwer Academic Publishers, 2000. 4
- [PSS99] PREETHAM A. J., SHIRLEY P., SMITS B.: A practical analytic model for daylight. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 91–100. 2, 6
- [REHL03] RILEY K., EBERT D., HANSEN C., LEVIT J.: Visually accurate multi-field weather visualization. *Proceedings IEEE Visualization 2003* (Oct. 2003), 279–286. 2
- [RT87] RUSHMEIER H., TORRANCE K.: The zonal method for calculating light intensities in the presence of a participating medium. *Computer Graphics* 21, 4 (July 1987), 293–302. 2
- [SF95] STAM J., FIUME E.: Depiction of fire and other gaseous phenomena using diffusion processes. *SIGGRAPH 95 Conference Proceedings, Annual Conference Series* (Aug. 1995), 129–136. 2
- [SSEH03] SCHPOK J., SIMONS J., EBERT D., HANSEN C.: A real-time cloud modeling and rendering system. *Symposium on Computer Animation 2003* (July 2003). 2
- [Sta95] STAM J.: Multiple scattering as a diffusion process. *Proceedings of the 6th Eurographics Workshop on Rendering, Dublin, Ireland* (June 1995), 51–58. 2
- [Tes87] TESSENDORF J.: Radiative transfer as a sum over paths. *Phys. Rev. A* 35, 2 (Jan. 1987), 872–878. 2