Photorealistic Real-Time Outdoor Light Scattering

Some of the most striking aspects of outdoor scenes are the result of light interacting with the atmosphere: shades of blue in a clear noon sky; the red and gold colors of sunset; the purple tint of distant hills; the gray, washed-out look of a foggy day.

In this article, we will explain the basic principles of scattering physics, and use them to derive a scattering model. We will then show how to implement this model with a vertex shader, so that these effects can be generated and changed in real time.

Pixel values are RGB triples. However, radiance is distributed along a continuous range of frequencies. There are two ways to derive RGB values from a set of wavelength-dependent equations. The fast way (commonly used for real-time graphics) is to plug three sample frequencies into the equations, resulting in an RGB triple. The more precise way (often used for offline rendering) is to use several dozen samples evenly distributed throughout the visible spectrum. The resulting series of numbers is converted to an RGB triple via perceptual weighting and integration.

Scattering Fundamentals

We will start with some fundamental concepts as a backgrounder:

Radiant flux (ϕ) measures a quantity of light through a surface (through all points and in all directions). Radiant flux is power, which is measured in watts.

Solid angle (w) measures a sheaf of directions in 3D, like an angle measures a sheaf of directions in 2D. While angles are defined as arcs on a circle and measured in radians (2π in a complete circle), solid angles are defined as patches on a sphere and measured in steradians (4π in a complete sphere).

Radiant intensity (I) measures a quantity of light through all points in a surface going in a single direction. Radiant intensity is power over solid angle, and is measured in watts per steradian.

Radiance (L) measures a quantity of light in a single ray (through a single point in a single direction). Radiance is power over (area times solid angle), and is measured in watts per steradian per meter squared. The pixel values of the final rendered image are derived from the radiance values for rays going through each pixel into the camera.

Image by Solomon Srinivasan

NATY HOFFMAN | *Naty has been leading the development of the Earth & Beyond graphics engine at Westwood Studios since 1997. Previously he worked at Intel as the lead microprocessor architect for the Pentium with MMX chip and contributed to the MMX, SSE, and SSE 2 instruction sets. Contact him at naty@westwood.com.*

ARCOT J. PREETHAM | *Preetham (preetham@ati.com) is a software engineer working on various rendering techniques for next-generation graphics hardware at ATI Research. Prior to this, he developed 3D modeling and reverse-engineering software at Paraform, and worked on rendering atmospheric effects for flight simulators at Evans & Sutherland.*

We will use the fast method, but at a cost. Figure 1 shows the spectral sensitivity for the three kinds of cones in the human retina. We can see that no matter which three sampling frequencies we pick, we will lose information on the spectral structure between them, which introduces inaccuracies.

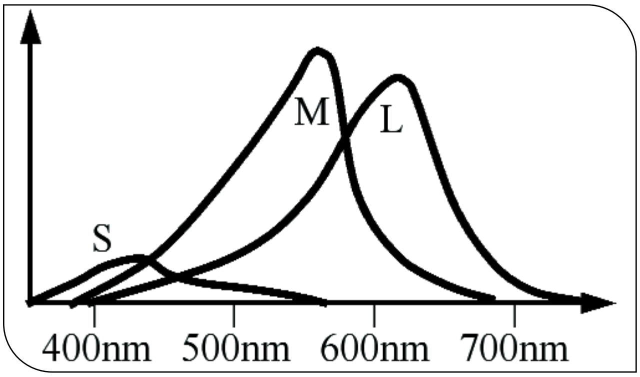


FIGURE 1. Spectral sensitivity curves for cones in the retina.

Atmospheric Light Scattering

There are three types of interactions that can occur between a photon and a particle (an atom, molecule, dust speck, water droplet, and so on). The particle may scatter the photon into the line of sight (in-scattering), it may scatter it out of the line of sight (out-scattering), or it may absorb the photon altogether (absorption).

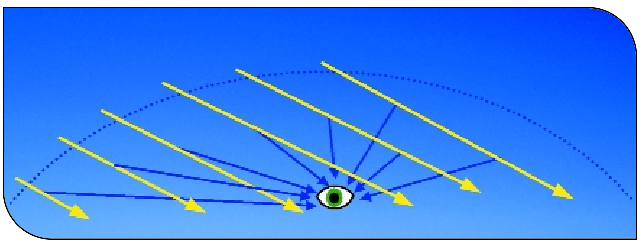


FIGURE 2. Skylight is scattered toward the eye by the atmosphere. Because blue light scatters more than red light, the sky usually appears blue.

Atmospheric light scattering (we include both scattering and absorption under this term) is responsible for many varied visual effects in outdoor scenes, but for the purposes of this article we will concentrate on three: the sky, sunlight, and aerial perspective. First we will discuss the sky. When looking at a clear sky you would see nothing but black if atmospheric light scattering was not present. In Figure 2 we can see how the atmosphere scatters sunlight toward the eye. Since blue light tends to scatter more than red light (more on this later), the sky usually appears blue.

In Figure 3 we can see how the atmosphere, via out-scattering and absorption, removes part of the sunlight before it reaches the eye. Again, mostly blue light is affected, which causes the

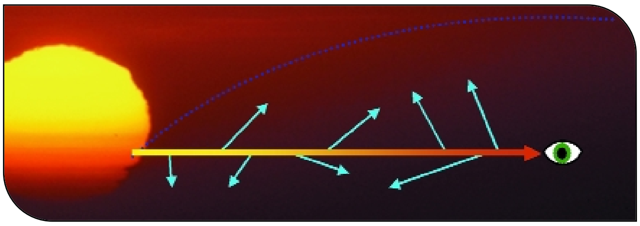


FIGURE 3. The atmosphere removes some sunlight before it reaches the eye by out-scattering an absorption. The scattering of blue light makes the sky appear red when the sun is near the horizon and its light must travel farther through the atmosphere to reach the eye.

color of the remaining sunlight to shift toward yellow and red. When the sun is near the horizon, sunlight travels a much larger distance through air than when it is at the zenith. This explains why this effect is strongest at sunrise and sunset.

Aerial perspective causes distant objects to shift in color. In Figure 4 we can see how the atmosphere attenuates the light from distant objects via out-scattering and absorption, and adds new light via in-scattering. Since mostly blue light is involved, this causes distant dark objects to appear blue and distant bright objects to appear reddish.

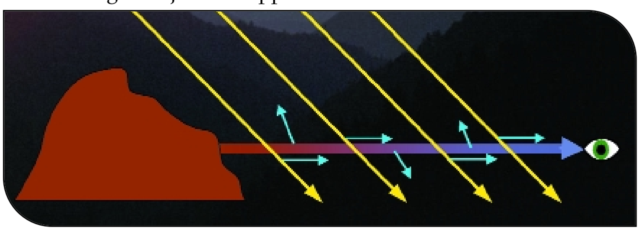


FIGURE 4. Aerial perspective causes distant bright objects to appear reddish and distant dark objects bluish.

We can divide the scattering phenomena into two groups: those which remove light from a ray (absorption and out-scattering, which we will combine under the term "extinction"), and those which add it (in-scattering). Let's compare the radiance in a ray before and after it is affected by atmospheric light scattering (L_0 and $L_{\text{scattering}}$, respectively). Extinction has a multiplicative effect on L_0, which we can express as a dimensionless factor F_{ex}. In-scattering has an additive effect on L_0, which we can express as a radiance value L_{in}. This gives us Equation 1:

$$L_{\text{scattering}} = F_{\text{ex}} L_0 + L_{\text{in}} \qquad \text{Eq. 1}$$

The rest of this article will focus on how to calculate F_{ex} and L_{in} for all objects in a scene in real time.

Absorption

The absorption cross section measures how well a single particle absorbs light around it. In Figure 5 we see a single absorbent particle. The first assumption we will make is that the particle interacts with light in an isotropic manner, that is, it doesn't matter from which direction the light comes. Given this, we will look only at the light coming from a single direction and ignore light coming from other directions. In this example, the light from this direction has a constant radiance L. The particle will absorb a certain amount of total radiant flux ϕ_{ab}, however we only care about the absorbed flux coming from one direction (the absorbed radiant intensity, I_{ab}).

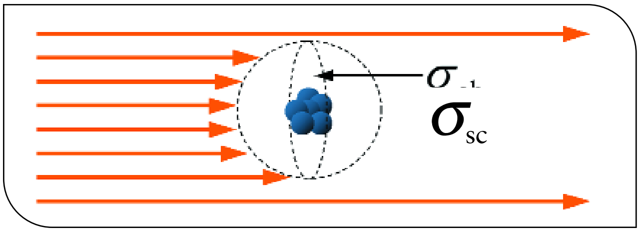


FIGURE 5. A single particle absorbing light.

We define the absorption cross section as the absorbed radiant intensity per unit incident radiance, or I_{ab}/L (this is equivalent to the commonly used definition of absorbed flux per unit irradiance and is easier to explain). If we assume that the particle is a solid absorbent sphere, then for the purpose of absorbing light from this direction we can treat it as a flat disc perpendicular to the light. Each point in this disc absorbs an amount of radiance. Integrating over the disc's area A gives us $I_{ab} = AL$, so $\sigma_{ab} = A$.

If the particle is very large compared to the light wavelength, then its absorption cross section is equal to its geometric cross section. Smaller particles cannot really be treated as spheres (or as having any shape at all), but fortunately we don't need to care — σ_{ab} captures everything we need to know about how well they absorb light. Note that σ_{ab} varies as a function of wavelength, so it is actually an RGB triple: σ_{ab}^{R}, σ_{ab}^{G}, σ_{ab}^{B}.

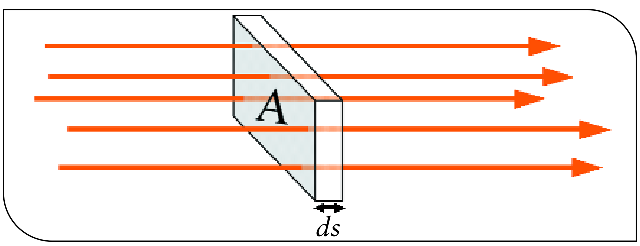


FIGURE 6. A thin slab of absorbent medium.

Understanding how a single particle absorbs light is all well and good, but how is light affected by passing through an absorbent medium containing many such particles? We will characterize this with a new quantity: the medium's absorption coefficient , defined as the particle density multiplied by . Since density is measured in meters–3 (particles per cubic meter), the units of work out to be meters–1, or inverse length. This seems a bit odd at first but will make perfect sense in a moment.

In Figure 6, we see a thin slab of absorbent medium with depth ds and area A, through which photons are passing in a perpendicular direction. The total absorption cross section of the slab is σ_{ab} multiplied by the number of particles in the slab. The number of particles is equal to ρ_{ab} multiplied by the slab volume, which is equal to Ads. This gives us a total absorption area of $A_{ab} = \sigma_{ab}\rho_{ab}Ads$ for the slab. The probability P_{ab} that any given photon will be absorbed is equal to the ratio of the total absorption area to the slab area, which is:

$$P_{ab} = A_{ab}/A = \sigma_{ab}\rho_{ab}ds = \beta_{ab}ds \qquad \text{Eq. 2}$$

So the significance of β_{ab} is that it relates the distance a photon travels through the medium to its chance of being absorbed. This explains why it has units of inverse length — β_{ab} times distance equals probability, a dimensionless number. Another way to look at β_{ab} is that it relates the distance a ray of light travels through the medium to the degree by which its radiance is attenuated by absorption. With this in mind, we can rewrite Equation 2 as a differential equation:

$$\frac{dL}{ds} = -\beta_{ab}L$$

We will assume that β_{ab} is constant along the ray's path. Then we can solve this equation to get the radiance of a ray (with starting radiance L_0) after traveling a distance through the medium:

$$L(s) = L_0 e^{-\beta_{ab}s}$$

This is a simple exponential decay formula. If β_{ab} is not constant, the solution is more complicated (see Hoffman and Preetham in For More Information). Note that if we have different types of absorbent particles in the medium we can just add their absorption coefficients together and use the sum as the total absorption coefficient. $L(s)$, L_0, and β_{ab} are RGB triples.

Out-Scattering

The derivation for out-scattering is similar to that for absorption. We have a scattering cross section σ_{sc} (see Figure 7), scattering coefficient $\beta_{sc} = \sigma_{sc}\rho_{sc}$ and the equation for radiance attenuation due to out-scattering in a constant medium is:

$$L(s) = L_0 e^{-\beta_{sc}s}$$

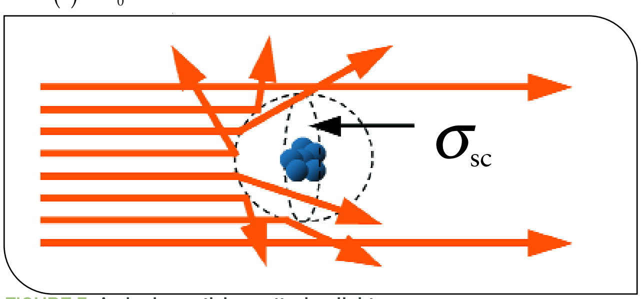


FIGURE 7. A single particle scattering light.

We can also add up the scattering coefficients for different particle types and use the sum as the total scattering coefficient.

Extinction

Since both absorption and out-scattering cause attenuation of light, we can sum the absorption and scattering coefficients to get the extinction coefficient:

$$\beta_{ex} = \beta_{ab} + \beta_{sc}$$

Then the total attenuation due to extinction is:

$$F_{ex}(s) = e^{-\beta_{ex} s} \qquad \text{Eq. 3}$$

In-Scattering

Light is scattered into the view ray from all directions; we will only handle in-scattering from the sun. The scattering coefficient tells us how much light is scattered but not in which direction. For this we define the scattering phase function $f(\theta,\varphi)$. This is a density function for the probability of a photon being scattered in the direction θ,φ. We assume that $f(\theta,\varphi)$ depends only on the angle θ between the incoming direction and the scatter direction ($f(\theta,\varphi) = f(\theta)$), and that it is not wavelength-dependent. Both assumptions are reasonably accurate for most classes of atmospheric particles. The phase function's units are inverse solid angle, and integrating it over the sphere yields a result of 1.0.

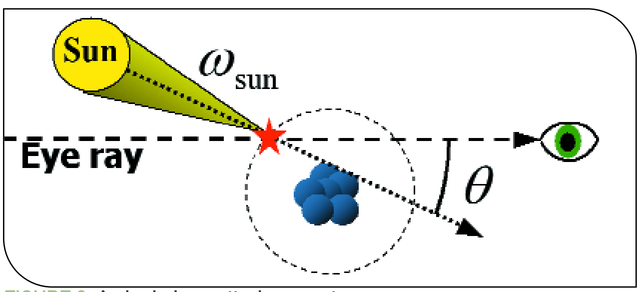


FIGURE 8. A single in-scattering event.

In Figure 8 we can see that the view ray intersects the scattering cross section of the particle, so an in-scattering event is happening (as well as an out-scattering event, but that isn't relevant to the present discussion). How much radiance is scattered into the view ray by this event? First we need to integrate $f(\theta)$ over the sun's solid angle to get the probability that radiance is in-scattered from there. Since the sun covers a small cone (about half a degree across), we can assume $f(\theta)$ does not vary within it. In this case the in-scattering probability is equal to $f(\theta)\omega_{sun}$. To find the amount of radiance in-scattered by this event, we multiply the in-scattering probability by the sun's radiance, L_{sun}, to get $f(\theta)\omega_{sun}L_{sun}$. We define a new constant $E_{sun} = \omega_{sun}L_{sun}$, which expresses the total illumination intensity of the sun and is similar to intensity values used in point light source lighting equations (there the intensity is "squeezed" into a zero solid angle, which implies an infinite radiance — point light sources don't really exist). Then the radiance added by a single in-scattering event is $E_{sun}f(\theta)$.

To get the total in-scattered radiance over a short distance ds, we need to multiply $E_{sun}f(\theta)$ by the probability of an in-scattering event, which is $\beta_{sc}ds$. The result is $E_{sun}f(\theta)\beta_{sc}ds$. We define the angular scattering coefficient $\beta_{sc}(\theta)$ as equal to $\beta_{sc}f(\theta)$. Then the in-scattered radiance over the distance ds is $E_{sun}\beta_{sc}(\theta)ds$. This gives us another differential equation:

$$\frac{dL}{ds} = E_{sun}\beta_{sc}(\theta)$$

Unfortunately, we can't solve this equation without taking extinction into account, since in-scattered light undergoes extinction before it reaches the eye. Adding extinction gives us the following differential equation:

$$\frac{dL}{ds} = E_{sun}\beta_{sc}(\theta) - \beta_{ex}L$$

If we assume E_{sun}, $\beta_{sc}(\theta)$, and β_{ex} are constant along the path, then the solution to this equation is fairly simple (otherwise the solution is much more involved, see Hoffman and Preetham in For More Information). It is essentially Equation 3, plus a new in-scattering factor:

$$L_{in}(s,\theta) = \frac{1}{\beta_{ex}} E_{sun}\beta_{sc}(\theta)\left(1 - e^{-\beta_{ex} s}\right) \qquad \text{Eq. 4}$$

The in-scattered radiance is a function of s (the distance from the eye) and θ (the angle between the viewing ray and the sun). Equations 1, 3, and 4 together describe the complete scattering equation.

Filling in the Parameters

Now that we have the complete scattering equation, we need to determine the parameter values to plug into it: β^R_{ex}, β^G_{ex}, β^B_{ex}, β^R_{sc}, β^G_{sc}, β^B_{sc}, E_{sun}, and $f(\theta)$. E_{sun} is itself dependent on extinction — we will take care of it in the implementation section. In the next two sections we will look at two kinds of particles and determine the coefficients and phase functions for each. We will sum β_{ex} for the two types to get the total β_{ex}, and the two $\beta_{sc}(\theta)$ functions will be added to get the total $\beta_{sc}(\theta)$.

Air Molecules and Rayleigh Scattering

First we will look at particles much smaller than the wavelength of visible light, such as air molecules. These particles do not absorb light, so we will look only at scattering. The scattering coefficients for these particles were discovered by Lord Rayleigh around 1870, so this type of scattering is called Rayleigh scattering (see For More Information). For air we use the following scattering coefficient:

$$\beta_{scAir} = \frac{8\pi^3(n^2-1)^2}{3N\lambda^4}\left(\frac{6+3p_n}{6-7p_n}\right)$$

Where n is the refractive index of air (a dimensionless quantity, equal to 1.0003 in the visible spectrum), N is the number of molecules per cubic meter (equal to 2.545×1025 for air at 0° C and 1 atmosphere) and p_n is the depolarization factor (a dimensionless quantity, equal to 0.035 for air). Plugging in the values for air, together with the R, G, and B sample frequencies (650, 570, and 475 nm respectively) yields the following numbers:

$$\beta^{R}_{scAir} = 6.95 \times 10^{-6} m^{-1}$$

$$\beta^{G}_{scAir} = 1.18 \times 10^{-5} m^{-1}$$

$$\beta^{B}_{scAir} = 2.44 \times 10^{-5} m^{-1}$$

If your game has significantly different conditions (for example, high altitudes or a planet with very high air pressure), you can work out new values. The important thing to note here is that Rayleigh scattering has a very strong preference for shorter wavelengths, so blue is scattered much more than red. The Rayleigh phase function for air scattering is:

$$f_{Air}(\theta) = \frac{3}{16\pi}\left(1 + \cos^2\theta\right)$$

Figure 9 is the polar plot for this function. We can see that Rayleigh scattering is weakly directional and includes equal amounts of forward and backward scattering.

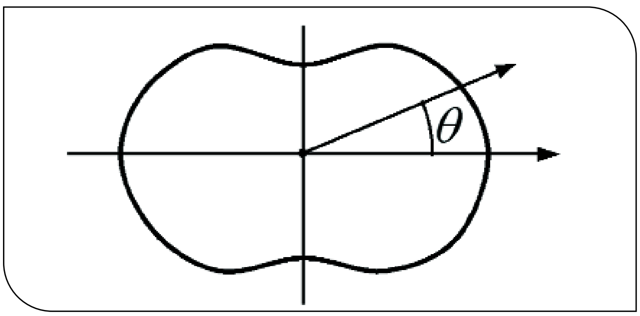


FIGURE 9. Rayleigh phase function polar plot.

Haze Particles and Mie Scattering

Particles much larger than air molecules (soot, dust, water vapor, ice crystals, and so on) are called haze particles. The β_{abHaze} coefficient can vary from 0 to about 5×10^{-5} m^{-1}; it is usually negligible unless there is a lot of pollution present (β_{abHaze} usually has no strong wavelength dependence).

A theoretical model which covers scattering for these particles was published by Gustav Mie in 1908, so this type of scattering is called Mie scattering. Mie equations are very complex and highly dependent on particle size. Haze particle size distributions in the real world are also highly varied, and it is difficult to model β_{scHaze} and $f_{haze}(\theta)$ analytically. Fortunately, many empirical measurements are available. The phase function can be approximated by the Henyey-Greenstein phase function (see For More Information):

$$f_{HG}(\theta) = \frac{(1-g)^2}{4\pi\left(1 + g^2 - 2g\cos(\theta)\right)^{3/2}}$$

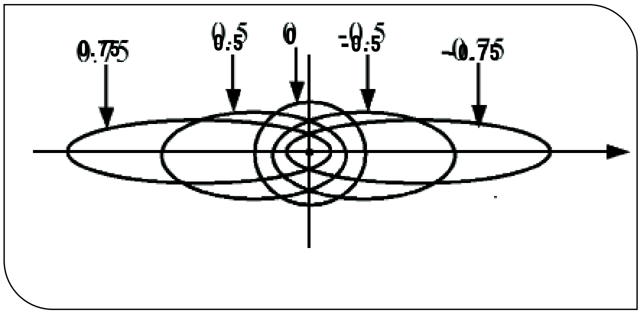


FIGURE 10. Henyey-Greenstein phase function polar plot.

The equation may look scary, but from Figure 10 we can see that this is simply the polar form of an ellipse, where is the eccentricity parameter (and also controls whether the ellipse points forward or backward). For most haze distributions, should be negative. As β_{scHaze} increases, g increases in magnitude and β_{scHaze} becomes more monochromatic. A group of typical values derived from empirical measurements can be seen in Table 1.

TABLE 1. Typical haze parameter values.

Description	β^{R}_{scHaze}	β^{G}_{scHaze}	β^{B}_{scHaze}	g
Light haze	2×10^{-5}	3×10^{-5}	4×10^{-5}	-1
Heavy haze	8×10^{-5}	10^{-4}	1.2×10^{-4}	-3
Light fog	9×10^{-4}	10^{-3}	1.1×10^{-3}	-10
Heavy fog	10^{-2}	10^{-2}	10^{-5}	-30

These values are for "normal" real-world environments. For more unusual environments, almost any values can be used; feel free to have a strongly colored absorption coefficient, or even a red-colored scattering coefficient (those even happen in the real world on rare occasions, thus the expression "once in a blue moon").

Aerial Perspective

Aerial perspective is caused by both extinction and in-scattering. Since the viewing rays are close to the ground, the constant density atmospheric model is a reasonable assumption and all the equations hold up. We treat the original (without scattering) color of the object as L_0 and multiply with $F_{ex}(s)$ and add $L_{in}(s,\theta)$ to get the final color. These factors can be precalculated into textures and rendered using functions of s and θ as texture coordinates, calculated per-vertex, or calculated per-pixel either on the fly or in a post-processing pass. In our implementation we chose to calculate them per-vertex in a vertex shader or vertex program.

This approach makes good use of modern hardware capabilities, enables changing parameters on-the-fly efficiently, and should work reasonably well even on older hardware with software vertex processing. Our implementation happens to use a DirectX 8 pixel shader for combining the factors with the original color, but advanced fragment processing is not necessarily required — depending on what else is happening in that pass, it could be possible simply to store the factors in the diffuse and specular vertex colors and combine them using the standard fragment pipeline. We can see L_0, $F_{ex}(s)$, and $L_{in}(s,\theta)$ being combined in Figure 11.

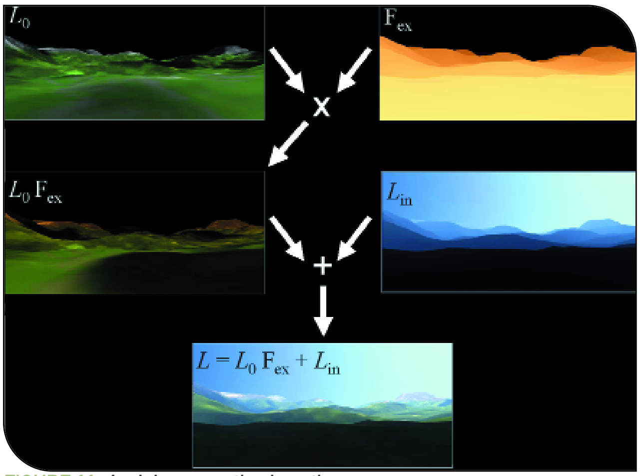


FIGURE 11. Aerial perspective in action.

Sunlight

The sun intensity factor E_{sun} is used for lighting the scene and for in-scattering calculations. We calculate it once a frame by applying an extinction factor $F_{ex}(s)$ to the sun's intensity in outer space E^0_{sun}. We could get an exact value for E^0_{sun} in watts per meter squared for R, G, and B, but we would still need to convert the resulting radiance values at the end to pixel values we can handle. Since this factor scales every radiance value in the scene, we will simply set it to the largest illumination value we can handle. On older systems this may be 1,1,1 (perhaps 2,2,2 with some careful use of overbrightening techniques), but on newer hardware and graphics engines we should be able to use larger values.

We will use the same parameters and model to calculate $F_{ex}(s)$ as we used for aerial perspective. However, it is not clear what to use for s, and the atmospheric density along the path is far from constant. We solve both problems at the same time by using optical length for s. Optical length is a distance defined as the integrated density along the ray divided by the density at ground level. So if we use our constant-density atmosphere model and use optical length for s, we will get the right extinction results.

The optical length of the atmosphere for air molecules is about 8.4 kilometers at the zenith (straight up). The exact optical length for haze particles depends on various factors, but a reasonable value to use is 1.25 kilometers (haze particles thin out faster with height than air molecules, so the optical length is shorter for them). For other directions the length follows Equation 5 (see Iqbal in For More Information):

$$l(\theta_s) = \frac{l_{Zenith}}{\cos(\theta_s) + 0.15(93.885 - \theta_s)^{-1.253}} \qquad \text{Eq. 5}$$

Note that in this equation θ_s is in degrees. Since we are using two different values of s (for air molecules and for haze), the equation for F_{ex} looks a little different:

$$F_{ex}(s_{Air}, s_{Haze}) = e^{-(\beta_{exAir} s_{Air} + \beta_{exHaze} s_{Haze})}$$

Sky Color

The sky color in all directions is the result of in-scattering. An accurate model would take multiple scattering into account and would be quite complex and expensive to evaluate, especially since it needs to be evaluated for many points every frame. In this case we go for consistency over accuracy and use the same scattering model for the sky as we used for the sun and other objects. A sky mesh is created, reasonably well tessellated, which conforms in size to the air molecule optical length values in all directions. Note that the sky mesh is always centered on the camera.

The sky mesh needs to be rendered with a similar vertex shader or vertex program to that used for the aerial perspective. The main difference is that here we have different values of s for the two particle types. Fortunately, the ratio between the two is a constant, so we can size the mesh to the air molecule optical lengths and then pass in the ratio as an additional parameter to the vertex shader. The vertex shader can then internally generate s_{Haze} from the vertex distance and the ratio constant. We assume that $s_{Air} > s_{Haze}$, so we can treat the atmosphere as two shells: the inner shell contains both air and haze and the outer contains only air. The resulting equation for L_{in} is:

$$L_{in}(s_{Air}, s_{Haze}, \theta) =$$

$$E_{sun}\left(\left(\frac{\beta_{scAir}(\theta) + \beta_{scHaze}(\theta)}{\beta_{exAir} + \beta_{exHaze}}\right)\left(1 - e^{-(\beta_{exAir} + \beta_{exHaze})s_{Haze}}\right)\right.$$

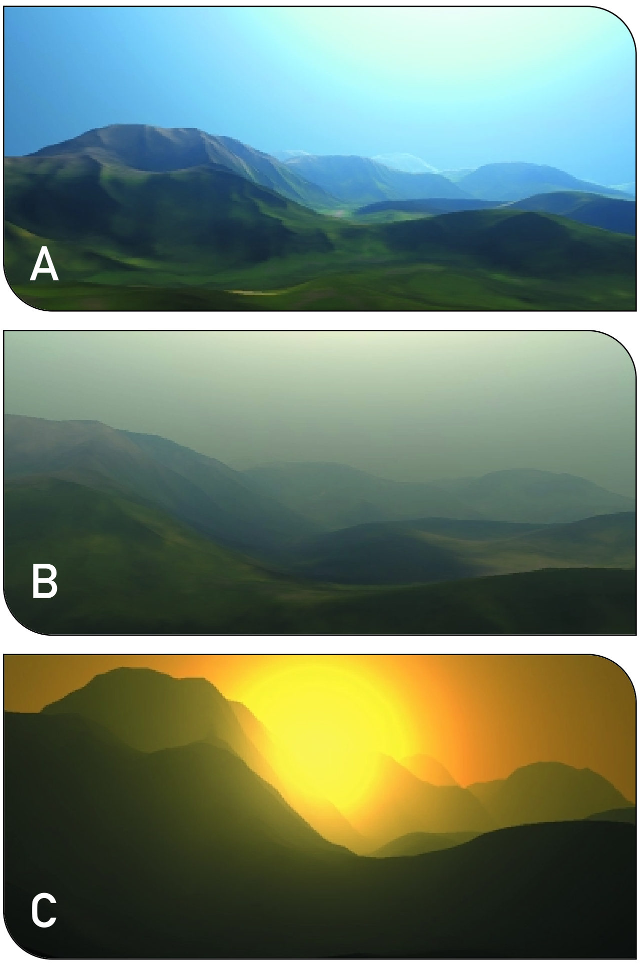
$$\left.\frac{\beta_{scAir}(\theta)}{\beta_{exAir}}\left(1 - e^{-\beta_{exAir}(s_{Air} - s_{Haze})}\right)e^{-(\beta_{exAir} + \beta_{exHaze})s_{Haze}}\right)$$

Since the sky uses a different vertex shader, we can also take advantage of the fact that the starting color is black (outer space) and skip the extinction factor calculations. We can also use a simpler fragment pipeline setup that just copies the interpolated in-scattered color.

Vertex Shader

In our implementation, we used a Direct3D 8.1 vertex shader. It should be straightforward to implement this in OpenGL as well, given access to the appropriate extensions. The vertex shader computes $F_{ex}(s)$ and $L_{in}(s,\theta)$, then writes them into `oD0` and `oD1`.

The inputs to the vertex shader are vertex position, transformation matrices, sunlight intensity factor E_{sun}, the sun direction (for computing $\cos\theta$), the various extinction and scattering coefficients, and the Henyey-Greenstein asymmetry factor g. The equations to compute $F_{ex}(s)$ and $L_{in}(s,\theta)$ are the same ones presented earlier in this article, and the vertex shader is a straightforward implementation of these equations. Our current version (which has not yet been thoroughly optimized) uses 33 instructions (not including macro expansions) and eight temporary registers.



FIGURES 12A–12C. Results of low haze with Rayleigh scattering (12a), high haze with Mie scattering (12b), and intermediate haze with a low sun angle (12c).

Results and Sample Demo

We can see some results in Figures 12a–c. Figure 12a shows a scene with a low concentration of haze particles, so Rayleigh scattering is predominant. The sun is high in the sky. Figure 12b shows a scene with a high concentration of haze particles (Mie scattering is predominant) and a high sun angle. In Figure 12c there is an intermediate haze concentration, and the sun angle is low. These images were rendered on a 600MHz Pentium III with an ATI Radeon 8500 at about 60 fps.

The sample demo (available at www.gdmag.com) requires graphics hardware that supports Direct3D pixel and vertex shaders and includes shader source code. The application has some sliders which control the various parameters and a fly-through demo mode.

Getting Light Right

With the right simplifications and assumptions, a full model of the interaction of light with the atmosphere can be expressed with a few reasonably simple equations. These equations can be evaluated in real time to add light scattering and absorption to any outdoor scene without unduly impacting performance. For future work we would like to make the sky color model more accurate, and also handle clouds within a physical scattering framework.

ACKNOWLEDGEMENTS

We would like to thank Kenny Mitchell for providing the terrain-rendering and lighting engine used as the basis for the demo, and Solomon Srinivasan for help with the fly-through mode.

FOR MORE INFORMATION

Blinn, J. F. "Light Reflection Functions for Simulation of Clouds and Dusty Surfaces." *Computer Graphics*, 16(3): 21–29, July 1982.

Henyey, L. G., and J. L. Greenstein. Diffuse Reflection in the Galaxy. *Astrophysical Journal*, vol. 93: 70, 1941.

Hoffman, N., and A. J. Preetham. "Rendering Outdoor Light Scattering in Real Time." *Proceedings of the Game Developers Conference*, March 2002.

Iqbal, M. An Introduction to Solar Radiation. Academic Press, 1983.

Klassen, R. V. "Modeling the Effect of the Atmosphere on Light." *ACM Transactions on Graphics*, 6(3): 215–237, July 1987.

Mie, G. "Bietage zur Optik truber Medien Speziell Kolloidaler Metallosungen." *Annallen der Physik*, 25(3): 377, 1908.

Preetham, A. J., P. Shirley, and B. E. Smits. "A Practical Analytic Model for Daylight." *Computer Graphics (Proceedings of SIGGRAPH 1999)*: 91–100, August 1999.

Strutt, J. W. (Lord Rayleigh). "On the Light from the Sky, Its Polarization and Colour." *Philosophical Magazine*, vol. 41: 107–120, 274–279, April 1871.