Atmospheric Rendering Notes

Jeff Boody | jeffboody@3dgesoftware.com

About

These notes are intended to augment the atmospheric rendering algorithm described by the Rendering Parametrizable Planetary Atmospheres with Multiple Scattering in Real-Time paper.

1 Mathematical Model

1.1 Wavelength of Light

The wavelength of light is a continuous value, however, we will select 3 discrete wavelengths to represent red, green and blue light.

The typical range of wavelengths include:

- $\lambda_r = (620, 750)$ nm
- $\lambda_g = (495, 570)$ nm
- $\lambda_b = (380, 500)$ nm

The Atmospheric Rendering implementation selected the wavelengths:

- $\lambda_r = 650$ nm
- $\lambda_g = 510$ nm
- $\lambda_b = 475$ nm

1.2 Atmospheric Boundary

The atmospheric rendering algorithm models the atmosphere as the region where the effects of the atmosphere are non-negligible, extending from the Earth's surface to an atmospheric boundary.

The Atmospheric Rendering implementation uses the following constants for these radii.

- Radius of the planet: $R_p = 6371000$ m
- Radius of the atmospheric boundary: $R_a = 6471000$ m

1.3 Atmospheric Particles

The atmospheric rendering algorithm models two types of particles.

- Smaller particles (e.g. air molecules such as oxygen, nitrogen and carbon dioxide) are modeled by Rayleigh scattering.
- Larger particles (e.g. ice crystals, water droplets, dust and air polution) are modeled by Mie scattering.

1.4 Density Function

The density function expresses the decrease in atmospheric density in dependence on h, the altitude of a point P over the ground.

Rayleigh/Mie density function, $p_{R,M}(h)$:

$$p_{R,M}(h) = \exp\left(-\frac{h}{H_{R,M}}\right)$$

Where $H_R \approx 8000m$ and $H_M \approx 1200m$ are the Rayleigh/Mie scale heights (e.g. the altitude where the density of particles scales down by a $\frac{1}{e}$ term).

1.5 Phase Function

The phase function expresses the relative amount of light that is scattered in a particular direction due to interactions with a particle.

Rayleigh scattering phase function, F_R:

$$F_R(\cos(\theta)) = \frac{3}{4}\left(1 + \cos^2(\theta)\right)$$

Where the scattering angle θ represents the angle between the incoming light ray and the scattered light ray.

$$\cos(\theta) = dot\left(L_{Incoming}, L_{Scattered}\right)$$

The Rayleigh scattering phase function F_R may be modified, in practice, to produce more natural results given simplifications introduced for the scattering intensity parameterization.

$$F_R(\cos(\theta)) = \frac{8}{10}\left(\frac{7}{5} + \frac{1}{2}\cos^2(\theta)\right)$$

Mie scattering phase function, F_M:

$$F_M(\cos(\theta)) = \frac{3\left(1-g^2\right)}{2\left(2+g^2\right)} \frac{\left(1+\cos^2(\theta)\right)}{\left(1+g^2+2g\cos(\theta)\right)^{\frac{3}{2}}}$$

The parameter g is an asymmetry factor denoting the width of the forward lobe. Typical values of g are in the range $(-0.75, -0.999)$ and the Accurate Atmospheric Scattering implementation uses $g = -0.99$.

1.6 Scattering Coefficient

The scattering coefficients $\beta_{R,M}$ represent the probability of light being scattered as it travels through the scattering medium.

The scattering coefficients $\beta_{R,M}$ are defined in terms of a Rayleigh/Mie particle polarizability constants $\alpha_{R,M}$.

$$\alpha_{R,M} = \frac{2\pi^2\left(n_e^2-1\right)^2}{3N_{R,M}^2}$$

$$\beta_R(\lambda) = 4\pi\frac{N_R}{\lambda^4}\alpha_R$$

$$\beta_M = 4\pi N_M \alpha_M$$

The following constants are used to compute the scattering coefficients:

- Index of refraction of the Earth's atmosphere at sea level: $n_e = 1.0003$
- Rayleigh particles molecular number density of the Earth's atmosphere at sea level: $N_R = 2.454e25$
- Mie particles molecular number density of the Earth's atmosphere at sea level: $N_M = UNDEFINED$

The Atmospheric Rendering implementation and the Efficient and Dynamic Atmospheric Scattering paper use the following equations to compute the scattering coefficients. The equation for $\beta_R(\lambda)$ matches the previous definition after expanding α_R. See the Efficient and Dynamic Atmospheric Scattering paper for an explanation of β_M.

$$\beta_R(\lambda) = \frac{8\pi^3 \left(n_e^2 - 1\right)^2}{3N_R\lambda^4}$$

$$\beta_R(r, g, b) = (6.55e - 6, 1.73e - 5, 2.30e - 5)$$

$$\beta_M = 2e - 6$$

1.7 Transmittance

The transmittance, or optical length, $t_{R,M}(P_1P_2, \lambda)$ expresses the amount of attenuated light after it passes between two points through the scattering medium.

$$t_{R,M}(P_1P_2, \lambda) = \beta_{R,M}(\lambda) \int_{P_1}^{P_2} p_{R,M}(s)\, ds$$

Attenuation is a consequence of out-scattering in the scattering medium.

1.8 Single-scattering

The single-scattering equation $I_{S_{R,M}}^{(1)}$ describes the intensity of light that reaches an observer P_0 looking in the direction V after exactly one scattering event.

$$I_{S_{R,M}}^{(1)}(P_0, V, L, \lambda) = I_I(\lambda) \cdot F_{R,M}(\cos(\theta)) \cdot \frac{\beta_{R,M}(\lambda)}{4\pi} \cdot$$

$$\cdot \int_{P_a}^{P_b} p_{R,M}(h) \cdot \exp\left(-t_{R,M}(PP_c, \lambda) - t_{R,M}(P_aP, \lambda)\right) ds$$

Where the following parameters are defined:

- $I_I(\lambda)$: The spectral intensity of incident light from the Sun may be modeled using white light, $I_I(r, g, b) = (1.0, 1.0, 1.0)$.
- P: Sample point parameterized by s, $P = P_a + s(P_b - P_a)$
- V: Viewing direction, $V = \mathit{nomalize}(P_b - P_a)$
- L: Direction from the Sun
- θ: Scattering angle, $\cos(\theta) = \mathit{dot}(L, -V)$
- h: Altitude of the point P

- P_a: First point along V where the atmosphere is nonzero (e.g. P_0 or the atmospheric boundary).
- P_b: Last point along V where the atmosphere is nonzero (e.g. the atmospheric boundary or the Earth's surface).
- P_c: The intersection point with the upper atmosphere boundary of the ray from P in the direction $-L$ to the Sun.

The phase function $F_{R,M}$ can be excluded from integration if we assume all light rays coming from the Sun are parallel.

If the sample point P is shadowed by the Earth then it does not contribute to the single-scattering intensity.

The total intensity of the single-scattering light:

$$I_S^{(1)} = I_{S_R}^{(1)} + I_{S_M}^{(1)}$$

1.9 Multiple-scattering

The multiple-scattering equation $I_{S_{R,M}}^{(k)}$ describes the intensity of light that reaches an observer P_0 looking in the direction V for the kth scattering event.

$$I_{S_{R,M}}^{(k)}\left(P_0, V, L, \lambda\right) = \frac{\beta_{R,M}\left(\lambda\right)}{4\pi} \cdot$$

$$\cdot \int_{P_a}^{P_b} G_{R,M}^{(k-1)}\left(P, V, L, \lambda\right) \cdot p_{R,M}\left(h\right) \cdot \exp\left(-t_{R,M}\left(P_a P, \lambda\right)\right) ds$$

Where the gather-scattering equation $G_{R,M}^{(k)}$ describes the intensity of light that reaches the point P that is scattered from all directions ω towards the observer for the kth order scattering event.

$$G_{R,M}^{(k)}\left(P, V, L, \lambda\right) = \int_{4\pi} F_{R,M}\left(\cos\left(\theta\right)\right) \cdot I_{S_{R,M}}^{(k)}\left(P, \omega, L, \lambda\right) d\omega$$

Where the following parameters are defined:

- ω: Gathering direction for the kth order scattered light source
- θ: Scattering angle, $\cos\left(\theta\right) = dot\left(-\omega, -V\right)$

The first order of $I_{S_{R,M}}^{(k)}$ is initialized with the single-scattering output $I_{S_{R,M}}^{(1)}$ while subsequent orders of $I_{S_{R,M}}^{(k)}$ are computed iteratively using the previous output $I_{S_{R,M}}^{(k-1)}$.

Observe that the light vector L is used in computation of the phase function $F_{R,M}$ for the single-scattering equation $I_{S_{R,M}}^{(1)}$, but not for the gather-scattering equation $G_{R,M}^{(k)}$. As a result, the phase function $F_{R,M}$ in the gather-scattering equation $G_{R,M}^{(k)}$ may not be excluded from integration because θ depends on the integration variable ω. The light vector L is also required to parameterize the Sun-Zenith angle for all scattering orders as shown in subsequent sections.

The total intensity of the multiple-scattering light:

$$I_S = \sum_{k=1}^{K} I_{S_R}^{(k)} + I_{S_M}^{(k)} = I_{S_R} + I_{S_M}$$

2 Implementation

2.1 Scattering Equation Factorization

The following derivation factors the constant phase function and the spectral intensity of incident light from the scattering equations.

The phase function parameter $cos\left(\theta\right)$ has been replaced with the dot product equivalent to distinguish between constant phase function $F_{R,M}\left(dot\left(L,-V\right)\right)$ and integration dependent phase function $F_{R,M}\left(dot\left(-\omega,-V\right)\right)$.

The factored single-scattering equation $\bar{I}_{S_{R,M}}^{(1)}$:

$$\bar{I}_{S_{R,M}}^{(1)}\left(P_0,V,L,\lambda\right) = \frac{\beta_{R,M}\left(\lambda\right)}{4\pi}\cdot$$

$$\cdot\int_{P_a}^{P_b} p_{R,M}\left(h\right)\cdot\exp\left(-t_{R,M}\left(PP_c,\lambda\right)-t_{R,M}\left(P_aP,\lambda\right)\right)ds$$

Where $I_{S_{R,M}}^{(1)}$:

$$I_{S_{R,M}}^{(1)}\left(P_0,V,L,\lambda\right) = I_I\left(\lambda\right)\cdot F_{R,M}\left(dot\left(L,-V\right)\right)\cdot \bar{I}_{S_{R,M}}^{(1)}\left(P_0,V,L,\lambda\right)$$

The factored multiple-scattering equation $\bar{I}_{S_{R,M}}^{(k)}$:

$$\bar{I}_{S_{R,M}}^{(k)}\left(P_0,V,L,\lambda\right) = \frac{\beta_{R,M}\left(\lambda\right)}{4\pi}\cdot$$

$$\cdot \int_{P_a}^{P_b} \bar{G}_{R,M}^{(k-1)} (P, V, L, \lambda) \cdot p_{R,M} (h) \cdot \exp (-t_{R,M} (P_a P, \lambda)) \, ds$$

Where $I_{S_{R,M}}^{(k)}$:

$$I_{S_{R,M}}^{(k)} (P_0, V, L, \lambda) = I_I (\lambda) \cdot F_{R,M} (dot (L, -V)) \cdot \bar{I}_{S_{R,M}}^{(k)} (P_0, V, L, \lambda)$$

The factored gather-scattering equation $\bar{G}_{R,M}^{(k)}$:

$$\bar{G}_{R,M}^{(k)} (P, V, L, \lambda) = \int_{4\pi} F_{R,M} (dot (-\omega, -V)) \cdot \bar{I}_{S_{R,M}}^{(k)} (P, \omega, L, \lambda) \, d\omega$$

Where $G_{R,M}^{(k)}$:

$$G_{R,M}^{(k)} (P, V, L, \lambda) = I_I (\lambda) \cdot F_{R,M} (dot (L, -V)) \cdot \bar{G}_{R,M}^{(k)} (P, V, L, \lambda)$$

The total scattering intensity using the factored equations:

$$I_S = I_I (\lambda) \cdot \left(F_R (dot (L, -V)) \cdot \bar{I}_{S_R} + F_M (dot (L, -V)) \cdot \bar{I}_{S_M} \right)$$

This factorization includes two important properties that facilitate an efficient rendering implementation. First, the wavelength dependent components, $I_I (\lambda)$ and $\bar{I}_{S_R}$, may be separated from the wavelength independent component $\bar{I}_{S_M}$. The spectral intensity of incident light $I_I (\lambda)$ may be applied directly in the fragment shader and the factored scattering intensity $\bar{I}_S$ may be determined by performing a single 3D texture fetch. The Rayleigh factored scattering intensity $\bar{I}_{S_R}$ is stored in the RGB channels while the Mie factored scattering intensity $\bar{I}_{S_M}$ is stored in the alpha channel. Second, the constant phase function $F_{R,M} (dot (L, -V))$ may be applied directly in the fragment shader which partially accounts for the omitted Sun-View Azimuth parameter. These properties are fundamental to the scattering intensity parameterization that is described in the next section.

2.2 Scattering Intensity Parameterization

To precompute every scattering intensity from every position $P_0 (x, y, z)$, in every viewing direction $V (x, y, z)$ and every light direction $L (x, y, z)$ would require 9 parameters. However, by taking advantage of symmetries and making a few assumptions the parameter count may be reduced to 4 scalar parameters.

- Altitude: $h = (0, H_a)$ where $H_a = R_a - R_p$

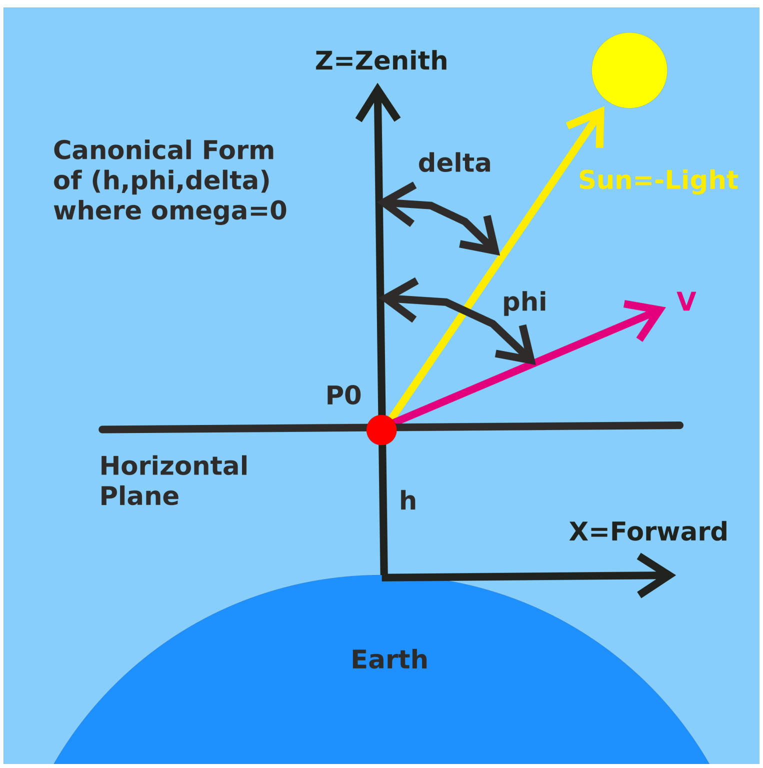
- View-Zenith Angle: $\phi = (0, \pi)$ where $\cos(\phi) = dot(V, Zenith)$
- Sun-Zenith Angle $\delta = (0, \pi)$ where $\cos(\delta) = dot(-L, Zenith)$
- Sun-View Azimuth $\omega = (0, \pi)$ where $cos(\omega) = dot(proj(V), proj(-L))$

Where the Zenith is for the point P_0 and $proj(X)$ takes a unit vector and projects it onto the horizontal plane perpendicular to the Zenith:

$$Zenith = normalize(P_0)$$

$$proj(X) = normalize(X - dot(X, Zenith) \cdot Zenith)$$

The following diagram shows canonical form of the scattering intensity parameterization.



The parameter h may be converted to the canonical form P_0.

$$P_0 = (0, 0, h + R_p)$$

The parameters (h, ϕ, δ) may be converted to the canonical form vectors V and L by using the spherical coordinate system.

$$x = r \sin(\Theta) \cos(\Phi)$$

$$y = r \sin(\varTheta) \sin(\varPhi)$$

$$z = r \cos(\varTheta)$$

Therefore V, Sun and L are defined where $r = 1$, $\varTheta = \phi$ for V, $\varTheta = \delta$ for Sun and $\varPhi = \omega = 0$.

$$V = (\sin(\phi), 0, \cos(\phi))$$

$$Sun = (\sin(\delta), 0, \cos(\delta))$$

$$L = -Sun$$

The parameters (h, ϕ, δ) may be converted to 3D texture coordinates (u, v, w) for array lookups.

$$u = \sqrt{\left(h^2 - R_p^2\right) / \left(R_a^2 - R_p^2\right)}$$

$$v = (1 + \cos(\phi))/2$$

$$w = \left(1 - e^{-2.8\cos(\delta) - 0.8}\right) / \left(1 - e^{-3.6}\right)$$

As described by the Precomputed Atmospheric Scattering paper, the equations for u and w use ad-hoc non-linear equations so as to get better precision near the ground and for Sun-Zenith angles near 90 degrees. As a result, the equation for w may have negative values (for $\delta \gtrsim 105$ degrees) and should be clamped to the range $[0, 1]$.

The 3D texture coordinates may be converted back to parameters as follows.

$$h = \sqrt{\left(R_a^2 - R_p^2\right) u^2 + R_p^2}$$

$$\phi = acos(2v - 1)$$

$$\delta = acos\left(-\left(ln\left(1-\left(1-e^{-3.6}\right)w\right)+0.8\right)/2.8\right)$$

The 3D array indices (x, y, z) with dimensions $(width, height, depth)$ may be converted to 3D texture coordinates (u, v, w) as follows.

$$(u, v, w) = \left(\frac{x}{width-1}, \frac{y}{height-1}, \frac{z}{depth-1}\right)$$

The 3D array indices (x, y, z) may be interpreted as a 1D array index (i) as follows.

$$i = x + y * width + z * width * height$$

The Rendering Parametrizable Planetary Atmospheres with Multiple Scattering in Real-Time paper uses the texture dimensions.

$$(width, height, depth) = (32, 256, 32)$$

Since the h and δ parameter conversions optimize the parameter space, fewer samples are required to represent the width and depth dimensions.

To reduce the parameter count, the paper proposes to omit the parameter ω from precomputation. This omission causes uniformity of the atmospheric color with respect to ω and is primarily visible during sunsets in parts of the sky where there is no direct illumunation. Two techniques are proposed to address the omission of ω. First, the modified Rayleigh scattering phase function F_R ensures that the darkest area of the sky during sunset is on the opposite side of the sky from the Sun. Second, the parameter ω is dependent on the scattering angle θ. By evaluating the constant phase function $F_{R,M}\left(dot\left(L, -V\right)\right)$ during rendering, we are able to reduce the uniformity of the atmospheric color with respect to ω.

Note: The the Sun-View Azimuth ω is a separate variable from the gathering direction ω described in other sections.

2.3 Numerical Integration

The atmospheric integrals must be solved using numerical integration due to complexity of the equations.

The trapezoidal rule may be applied to approximate the area of the region under the graph of the function $f(x)$ as a trapezoid.

$$\int_a^b f(x)\,dx \approx (b-a)\cdot\frac{1}{2}\left(f(a)+f(b)\right)$$

Where the accuracy of the solution may be improved by reducing the step size ∇x_k.

$$\int_a^b f(x)\,dx \approx \sum_{k=1}^{N} \frac{f(x_{k-1}) + f(x_k)}{2} \nabla x_k$$

The Rendering Parametrizable Planetary Atmospheres with Multiple Scattering in Real-Time paper uses $N = 30$ steps for each integration with a variable step size ∇x_k.

References

- Rendering Parametrizable Planetary Atmospheres with Multiple Scattering in Real-Time
  - http://www.klayge.org/material/4_0/Atmospheric/Rendering%20Parametrizable%20Planetary%2 Time.pdf
- Display of The Earth Taking into Account Atmospheric Scattering
  - http://nishitalab.org/user/nis/cdrom/sig93_nis.pdf
- Real-Time Rendering of Planets with Atmospheres
  - https://naos-be.zcu.cz/server/api/core/bitstreams/235c2478-5bff-4510-b22f-d5d93001d574/content
- Accurate Atmospheric Scattering
  - https://developer.nvidia.com/gpugems/gpugems2/part-ii-shading-lighting-and-shadows/chapter-16-accurate-atmospheric-scattering
- Atmospheric Rendering
  - https://github.com/danielshervheim/atmosphere
- Efficient and Dynamic Atmospheric Scattering
  - https://publications.lib.chalmers.se/records/fulltext/203057/203057.pdf
- Precomputed Atmospheric Scattering
  - http://www.klayge.org/material/4_0/Atmospheric/Precomputed%20Atmospheric%20Scattering.p
  - http://evasion.inrialpes.fr/~Eric.Bruneton/
  - https://github.com/ebruneton/precomputed_atmospheric_scattering
- Deferred Rendering of Planetary Terrains with Accurate Atmospheres

  - https://www.gamedevs.org/uploads/deferred-rendering-of-planetary-terrains-with-accurate-atmospheres.pdf

- Solar System
  - https://github.com/SebLague/Solar-System
  - https://www.youtube.com/watch?v=DxfEbulyFcY