Atmosphere Plus

Jeff Boody | jeffboody@3dgesoftware.com

About

These notes describe the atmospheric rendering theory and implementation for the Atmosphere Plus demo.

1 Background

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 Spectral Intensity of Incident Light

The Atmospheric Rendering implementation defines the spectral intensity of incident light $I_I(\lambda)$ from the Sun.

$$I_I(\lambda) = SpectralIrridence(\lambda) \cdot Exposure \cdot SpectralToRGB(\lambda)$$

The spectral irradiance measures the power density of solar radiation at specific wavelengths (from measured values found in tables).

$$SpectralIrridence\,(r,g,b) = (0.1526, 0.191, 0.208)$$

The exposure represents the amount of light exposure which scales the overall intensity.

$$Exposure = 1.0$$

The spectral to RGB constants are used to convert specific wavelengths into high dynamic range RGB values. These constants were calculated by converting wavelengths into CIE XYZ color space, then converting XYZ to RGB, followed by normalization and scaling.

$$SpectralToRGB\,(r,g,b) = (133.3209, 88.51855, 112.7552)$$

The spectral intensity of incident light $I_I\,(r,g,b)$:

$$I_I\,(r,g,b) = (20.344770, 16.907042, 23.453083)$$

1.3 Tone Mapping and Gamma Correction

The atmospheric simulation inherently outputs high dynamic range (HDR) intensities that must be transformed for display on standard devices as the final step.

Tone mapping transforms the high dynamic range (HDR) intensities such that the colors are mapped to the displayable range $(0.0, 1.0)$ while preserving contrast in low-intensity areas and compressing high-intensity values.

$$ToneMapping\,((r,g,b)_{HDR}) = 1.0 - \exp\left(-\,(r,g,b)_{HDR}\right)$$

Gamma correction helps to distribute the compressed values more evenly across the perceptual range of human vision.

$$GammaCorrection(r,g,b) = (r,g,b)^{1.0/2.2}$$

The combined tone mapping and gamma correction transformation.

$$(r,g,b) = \left(1.0 - \exp\left(-\,(r,g,b)_{HDR}\right)\right)^{1.0/2.2}$$

1.4 Atmospheric Boundary

The atmospheric simulation 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 Scalable and Production Ready Sky and Atmosphere Rendering Technique paper uses the following constants for these radii.

- Radius of the planet: $R_p = 6360000$ m
- Radius of the atmospheric boundary: $R_a = 6460000$ m

Note: Special care may be required to perform some mathematical operations on floating point values of these magnitudes. To preserve numerical precision, convert single precision floats to double precision floats prior to performing dangerous mathematical operations. Once complete, the result may be converted back to single precision floats. For example, the *normalize* (P) function requires computing the square root of the sum of squares of floating point values of these magnitudes.

1.5 Atmospheric Particles

The atmospheric simulation 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.

1.6 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).

1.7 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:

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

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

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

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.

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

Mie scattering phase function, F_M:

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

The parameter g is an asymmetry factor denoting the width of the forward lobe and is in the range $(-1, 1)$.

The Scalable and Production Ready Sky and Atmosphere Rendering Technique paper uses $g = 0.8$.

1.8 Scattering, Absorption and Extinction Coefficients

The scattering/absorption/extinction coefficients represents the probability of light being scattered/absorbed/scattered+absorbed as it travels through a medium.

The Scalable and Production Ready Sky and Atmosphere Rendering Technique paper uses the following coefficients.

$$\beta_R^s\left(r, g, b\right) = \left(5.802e-6, 13.558e-6, 33.1e-6\right)$$

$$\beta_R^a\left(r, g, b\right) = \left(0, 0, 0\right)$$

$$\beta_M^s = 3.996e-6$$

$$\beta_M^a = 4.40e-6$$

$$\beta_{R,M}^e = \beta_{R,M}^s + \beta_{R,M}^a$$

1.9 Numerical Integration

The atmospheric simulation requires the computation of integrals that must be approximated using numerical integration due to complexity of the equations.

The trapezoidal rule may be used to approximate 1D integrals of the function $f(x)$.

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} \frac{f(x_{i-1}) + f(x_i)}{2} \cdot dx$$

Where the parameters are defined:

$$dx = \frac{x_n - x_0}{n}$$

$$x_i = x_0 + i \cdot dx$$

The endpoints of the 1D integral are defined as $x_0 = a$ and $x_n = b$.

The trapezoidal rule may be extended to approximate 2D integrals of the function $f(x, y)$.

$$\int_a^b \int_c^d f(x,y)\,dy \cdot dx \approx$$

$$\frac{1}{4} dx \cdot dy \cdot (f(x_0, y_0) + f(x_n, y_0) + f(x_0, y_m) + f(x_n, y_m) +$$

$$2\sum_{j=1}^{n-1} f(x_j, y_0) + 2\sum_{j=1}^{n-1} f(x_j, y_m) + 2\sum_{i=1}^{m-1} f(x_0, y_i) + 2\sum_{i=1}^{m-1} f(x_n, y_i) +$$

$$4\sum_{i=1}^{m-1}\sum_{j=1}^{n-1} f(x_j, y_i)\Bigg)$$

Where the parameters are defined:

$$dx = \frac{x_n - x_0}{n}$$

$$dy = \frac{y_m - y_0}{m}$$

$$x_j = x_0 + j \cdot dx$$

$$y_i = y_0 + i \cdot dy$$

The edges of the 2D integral are defined as $x_0 = a$, $x_n = b$, $y_0 = c$ and $y_m = d$.

Note: The trapezoidal rule for 2D integrals can also be separated into two passes by applying the 1D rule in the horizontal and vertical directions. The implementation using separate passes may be faster, however, the accuracy may be lower due to strong coupling between the x and y variables. As a result, the 2D approach is generally preferred for its robustness unless the function is well-understood and separability is confirmed.

1.10 Transmittance

The transmittance function $T(P_1, P_2, \lambda)$ expresses the amount of attenuated light after it passes between two points through a medium.

$$T(P_1, P_2, \lambda) = \exp(-t(P_1, P_2, \lambda))$$

$$t(P_1, P_2, \lambda) = \beta^e_{R,M}(\lambda) \int_{P_1}^{P_2} p_{R,M}(h(P))\, ds$$

Where the parameters are defined:

- P: Sample point parameterized by s, $P = P_1 + s(P_2 - P_1)$
- h: Height is parameterized by P, $h(P) = |P| - R_p$

The outputs are a 4-component vectors $(T_{R_r}, T_{R_g}, T_{R_b}, T_M)$ and $(t_{R_r}, t_{R_g}, t_{R_b}, t_M)$.

The Rendering Parametrizable Planetary Atmospheres with Multiple Scattering in Real-Time paper uses numerical integration with $n = 30$ steps to compute the transmittance.

The light ray may change direction as it interacts with particles while passing through a medium. Consider a light ray which passes between the points P_1, P_2, P_3 where the point P_2 represents an interaction which changes the direction of the light ray. The transmittance between the points P_1, P_2, P_3 may be computed as the product of the transmittance between each segment.

$$T\left(P_1, P_2, \lambda\right) \cdot T\left(P_2, P_3, \lambda\right) = \exp\left(-t\left(P_1, P_2, \lambda\right)\right) \cdot \exp\left(-t\left(P_2, P_3, \lambda\right)\right)$$

The implementation may utilize the laws of exponents rule for product of powers to optimize the computation.

$$a^m \cdot a^n = a^{m+n}$$

Where the functions $t\left(\right)$ may be rearranged in order to group similar terms.

$$T\left(P_1, P_2, \lambda\right) \cdot T\left(P_2, P_3, \lambda\right) = \exp\left(-t\left(P_1, P_2, \lambda\right) - t\left(P_2, P_3, \lambda\right)\right)$$

2 Sky Rendering

2.1 Single-scattering

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

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

$$\cdot \int_{P_a}^{P_b} p_{R,M}\left(h\left(P\right)\right) \cdot T\left(P_c, P, \lambda\right) \cdot T\left(P, P_a, \lambda\right) ds$$

Where the following parameters are defined:

- P: Sample point parameterized by s, $P = P_a + s\left(P_b - P_a\right)$
- h: Height is parameterized by P, $h\left(P\right) = |P| - R_p$
- V: Viewing direction, $V = nomalize\left(P_b - P_a\right)$
- L: Direction of light from the Sun
- θ: Scattering angle, $\cos\left(\theta\right) = dot\left(L, -V\right)$
- 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 nearest intersection point along the ray from P in the direction of the Sun (e.g. the atmosphere boundary or the Earth's surface). If the ray intersects the Earth then the point P is shadowed by the Earth and does not contribute to the single-scattering intensity. However, the point P may still contribute to the multiple scattering intensity as described in the next section.

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

The total intensity of the single-scattering light:

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

2.2 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.

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

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

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.

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

Where the following parameters are defined:

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

The gather-scattering equation $G_{R,M}^{(k)}$ requires the computation of an integral over all directions ω. This can be accomplished by rewriting the equation using

a spherical coordinate system. For example, a function $f(\omega)$ can be integrated over every direction by the double integral.

$$\int_{4\pi} f(\omega)\, d\omega = \int_{\Phi=0}^{2\pi} \int_{\Theta=0}^{\pi} f(\Theta, \Phi) \sin(\Theta)\, d\Theta \cdot d\Phi$$

Where the gathering direction ω and element area $d\omega$ are defined:

$$\omega = (\sin(\Theta)\cos(\Phi), \sin(\Theta)\sin(\Phi), \cos(\Theta))$$

$$d\omega = \sin(\Theta)\, d\Theta \cdot d\Phi$$

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

Observe that the light vector L is used in computation of the phase function $F_{R,M}$ for the single-scattering equation $I^{(1)}_{S_{R,M}}$, but not for the gather-scattering equation $G^{(k)}_{R,M}$. As a result, the phase function $F_{R,M}$ in the gather-scattering equation $G^{(k)}_{R,M}$ 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:

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

2.3 Scattering Equation Factorization

The following derivation factors the constant phase function $F_{R,M}(dot(L, -V))$ and the spectral intensity of incident light $I_I(\lambda)$ from the scattering equations.

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

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

$$\cdot \int_{P_a}^{P_b} p_{R,M}(h(P)) \cdot T(P_c, P, \lambda) \cdot T(P, P_a, \lambda)\, ds$$

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

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

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

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

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

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

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

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

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

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

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

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.

2.4 Scattering Intensity Parameterization

2.4.1 Canonical Form

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: $h \in [0, H_a]$ where $H_a = R_a - R_p$
- View-Zenith Angle: $\phi \in [0, \pi]$ where $\cos(\phi) = dot(V, Zenith)$
- Sun-Zenith Angle $\delta \in [0, \pi]$ where $\cos(\delta) = dot(-L, Zenith)$
- Sun-View Azimuth $\omega \in [0, \pi]$ where $cos(\omega) = dot(proj(V), proj(-L))$

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

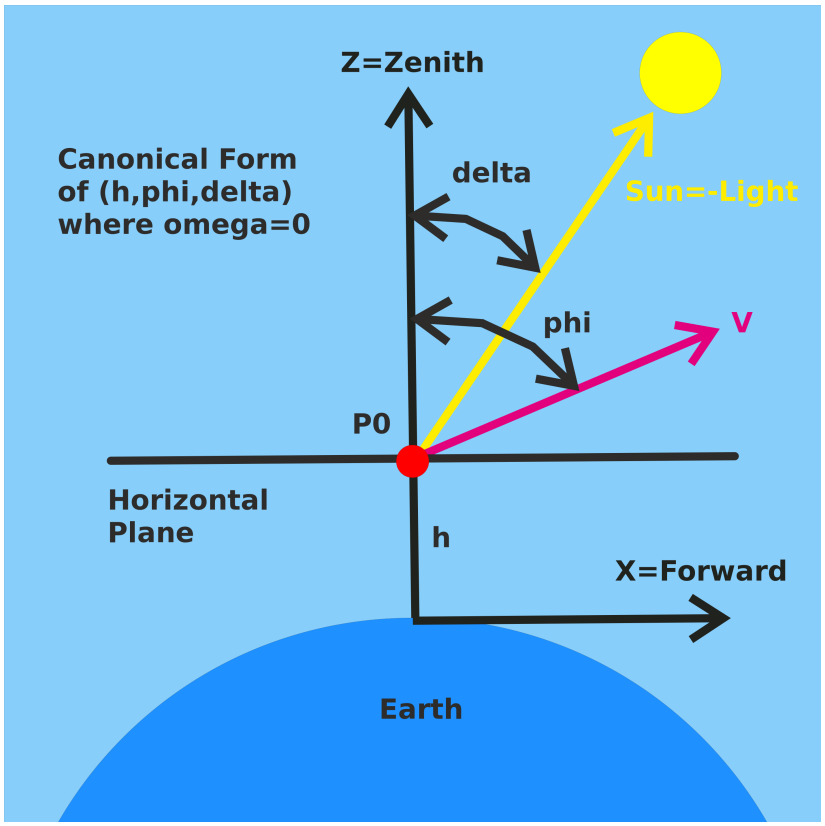
The Zenith is the up vector for the point P_0.

$$Zenith = normalize(P_0)$$

The $proj(X)$ function projects the unit vector X onto the horizontal plane perpendicular to the Zenith.

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

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.

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

$$y = r \sin(\Theta) \sin(\Phi)$$

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

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

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

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

$$L = -Sun$$

2.4.2 Linear Mapping

The parameters $(h, \cos(\phi), \cos(\delta))$ may be converted to 3D texture coordinates (u, v, w) for shader lookups.

The most straightforward mapping described by the Efficient and Dynamic Atmospheric Scattering paper is a linear mapping.

$$u = \frac{h}{H_a}$$

$$v = \frac{\cos(\phi) + 1}{2}$$

$$w = \frac{\cos(\delta) + 1}{2}$$

The linear mappings may be converted from 3D texture coordinates as follows.

$$h = u \cdot H_a$$

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

$$\cos(\delta) = 2w - 1$$

2.4.3 Nonlinear Mapping

Linear mappings are typically replaced with nonlinear mappings in order to optimize the parameterization space. This is achieved by eliminating unused values and improving precision for critical values by compressing less critical values. The Atmospheric Rendering implementation proposed the following nonlinear mapping.

$$u = \sqrt{\frac{h}{H_a}}$$

$$v = \frac{sign\left(\cos\left(\phi\right)\right) \cdot \left|\cos\left(\phi\right)\right|^{\frac{1}{3}} + 1}{2}$$

$$w = \frac{sign\left(\cos\left(\delta\right)\right) \cdot \left|\cos\left(\delta\right)\right|^{\frac{1}{3}} + 1}{2}$$

The nonlinear mappings may be converted from 3D texture coordinates as follows.

$$h = u^2 \cdot H_a$$

$$\cos\left(\phi\right) = sign\left(2v - 1\right) \cdot \left|2v - 1\right|^3$$

$$\cos\left(\delta\right) = sign\left(2w - 1\right) \cdot \left|2w - 1\right|^3$$

2.4.4 Power Mapping

The linear and nonlinear mappings may be generalized for an arbitrary power.

$$u\left(h, p_u\right) = \left(\frac{h}{H_a}\right)^{\frac{1}{p_u}}$$

$$v\left(\phi, p_v\right) = \frac{sign\left(\cos\left(\phi\right)\right) \cdot \left|\cos\left(\phi\right)\right|^{\frac{1}{p_v}} + 1}{2}$$

$$w\left(\delta, p_w\right) = \frac{sign\left(\cos\left(\delta\right)\right) \cdot \left|\cos\left(\delta\right)\right|^{\frac{1}{p_w}} + 1}{2}$$

The power mappings may be converted from 3D texture coordinates as follows.

$$h = u^{p_u} \cdot H_a$$

$$\cos\left(\phi\right) = sign\left(2v - 1\right) \cdot \left|2v - 1\right|^{p_v}$$

$$\cos\left(\delta\right) = sign\left(2w - 1\right) \cdot \left|2w - 1\right|^{p_w}$$

2.4.5 Weighted Power Mapping

The power mapping based approaches have several problems which lead to inefficient representation of the View-Zenith Angle.

- The horizon ϕ_H increases from 90 degrees at ground level towards 180 degrees as the height increases. However, the power mapping approach fixes the horizon ϕ_H at 90 degrees for all heights which causes incorrect angles to be compressed as the height increases.
- The horizon also vanishes quickly as the height increases. As a result, the View-Zenith Angle ϕ may be compressed around the horizon using a low power p_v near gound level but requires a higher power p_v as the height increases. Since the power mapping approach uses a single low power p_v, compression is ineffective as the height increases.
- A large percentage of the texture space is dedicated to parameters which represent little or no atmosphere. For example, at ground level there is no atmosphere when V is downwards.

The weighted power mapping addresses these problems by incorporating the height h into the parameterization for the v coordinate.

The horizon ϕ_H is defined by a triangle consisting of a point at the center of the planet, a point tangent to the planet and a point at the observer. The function $\cos(\phi_H)$ may be derived given the trigonometric identies $\cos(\theta) = \frac{adj}{hyp}$, $a^2 + b^2 = c^2$ and the fact that the horizon $\phi_H \geq 90$ (e.g. $\cos(\phi_H) \leq 0$).

$$\cos(\phi_H) = -\frac{\sqrt{(R_p + h)^2 - R_p^2}}{R_p + h}$$

The cosine of the horizon angle $\cos(\phi_H)$ and the horizontal angle $\cos(\phi_{90}) = 0$ are used to define several regions (U: upper atmosphere, L: lower atmosphere, S: surface) which may be parameterized independently.

In general, each region is defined by a weight $w_i \in (0, 1)$, a power mapping $f(\cos(\phi), p_i) \in (0, 1)$ and an offset $o_i \in (0, 1)$.

$$v_i = w_i \cdot \left(\frac{\cos(\phi) - \cos(\phi_{min})}{\cos(\phi_{max}) - \cos(\phi_{min})} \right)^{\frac{1}{p_i}} + o_i$$

The power base will never be negative and therefore the sign and abs functions are not required.

The upper atmosphere region is defined by the angles, $0 \leq \cos(\phi) \leq 1$.

$$v_U(\phi) = w_U \cdot (\cos(\phi))^{\frac{1}{p_U}} + (w_L + w_S)$$

The lower atmosphere region is defined by the angles, $\cos(\phi_H) \leq \cos(\phi) < 0$. Since $\cos(\phi_H) = 0$ when $h = 0$, include $\epsilon = 1e-5$ for numerical stability. Also, observe that $w_L = 0$ when $h = 0$ since $u(h, p_u) = 0$.

$$v_L(\phi) = w_L \cdot \left(\frac{\cos(\phi) - \cos(\phi_H)}{\max(-\cos(\phi_H), \epsilon)} \right)^{\frac{1}{p_L}} + w_S$$

The surface region is defined by the angles, $-1 \leq \cos(\phi) < \cos(\phi_H)$. The one-minus term in v_S ensures continuity between v_L and v_S after crossing the horizon ϕ_H.

$$v_S(\phi) = w_S \cdot \left(1 - \left(\frac{\cos(\phi) - \cos(\phi_H)}{-1 - \cos(\phi_H)} \right)^{\frac{1}{p_S}} \right)$$

The weights define the amount of texture space allocated for each region.

$$w_U = 1 - w_L - w_S$$

$$w_L = w_L^1 \cdot u$$

$$w_S = w_S^1 \cdot u + w_S^0$$

The sum of the weights is $w_U + w_L + w_S = 1$. In order to optimize the texture space near ground level, the weights w_L^1 and w_S^1 are scaled by u. This causes the weights w_L and w_S to grow from a minimum at $u = 0$ to their maximum at $u = 1$. The weight w_U is defined as the remainder of the other weights, causing it to shrink from the maximum at $u = 0$ to its minimum at $u = 1$. The weight w_S^0 defines a minimal texture parameter space for the surface region at $u = 0$.

The weighted power mappings may be converted from 3D texture coordinates as follows.

When $v \geq w_L + w_S$:

$$\cos(\phi) = \left(\frac{v - (w_L + w_S)}{w_U} \right)^{p_U}$$

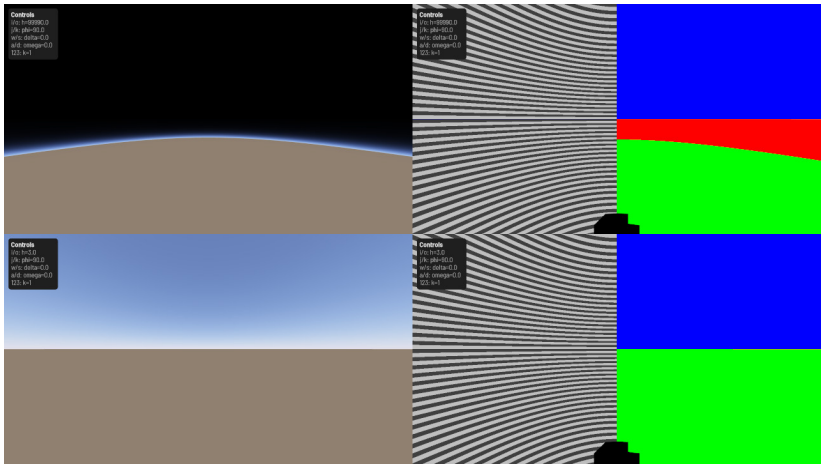
When $v \geq w_S$:

$$\cos(\phi) = -\cos(\phi_H)\left(\frac{v - w_S}{\max(w_L, \epsilon)}\right)^{p_L} + \cos(\phi_H)$$

Otherwise:

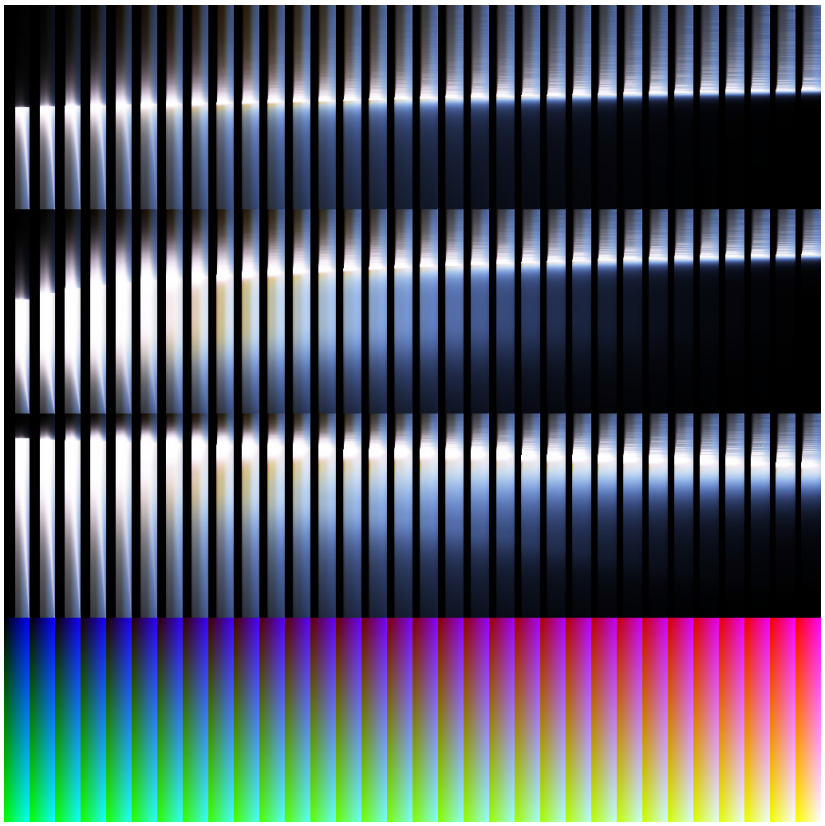
$$\cos(\phi) = (-1 - \cos(\phi_H))\left(1 - \frac{v}{w_S}\right)^{p_S} + \cos(\phi_H)$$

The following diagram shows a visualization of the weighted power mapping regions (U: blue, L: red, S: green) for $h = H_a$ and $h = 0$.



2.4.6 Mapping Comparision

The following diagram shows a comparision between the linear mapping, non-linear mapping, weighted power mapping and 3D texture coordinates of the scattering intensity I_S.



The following parameters where used for the weighted power mapping.

- $p_U, p_L, p_S = 2, 1, 2$
- $w_L^1, w_S^1, w_S^0 = \frac{20}{32}, \frac{4}{32}, \frac{4}{32}$

Note: $p_L = 1$ is important to improve numerical stability at the boundary between the upper and lower atmosphere.

2.4.7 Sun-View Azimuth Mapping

To reduce the parameter count, the Rendering Parametrizable Planetary Atmospheres with Multiple Scattering in Real-Time 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.

2.4.8 Texture Coordinates and Array Indices

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.

$$i = x + y \cdot width + z \cdot width \cdot height$$

The Efficient and Dynamic Atmospheric Scattering paper uses the following texture dimensions where optional optimizations discussed for v component.

$$(width, height, depth) = (32, 256|128|64, 32)$$

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
- Atmospheric Rendering
  - https://github.com/danielshervheim/atmosphere
- Efficient and Dynamic Atmospheric Scattering
  - https://publications.lib.chalmers.se/records/fulltext/203057/203057.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
- 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
- A Scalable and Production Ready Sky and Atmosphere Rendering Technique
  - https://sebh.github.io/publications/egsr2020.pdf
- Solar System
  - https://github.com/SebLague/Solar-System
  - https://www.youtube.com/watch?v=DxfEbulyFcY
- Trapezoidal Rule
  - https://en.wikipedia.org/wiki/Trapezoidal_rule
  - https://math.stackexchange.com/questions/2891298/derivation-of-2d-trapezoid-rule
- Spherical Coordinate System
  - https://en.wikipedia.org/wiki/Spherical_coordinate_system