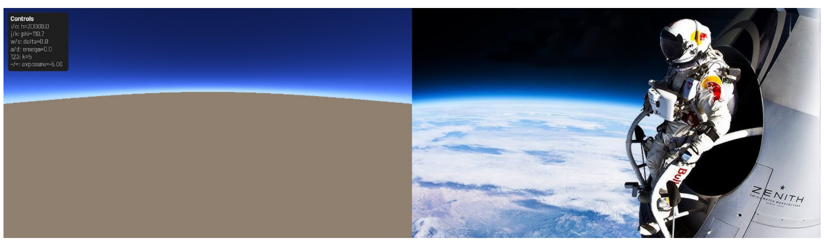
Atmosphere Plus

Jeff Boody | jeffboody@3dgesoftware.com

About

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

The following shows a side-by-side comparison between the rendered atmosphere and a picture of Felix Baumgartner (I Jumped From Space).



1 Background

1.1 Light and Color

1.1.1 Wavelength of Light

The wavelength of light is a continuous value which represents the light color.

The typical range of wavelengths for each light color include.

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

However, atmospheric simulations typically select a discrete set of wavelengths to represent the light color.

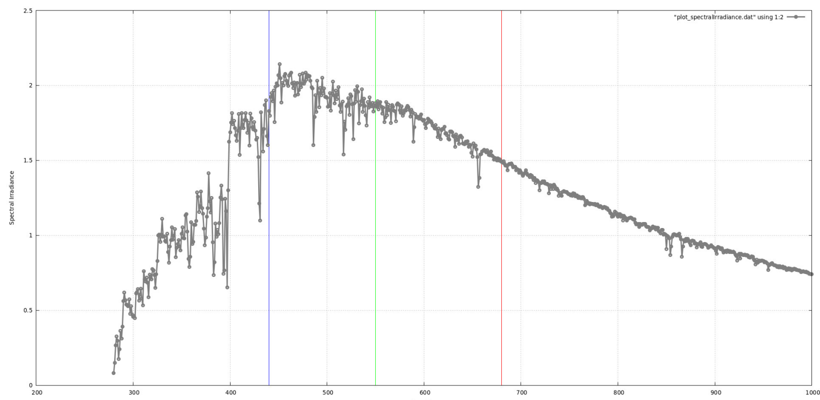
- $\lambda_r = 680$ nm

- $\lambda_g = 550$ nm
- $\lambda_b = 440$ nm

1.1.2 Spectral Irradiance

The spectral irradiance measures the power density of solar radiation at specific wavelengths at the top of atmosphere (TOA) and may be found in tables of measured values.

The following diagram shows the spectral irradiance over the visible spectrum from the ASTMG173 data set.



1.1.3 Spectral to RGB Color Conversion

To convert a spectral wavelength λ to an RGB color, we will follow a two step process. First, the wavelengths must be converted to a CIE XYZ (1931) color space by looking up the wavelength in a table. Next, the XYZ coordinates can be transformed into a color space via matrix multiplication followed by optional normalization. For our purposes we will select the sRGB color space since it has been designed to produce outputs that that are optimized for modern displays.

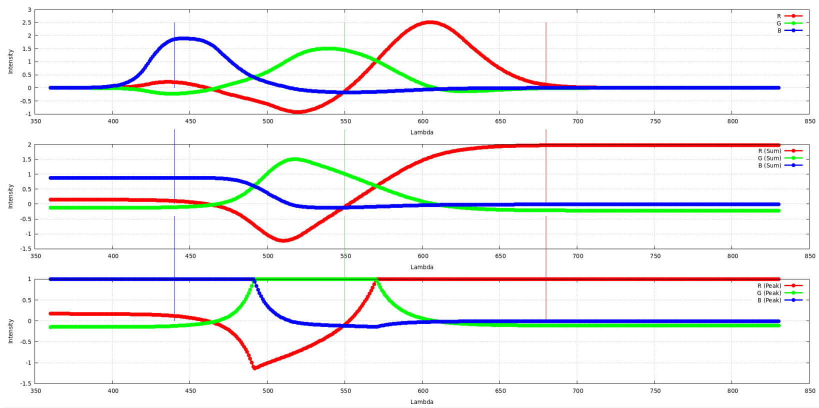
$$C_{rgb} = Normalize\left(M^{-1} \cdot XYZ\left(\lambda\right)\right)$$

Where the sRGB transformation matrix M^{-1} is defined.

$$M^{-1} = \begin{bmatrix} 3.2404542 & -1.5371385 & -0.4985314 \\ -0.9692660 & 1.8760108 & 0.0415560 \\ 0.0556434 & -0.2040259 & 1.0572252 \end{bmatrix}$$

The following plot shows the spectral to RGB conversion with various optional normalization techniques.

- None: No normalization
- Sum: Scale XYZ by the sum of XYZ values
- Peak: Scale RGB by the peak RGB value



Note: RGB colors should be clamped to $(0, 1)$ for display.

1.1.4 Spectral Intensity of Incident Light

The spectral intensity of incident light I is defined by the intensity of light at the top of atmosphere (TOA). The computation of I involves computing the integral of the spectral intensity over all wavelengths while simultaneously converting wavelengths into RGB colors (do not apply clamping or normalization when computing I). The resulting output is a high dynamic range (HDR) color.

$$I(\lambda) = M^{-1} \cdot \left(\int_{\lambda} SpectralIrradiance(\lambda) \cdot XYZ(\lambda) \cdot d\lambda \right)$$

The computed spectral intensity of incident light I.

$$I = I(\lambda_r, \lambda_g, \lambda_b) = (213.865952, 190.346115, 183.806488)$$

1.2 Exposure, Tone Mapping and Gamma Correction

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

1.2.1 Exposure

Exposure scales the overall intensity of color values and mimics the behavior of camera f-stops (e.g. one stop doubles or halves the intensity).

$$Exposure\left(C\right) = 2^{exposure} \cdot C$$

Where the default exposure is defined.

$$exposure = -2.5$$

1.2.2 Tone Mapping Operation (TMO)

Tone mapping operations transform the high dynamic range 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. The transformation from HDR-to-LDR (High Dynamic Range to Low Dynamic Range) is a lossy operation which produces subjective results. Experimental results suggest that the Hable (Uncharted 2) TMO is suitable for atmospheric simulations.

The TMO_{HP} defines a partial operation that is used to compute TMO_{H}.

$$A, B, C, D, E, F = (0.15, 0.50, 0.10, 0.20, 0.02, 0.30)$$

$$TMO_{HP}\left(X\right) = \left(\frac{X \cdot (A \cdot X + C \cdot B) + D \cdot E}{X \cdot (A \cdot X + B) + D \cdot F}\right) - \frac{E}{F}$$

The TMO_{H} operator is referred to as a filmic operation since it is designed to emulate real film.

$$ExposureBias = 2.0$$

$$W = vec3\left(11.2\right)$$

$$TMO_{H}\left(X\right) = \frac{TMO_{HP}\left(X \cdot ExposureBias\right)}{TMO_{HP}\left(W\right)}$$

1.2.3 Gamma Correction

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

$$Gamma(C) = C^{1.0/2.2}$$

1.2.4 HDR-to-LDR Transformation

The HDR-to-LDR conversion may include exposure, tone mapping and gamma correction.

$$C_{LDR} = Gamma\left(TMO\left(Exposure\left(C_{HDR}\right)\right)\right)$$

1.3 Atmospheric Boundary

The atmosphere consists of 5 layers where the height of each layer is defined as follows.

- Troposphere
  - 0 km to 20 km at the equator
  - 0 km to 9 km at 50 degree latitude
  - 0 km to 6 km at the poles
- Stratosphere
  - Troposphere to 50 km
- Mesophere
  - 50 km to 85 km
- Thermosphere
  - 85 km to 600 km
- Exosphere
  - 600 km to 10,000 km

However, atmospheric simulations commonly model the atmosphere as the region where it's effects are non-negligible. This region extends from the Earth's surface to an atmospheric boundary referred to as the "top of atmosphere" (TOA). The Scalable and Production Ready Sky and Atmosphere Rendering Technique paper uses the following constants for these radii.

- Radius of the planet: $R_p = 6360$ km
- Radius of the atmospheric boundary: $R_a = 6460$ km
- TOA: $TOA = R_a - R_p = 100$ km

1.4 Atmospheric Particles

The atmospheric simulation models three types of particles.

- Rayleigh Scattering: Smaller particles consisting of air molecules such as oxygen, nitrogen and carbon dioxide.
- Mie Scattering and Absorption: Larger particles consisting of ice crystals, water droplets, dust and air polution.
- Ozone Absorption: Ozone particles consist of O_3.

1.5 Density Function

1.5.1 Definition of the 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, $p_{R,M}(h)$:

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

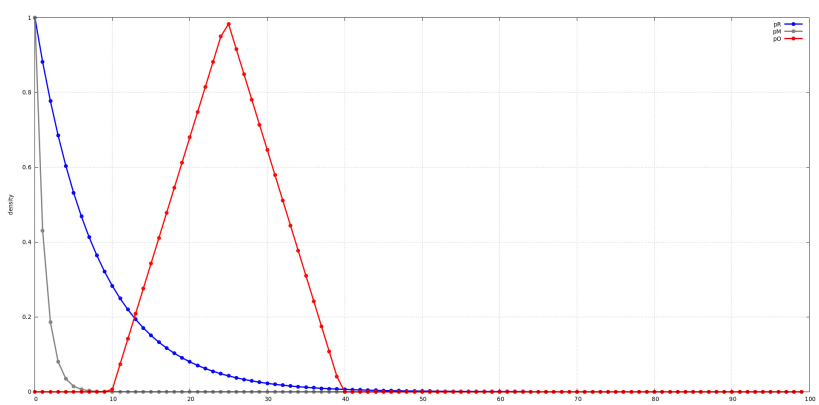
Where $H_R \approx 8000$ m and $H_M \approx 1200$ m are the Rayleigh/Mie scale heights (e.g. the altitude where the density of particles scales down by a $\frac{1}{e}$ term).

Ozone density function, $p_O(h)$:

$$p_O(h) = \max\left(0, 1 - \frac{|h - 25000|}{15000}\right)$$

1.5.2 Visualization of the Density Function

The following plot show the Rayleigh, Mie and Ozone density functions.



1.6 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.

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

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 a default value of $g = 0.8$.

1.7 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.

$$\beta_R^s = \beta_R^s\left(\lambda_r, \lambda_g, \lambda_b\right) = \left(5.802e - 6, 13.558e - 6, 33.1e - 6\right)$$

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

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

$$\beta_O^a = \beta_O^a\left(\lambda_r, \lambda_g, \lambda_b\right) = \left(0.65e - 6, 1.881e - 6, 0.085e - 6\right)$$

$$\beta^e_{R,M,O} = \beta^s_{R,M} + \beta^a_{M,O}$$

Note: The Rayleigh/ozone coefficients are 3-component vectors (color / wavelength dependent) while the Mie coefficients is a scalar (grayscale / wavelength independent).

1.8 Transmittance

1.8.1 Definition of Transmittance

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

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

$$t\left(P_1, P_2\right) = \sum_{i \in R,M,O} \beta^e_i \int_{P_1}^{P_2} p_i\left(h\left(P\right)\right) ds$$

Where the parameters are defined:

- T: Transmittance, $T \in (0, 1)$
- t: Optical depth, $t \in (0, \infty)$
- P: Sample point parameterized by s, $P = P_1 + s\left(P_2 - P_1\right)$
- h: Height is parameterized by P, $h\left(P\right) = |P| - R_p$

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 (e.g. the atmosphere). 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.

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

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

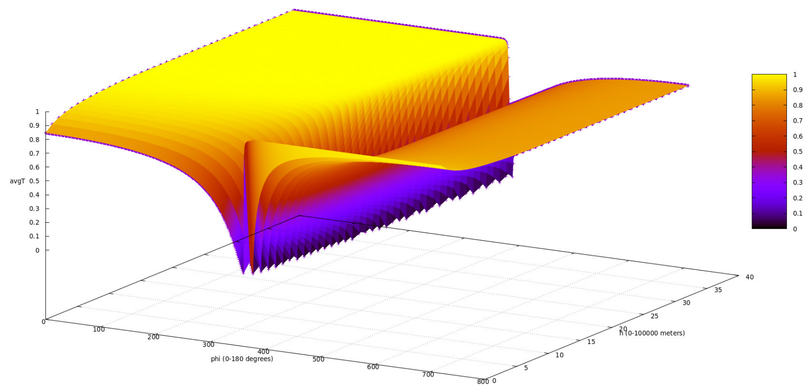
$$a^m \cdot a^n = a^{m+n}$$

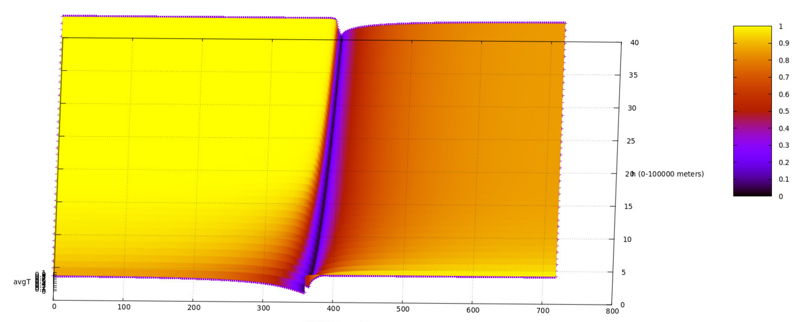
Where the functions $t(\ldots)$ may be rearranged in order to group similar terms.

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

1.8.2 Visualization of Transmittance

The following diagrams show the average transmittance as a function of altitude h and the View-Zenith Angle ϕ. Notice how the valley follows the horizon as the height increases. The transmittance is much lower at the horizon since the atmosphere is much thicker in the horizontal direction than the vertical direction. See the Scattering Intensity Parameterization section for more details.





2 Atmospheric Simulation Passes

The atmospheric simulation requires solving a series of complex equations for which there does not exist a closed form solution. These equations model different interactions of light with atmospheric particles and the planet surface. An iterative procedure will be applied to compute the contribution of each individual pass over multiple scattering orders.

2.1 Single-scattering Pass

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

$$S^{(1)}(P_0, V, L) = \sum_{i \in R,M} I \cdot F_i(dot(L, -V)) \cdot \frac{\beta_i^s}{4\pi} \cdot$$

$$\cdot \int_{P_a}^{P_b} p_i(h(P)) \cdot T(P_c, P) \cdot T(P, P_a)\, ds$$

Where the following parameters are defined:

- L: Direction of light from the Sun
- 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: Sample point parameterized by s, $P = P_a + s(P_b - P_a)$
- h: Height is parameterized by P, $h(P) = |P| - R_p$
- θ: Scattering angle, $\cos(\theta) = dot(L, -V)$
- P_c: The nearest intersection point along the ray from P in the direction of the Sun (e.g. the atmosphere boundary).
- ds: Step delta for the sample points P between $P_a$and P_b.

The single-scattering contribution is shadowed (zero) if the ray from P to P_c intersects the planet.

2.2 Diffuse Lighting Pass

The diffuse lighting equation describes the direct illumination color contribution to the planet surface by the incident light that was attenuated by the atmosphere.

$$D(P_p, L) = C \cdot dot(N, L) \cdot I \cdot T(P_p, P_c)$$

Where the following parameters are defined:

- C: The diffuse color of the surface
- P_p: A point on the planet surface
- P_c: The nearest intersection point along the ray from P_p in the direction of the Sun (e.g. the atmosphere boundary).
- N: The surface normal at P_p, $N = normalize\,(P_p)$

The diffuse contribution is shadowed (zero) if the ray from P_p to P_c intersects the planet.

When rendering the planet surface, the diffuse lighting should incorporate the actual terrain (P_p, N and shadow occlusion).

2.3 Ambient Lighting Passes

The ambient lighting equation describes the indirect illumination color contribution to the planet surface by the light scattered through the atmosphere.

$$A^{(1)}\,(P_p, L) = 0$$

$$A^{(k>1)}\,(P_p, L) = C \cdot \int_{2\cdot\pi} dot\,(N, \omega) \cdot S^{(k-1)}\,(P_p, \omega, L)\, d\omega$$

Where the following parameters are defined:

- C: The diffuse color of the surface
- N: The surface normal at P_p, $N = normalize\,(P_p)$

The total intensity of the ambient light is the sum over all orders.

$$A = \sum_{k=1}^{K} A^{(k)}$$

When rendering the planet surface, the ambient lighting may incorporate an ambient occlusion map to partially account for the actual terrain.

2.4 Multiple-scattering Passes

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

$$S^{(k)}\left(P_0, V, L\right) = \sum_{i \in R,M} \frac{\beta_i^s}{4\pi} \cdot$$

$$\cdot \int_{P_a}^{P_b} G_i^{(k-1)}\left(P, V, L\right) \cdot p_i\left(h\left(P\right)\right) \cdot T\left(P, P_a\right) ds$$

Where the gather-scattering equation $G_i^{(k)}$ 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. This variation of $G_i^{(k)}$ includes contributions from diffuse and ambient lighting passes.

$$G_i^{(k)}\left(P, V, L\right) = \int_{4\pi} F_i\left(dot\left(-\omega, -V\right)\right) \cdot H^{(k)}\left(P, \omega, L\right) d\omega$$

Where the following parameters are defined:

- ω: Gathering direction for the kth order scattered light source
- θ: Scattering angle, $\cos\left(\theta\right) = dot\left(-\omega, -V\right)$
- P_p: Intersection of the gathering direction ω with the planet surface
- $H^{(1)}\left(P, \omega, L\right) = S^{(1)}\left(P, \omega, L\right)$
- $H^{(2)}\left(P, \omega, L\right) = S^{(2)}\left(P, \omega, L\right) + \frac{1}{\pi} \cdot D\left(P_p, L\right) \cdot T\left(P_p, P\right)$
- $H^{(k>2)}\left(P, \omega, L\right) = S^{(k)}\left(P, \omega, L\right) + \frac{1}{\pi} \cdot A^{(k)}\left(P_p, L\right) \cdot T\left(P_p, P\right)$

The diffuse and ambient contributions are zero if ω does not intersect the planet.

In addition, the diffuse and ambient terms require a $\frac{1}{\pi}$ Lambertian normalization factor to conserve energy. The coefficient is a result of the BRDF (Bidirectional Reflectance Distribution Function) which describes how light reflects off a surface from different directions.

$$\int_{\Omega} f_r \cdot \cos\left(\theta\right) d\omega = f_r \cdot \int_0^{2\cdot\pi} \int_0^{\frac{\pi}{2}} \cos\left(\theta\right) \cdot \sin\left(\theta\right) \cdot d\theta \cdot d\phi = f_r \cdot \pi = 1$$

The total intensity of the multiple scattered light is the sum over all orders.

$$S = \sum_{k=1}^{K} S^{(k)}$$

3 Texture Parameterization

3.1 Scattering Texture Parameterization

3.1.1 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.

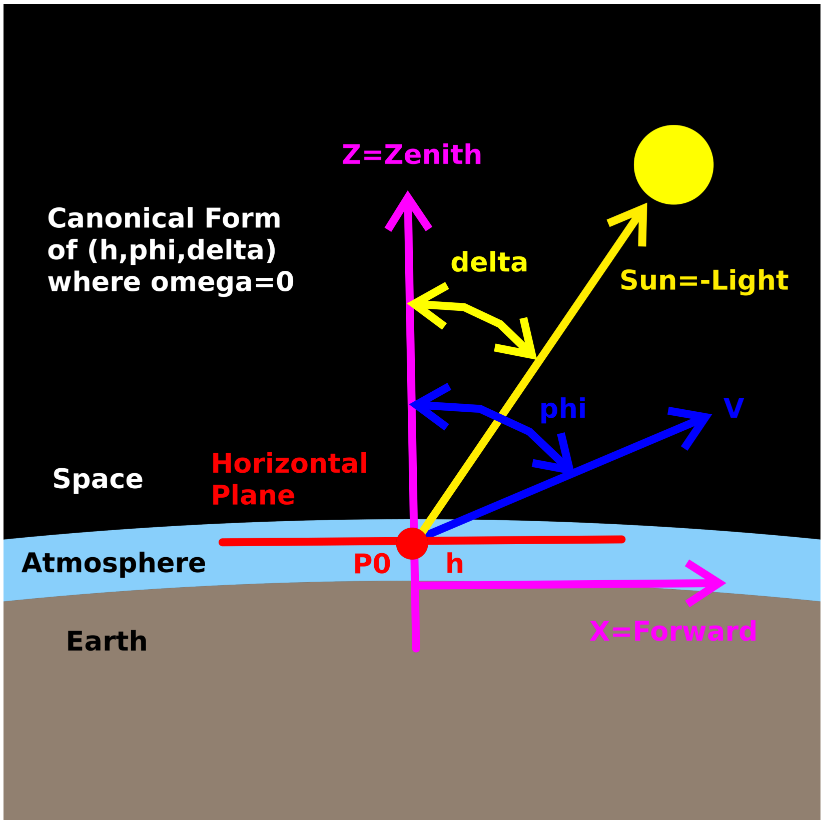
The Zenith is the up vector for the point P_0.

$$Zenith = normalize(P_0)$$

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. Observe that the atmosphere is much thinner in the vertical direction than the horizontal direction.



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 $\Phi = \omega = 0$.

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

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

$$L = -Sun$$

3.1.2 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.

$$h = u \cdot H_a$$

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

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

3.1.3 Power 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 linear and nonlinear mappings may be generalized for an arbitrary power as follows.

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

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

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

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

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

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

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

3.1.4 Weighted Power Mapping

The View-Zenith Angle is especially difficult to generalize due to the following conditions which lead to inefficient parameterization.

- The horizon ϕ_H increases from 90 degrees at ground level towards 180 degrees as the height increases. However, the published approaches typically fix the horizon ϕ_H at 90 degrees for all heights which causes incorrect angles to be compressed as the height increases.
- At ground level there is no atmosphere when looking downwards, whereas at the TOA there is no atmosphere when looking upwards. A large percentage of the texture space is dedicated to parameters which represent little or no atmosphere.

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 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 $v \geq w_S$:

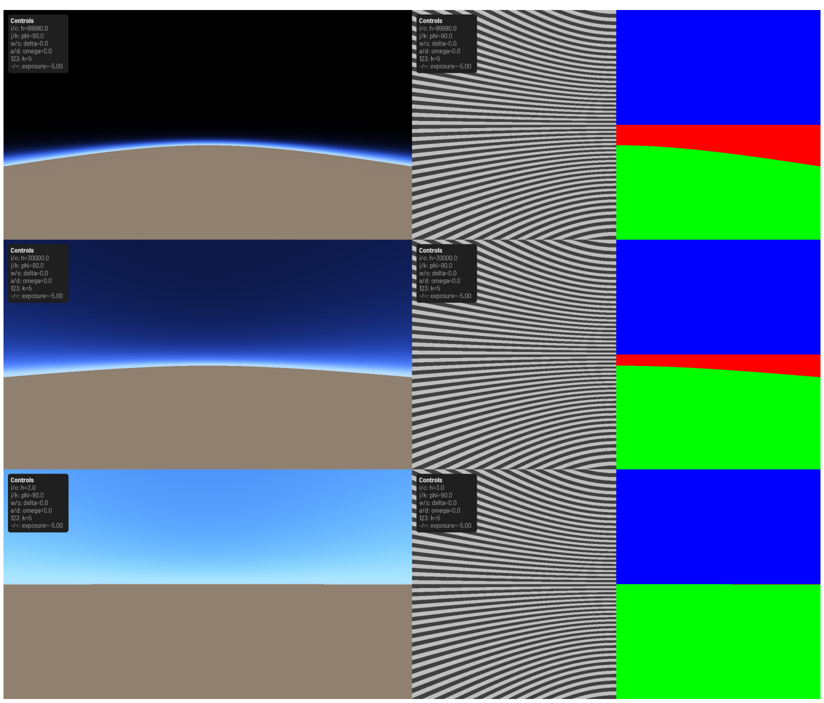
$$\cos(\phi) = -\cos(\phi_H)\left(\frac{v - w_S}{\max(w_L, \epsilon)}\right)^{p_L} + \cos(\phi_H)$$

Otherwise:

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

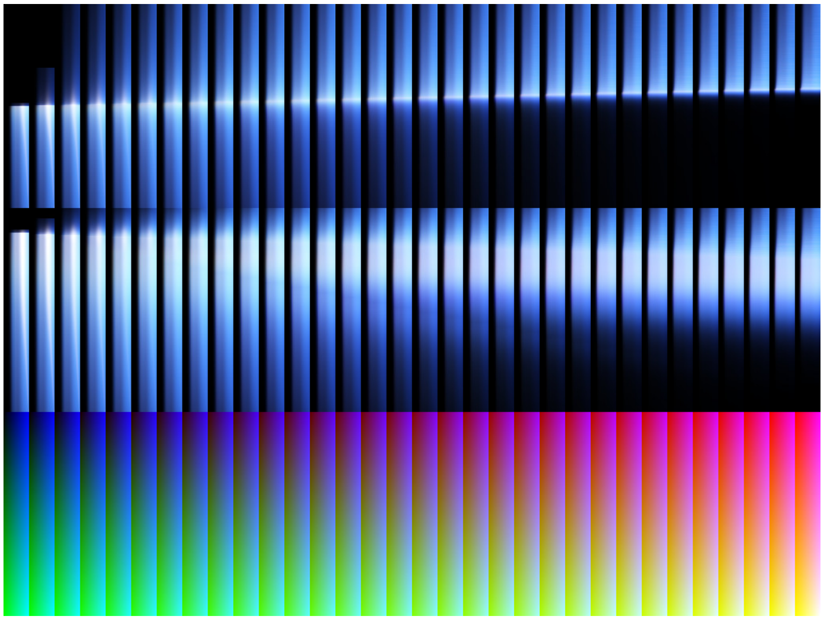
3.1.5 Weighted Power Mapping Visualization

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



3.1.6 Linear vs Weighted Power Mapping Comparision

The following diagram shows a comparision between the linear mapping and weighted power mapping of the scattering intensity I_S.



The following parameters where used for the weighted power mapping.

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

3.1.7 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.

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

4 Numerical Integration

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

4.1 1D Trapezoidal Rule

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 $x_n = b$.

4.2 2D Trapezoidal Rule

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

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

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

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

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

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 $y_m = d$.

4.3 Spherical Integration

Spherical integration is the process of evaluating integrals over three-dimensional regions by converting to spherical coordinates.

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

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

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

$$d\omega = \sin\left(\Theta\right) d\Theta \cdot d\Phi$$

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

- Tone Mapping
  - https://64.github.io/tonemapping/

- Filmic Tonemapping Operators
  - http://filmicworlds.com/blog/filmic-tonemapping-operators/

- Solar Spectra
  - https://www.nrel.gov/grid/solar-resource/spectra

- Color stuff with spectra
  - https://gist.github.com/cgobat/9d7f8957523f0ab925043231d431562f

- Some Common RGB Working Space Matrices

  - http://www.brucelindbloom.com/index.html?Eqn_RGB_XYZ_Matrix.html
- Spectral Intensity of Incident Light Questions with Perplexity
  - II-perplexity.pdf
- Layers of the Atmosphere
  - https://www.noaa.gov/jetstream/atmosphere/layers-of-atmosphere
- Solar System
  - https://github.com/SebLague/Solar-System
  - https://www.youtube.com/watch?v=DxfEbulyFcY
- I Jumped From Space (World Record Supersonic Freefall)
  - https://www.youtube.com/watch?v=Hz2F_S3Tl0Y
- 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