

# Efficient Atmospheric Rendering For Mobile Virtual Globes

José M. Santana  
University of Las  
Palmas de G.C.  
Imaging Technology  
Center (CTIM)  
Spain (35017), Las  
Palmas de G.C.  
josemiguel.santana  
@ulpgc.es

Agustín Trujillo  
University of Las  
Palmas de G.C.  
Imaging Technology  
Center (CTIM)  
Spain (35017), Las  
Palmas de G.C.  
agustin.trujillo  
@ulpgc.es

José P. Suárez  
University of Las  
Palmas de G.C.  
Division of  
Mathematics,  
Graphics and  
Computation  
(MAGiC), IUMA,  
Spain (35017), Las  
Palmas de G.C.  
josepablo.suarez  
@ulpgc.es

Sebastián Ortega  
University of Las  
Palmas de G.C.  
Division of  
Mathematics,  
Graphics and  
Computation  
(MAGiC), IUMA,  
Spain (35017), Las  
Palmas de G.C.  
sebastian.ortegatrujillo  
@gmail.com

## ABSTRACT

Most virtual globe systems feature a rendering of the atmosphere that surrounds the earth. This element is so widespread that seeing a virtual earth without a surrounding halo makes the image seem much more artificial. Atmospheric rendering is a costly process aimed to enhance the realism and beauty of the scene. For that reason many virtual globe systems, specially in low-resourced devices, rely on simplified schemes to represent the atmosphere. However, the accurate representation of the atmosphere implies the volumetric rendering of the semi-transparent air mass that covers the whole geographical scene. Thus, the color of each pixel is composed of the light scattered by all the points in space projected on that pixel. The present work takes advantage of the spherical symmetry of the atmosphere's mathematical model to implement efficiently this volumetric rendering on mobile devices.

## Keywords

Virtual Globe, Atmospheric Rendering, Volumetric Rendering, Rayleigh Scattering, Mobile Devices

## 1 INTRODUCTION

Volumetric rendering on mobile devices is a challenging field [9] and there are many proposals for general purposed volume visualizations, most of them based on 2D/3D textures that describe the composition of the gas. Fortunately, the features of the atmospheric model allows us to tackle it in an analytical manner. The contribution presented in this work consists of an implementational study of atmospheric *Rayleigh Scattering* that uses state of the art techniques. The atmosphere presents other visual effects like *Mie Scattering*, which represents the scattering due to particles of a comparable size to the wavelengths of visible light, for instance water droplets in clouds. However, these other effects are out of the scope of this research, and some of them use techniques very similar to the ones here presented. Hence, the final goal is to obtain a realistic rendering of the *Rayleigh scattering* effect using the tools and hardware limitations of an OpenGL ES 2.0 supported mobile device. To that effect, the 3D problem is translated to several 2D domains in which the problem is computationally reduced.

## 2 ATMOSPHERIC MODEL

The atmosphere of the earth can be thought of as a spherically-symmetric envelope of pure air. Consider-

ing also a spherical model of the earth, centred at the axis origin, it extends from the planetary surface (at a radius  $R_E$  of  $6,371 \times 10^9$  m.) to an undefined height, decreasing in density as we move upwards.

The density distribution of such a mass of gas can be fairly represented in terms of an exponential formula,  $D(h) = D_{max} * exp(-h/H_0)$  where  $p(h)$  is a density factor and  $H_0$  can be assigned 7994, following the work of Nishita and Sirai [7].

However, in this work we are only considering the illumination contributions of the troposphere and the stratosphere, which reaches a height  $H_{Max}$  of 50 km. Color contributions of upper layers are negligible and can be dismissed.

The real atmosphere thickness is around  $8,3 \times 10^{-4}$  % of the radius of the earth. Hence, our final implementation includes a scale factor for visualization purposes, as other systems apply to the rendering of terrain orography. The radius of the final atmosphere,  $R_A$  from now on, is of  $50 \times 10^3 \times S + R_E$  m., being  $S$  the scale factor. Likewise, the density factor must be rescaled as  $H'_0 = H_0 \times S$ .

On our model the sun is placed in the position  $P_S$  at a distance  $149.6 \times 10^9$  m. from the center of the scene on the X axis. All incoming light can be considered then

parallel, pointing on the X positive direction. Hence, a cylindrical shadow is projected in the same direction which is taken into account on our calculations.

As previously mentioned, the two most notable forms of atmospheric scattering are Rayleigh scattering and Mie scattering. Rayleigh scattering is caused by small molecules in the air, and it scatters light more heavily at shorter wavelengths (blue first, then green, and then red). In our rendering model, the atmosphere is composed by an isotropic pure air, so the only modeled interaction is Rayleigh scattering, which is responsible for most atmospheric effects, the predominant blue tone of the daytime sky, as well as the reddish dawns and sunsets.

There have been several attempts to accurately render the atmosphere in VG systems [8, 6]. The work of [7] has been specially relevant due to its clear understanding of the underlying numerical analysis of the light spectrum. A bright example of a GPU implementation is given by Sean O'Neil in the book [10]. Other papers have addressed more advanced effects, such as smoke and cloud rendering [4], the light reflected by different parts of the earth or mountain shadows (light shafts) [1].

### 3 THE ATMOSPHERIC RAYLEIGH SCATTERING MODEL

The model considers the refracted light in the direction of the observer given the illumination of the view ray by a directional light source (the Sun). Based on the Rayleigh scattering equation, the amount of sunlight of a particular wavelength refracted by a particular point in the direction of the camera is given by the equation:

$$F(\theta, g) = \frac{3(1-g^2)}{2(2+g^2)} \times \frac{(1+\cos^2(\theta))}{(1+g^2-2g \times \cos(\theta))^{3/2}} \quad (1)$$

$$I(\lambda, \theta, h) = \frac{I_0(\lambda)KFr(\theta)}{\lambda^4} * \rho(h) \quad (2)$$

This equation is based on the following terms:

- $I_0(\lambda)$  is the incoming intensity of sunlight for a given wavelength  $\lambda$  (in nanometers).
- $Fr(\theta)$  is the phase function, in this case implemented as the Henyey-Greenstein function, with a  $g = 0$ .
- $K$  is a constant based on the molecular density of the atmosphere (see [7]).
- $\rho(h)$  is the density coefficient explained in Section 2.

To compute the final light incident on our camera, it is also necessary to take into account the amount of light scattered out by upper layers of the atmosphere before

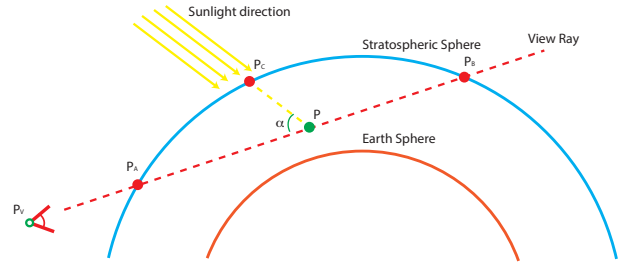


Figure 1: Geometrical model of a view ray through the atmosphere.

reaching the ray  $P_aP_b$ . Besides, the light is scattered out in its way to the observer by the air in the ray  $P_aP_b$  itself.

Both terms are based on the *Optical Depth* of the rays  $P_cP$  and  $PP_V$  respectively.  $P$  represents each of the points of the main ray,  $P_C$  is the point where the sunlight hitting  $P$  enters the atmosphere and  $P_V$  is the camera position. The physical system is depicted in Figure 1: By applying the *Beer's Exponential Extinction Law*, we can determine that light will be scattered out exponentially to those optical depths, resulting in:

$$t(P_aP_b, \lambda) = \frac{4\pi K}{\lambda^4} \int_{S(P_a)}^{S(P_b)} \rho(h(P(s))) ds \quad (3)$$

$$W(\lambda, \theta) = I_0(\lambda)Fr(\theta) \frac{K}{\lambda^4} \quad (4)$$

$$O(P_a, P, P_c) = \exp(-t(P_aP, \lambda) - t(P_cP, \lambda)) \quad (5)$$

$$I(\lambda) = W(\lambda, \theta) \int_{S(P_a)}^{S(P_b)} \rho(h(P(s)) \times O(P_a, P(s), P_c(s)) ds \quad (6)$$

### 4 FULL-SPECTRAL RENDERING

The light emitted by the sun can be fairly modeled as the radiation of a black-body, which is an opaque object at thermodynamic equilibrium. This radiation is classically described by Planck's equation. Considering the standardized color space CIE1931, the visible spectrum of light ranges from 380 nm. to 780 nm. of wavelength.

To convert these wavelengths to specific colors in the XYZ space of CIE 1931, the *Simple, Single-Lobe Fit* model has been used. Assuming a sRGB display, an approximation of these values can be achieved by a linear transformation. The tristimulus components RGB of such colors are precalculated, so they do not need to be computed on each frame.

### 5 EXPLOITING MODEL SYMMETRY

Due to the spherical symmetry of the atmospheric density distribution, we can define a coordinate system so ray  $\vec{AB}$  is in the XY plane and parallel to Y.

$$M = \begin{bmatrix} \hat{D} \\ \hat{C} \\ \hat{C} \times \hat{D} \end{bmatrix} \text{ where } \begin{matrix} D = (B - A) \\ C = A + (-A \cdot D)D \end{matrix} \quad (7)$$

The starting and ending points of the ray can therefore be computed as  $P_A = M \times A'$  and  $P_B = M \times B'$ . The points in this 2D space can be expressed in polar coordinates, being  $h = \sqrt{x^2 + Y_0^2}$ , and  $\varphi = \text{atan2}(Y_0, x)$ . Note that  $Y_0^2$  can be precomputed for all points along the ray. This new parametrization of a ray traversing the atmosphere can be seen in Figure 2.

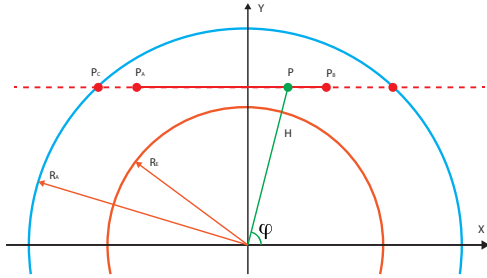


Figure 2: Ray system simplified to 2D equivalent.

## 6 PRECALCULATING OUT-SCATTERING OPTICAL DEPTHS

The current proposal offers some computational shortcuts to obtain the out-scattering coefficients needed to compute the full atmospheric model. The main idea was first proposed in the work of Nishita and Sirai [7] and to a further extent by Woolley [10]. However, they do not explicitly apply the transformation explained in Section 5 from which these caching techniques benefit, nor explore the limits of our integrating function due to inherent symmetries.

The out-scattering factors require the integration of the rays  $\vec{P_C \hat{P}}$  (coming from the sun) and  $\vec{P_A \hat{P}}$  (coming from the observer). By translating these two rays to our 2D plane, these two integrals can be expressed in the form:

$$f(X_0, X_1, Y_0) = \int_{X_0}^{X_1} \rho(h(x, Y_0)) dx \quad (8)$$

$$f(X_0, X_1, Y_0) = g(X_1, Y_0) - g(X_0, Y_0) \quad (9)$$

where  $g(X, Y) = \int_{-R_A}^{X_1} \rho(h(x, Y_0)) dx$ .

This function can be transformed to polar coordinates as  $g(\varphi, h)$  offering a much more compact, rectangular definition of its domain, as seen in Figure 3. This table can be further compressed, taking into account the following rotational symmetries:

- $g(\varphi, h) = g(-\varphi, h)$ .
- For  $h > R_A$ ,  $g(\varphi, h) = 0$ .
- For  $h < R_E$ ,  $g(\varphi, h) = \infty$ .
- For  $0 \leq \varphi \leq \pi/2$ ,  $g(\varphi, h) = g(\pi/2, h * \sin(\varphi)) + (g(\pi/2, h * \sin(\varphi)) - g(\pi - \varphi, h)) = 2 * g(\pi/2, h * \sin(\varphi)) - g(\pi - \varphi, h)$ .

Thus,  $g(\varphi, h)$  is fully represented by mapping the domain  $\varphi = \pi/2 - \pi$  and  $h = R_E - R_A$ . As a counterpart, a reading of the optical depth function,  $f(X_0, X_1, Y_0)$  could require two accesses to the table, or used texture. It is also important to note that, due to the polar nature of the function  $g(\varphi, h)$  we lose spatial resolution as we move upwards, as it is pointed out by [1]. Ensuring a minimum spatial resolution of  $n$  meters requires an angular resolution of  $\Delta\alpha = \arcsin(\frac{n}{E_A})$ .

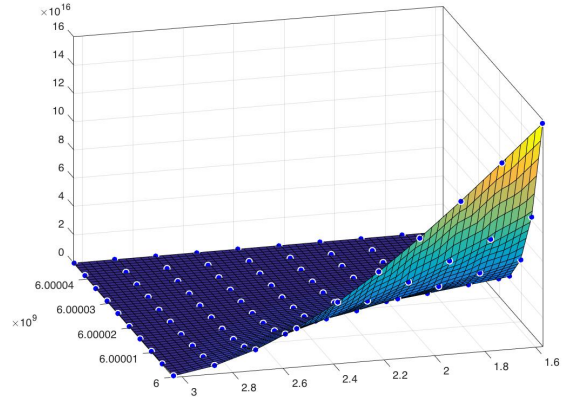


Figure 3: Optical depth map in polar coordinates.

## 7 COMPUTING INTERPOLATION POINTS

As a contribution, the present work proposes the use of a precomputed array of an odd length ( $n$ ) of the form  $V_i = 1 - \exp(-\frac{h}{n} \times i)$ . For rays starting outside of the atmosphere, the highest points are separated from the outer atmosphere a ratio of  $\exp(-\mu)$ , thus a value of  $\mu = 2.5$  produces a separation of 0.08 which seems reasonable. Using the same formula, two other vectors  $V'$  and  $V''$ , of lengths  $\lfloor \frac{n}{2} \rfloor$  and  $\lfloor \frac{n}{2} \rfloor$  respectively, are precalculated. Once the ray has been transformed to the  $X_0, X_1, Y_0$ , from it is determined whether it crosses the  $Y$  axis ( $X_1 > 0$ ) or not.

For rays not crossing that line, intersection points  $P_i$  are calculated by Expression 10, and rays that do by Expression 11.

$$P_i = \{X_0 + V_i * (X_1 - X_0), Y_0\} \quad (10)$$

$$P_i = \begin{cases} \{X_0 - V'_i * X_0, Y_0\} & \text{if } i < n/2 \\ \{X_0 + V''_i * X_1, Y_0\} & \text{if } i > n/2 \end{cases} \quad (11)$$

## 8 NUMERICAL ANALYSIS OF THE OPTICAL DEPTH MAP

The shading algorithm has been fully implemented in a Matlab program that allows to render the atmosphere from an outside perspective. This allowed us to perform numerical analysis on the intermediate results of

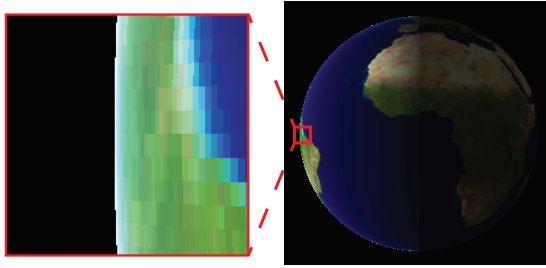


Figure 4: Final atmospheric rendering (full frame and close-up) at real scale.

the process as numerical images. Figure 4 presents a test image for not scaled atmosphere.

### 8.1 Simplification of the optical depth map

The out-scattering intensity 2D map proposed in Section 6 avoids two optical depth integrations of the final scattering model. The most straight-forward way to access such map for the shading program is to have the map as a preallocated 2D texture. However, this map is accessed up to four times for rays that cross the atmosphere, implying 4 accesses to GPU memory. Besides the read texture must be interpolated which internally derives in further memory access. In order to speed this process up, a numerical fit of the  $g(\varphi, h)$  grid data is proposed. However, after several attempts, it seems difficult to adjust properly a single 3D function to such data. Instead, a multi-curve adjustment at linearly spaced heights is proposed. Our best approximation uses 10 different equations achieving a mean error of 2.84%. All these equations can be coded into the GLSL program, so any value  $g(\varphi, h)$  can be interpolated from two polynomial evaluations.

### 8.2 Effects of view ray sampling

The light scattered in our direction by each ray is computed as a function of the amount of air it traverses and the amount of incoming lights. These punctual contributions have to be integrated along the ray at a number of locations that is implementation-specific. As an approach to reduce the computational costs of the rendering, we intend to minimize the number of sampling points used to estimate the contribution of a particular view ray. To that end, a numerical study of the different wavelength intensities has been conducted.

As depicted in Figure 5, the relative error achieved in the final image decreases as we measure the contributions of more intermediate points. However, the addition of more sampling points seems to reduce the error following a logarithmic curve. Thus, according to experimentation reducing from 13 to 5 sampling points only introduces a 16,15% of error, while improving the performance by a 2,6 speed-up factor.

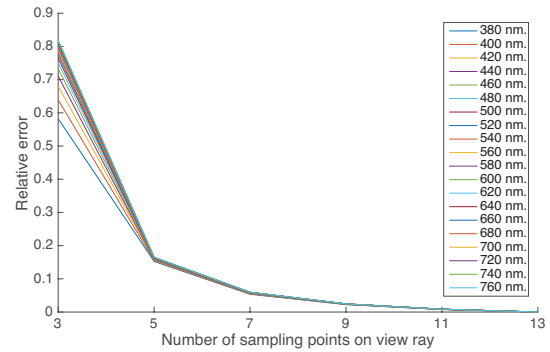


Figure 5: Wavelength intensity error vs. number of sampling points.

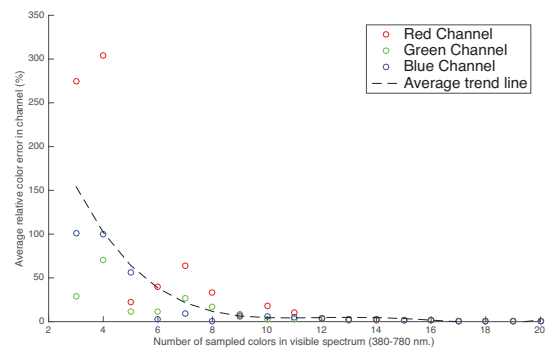


Figure 6: Color relative error due to spectrum sampling.

### 8.3 Effects of visible light spectrum sampling

As there is an uncountable infinity of possible wavelengths of light on the sun radiation, the spectrum must be sampled on several points and their color contributions integrated. This final experiment intends to show the effects of the sampling density of the visible spectrum of light wavelengths. As previous examples, we find the approximation of [5] that uses 14 integration points. The goal is to minimize the number of sampled wavelengths to produce an accurate rendering. To that end, the visible spectrum has been split in intervals of similar size that are represented by their central value. The number of samples in this experiment ranges from 3 to 20, being the "ground truth" the color achieved by this last partitioning.

This numerical study shows that the error tends to stabilize rapidly, as can be seen in Figure 6. The final sampling is discretionary and case-dependant but 10 samples offer a 9.81% error and hence seems to be a reasonable choice for many applications.

## 9 CONCLUSIONS AND FUTURE WORK

The here-presented ongoing research proposes some strategies aimed to reduce the number of operations

needed to render atmospheric effects on virtual globe environments.

To that end, the mathematics behind the Rayleigh scattering and the model of the atmosphere's gas distribution are analyzed. From this analysis, a simplified 2D representation of the involved light rays has been achieved. Thus, the overall number of needed calculations is reduced. Moreover, a compact optical depth map is generated.

This 2D simplification of the rendering algorithm has been successfully implemented on a simplified version of the atmospheric model. This first approximation disregards the out-scattering factors but at this point offers a semi-realistic look, running at 60 FPS on several tested iOS and Android devices.

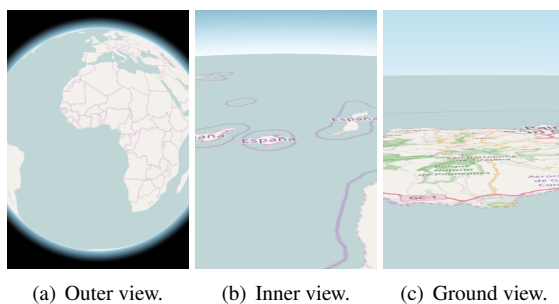


Figure 7: Atmospheric rendering. Atmosphere height exaggerated x15. This implementation was carried out using the Glob3 Mobile engine [11].

Finally, it has been experimentally demonstrated the effects of reducing the sampling rate both geometrically on the projected rays and on the light spectrum. The conducted numerical simulations suggest that 10 sampled wavelengths and 5 intermediate points per ray may be sufficient.

As future work, we intend to implement the full model on both desktop and hand-held device in order to confirm its applicability on different GPU architectures. On that basis, new factors could be added to the model, such as a Mie Scattering component, in order to enhance the realism of the atmospheric rendering.

## 10 ACKNOWLEDGMENTS

The first author wants to thank Agencia Canaria de Investigación, Innovación y Sociedad de la Información, and the European Social Fund, for the grant “Formación del Personal Investigador-2012” that made possible this work.

The fourth author wants to thank Universidad de Las Palmas de Gran Canaria for its grant “Programa de personal investigador predoctoral en formación 2015”, which made possible this work.

## 11 REFERENCES

- [1] Eric Bruneton and Fabrice Neyret. Precomputed atmospheric scattering. *Computer Graphics Forum*, 27(4):1079–1086, 2008.
- [2] Richard L. Burden and John Douglas Faires. *Numerical Analysis*. 2011.
- [3] Yoshinori Dobashi. Interactive Rendering of Atmospheric Scattering Effects Using Graphics Hardware. *Hwvs*, pages 99–109, 2002.
- [4] Yoshinori Dobashi, Kazufumi Kaneda, Hideo Yamashita, Tsuyoshi Okita, and Tomoyuki Nishita. A simple, efficient method for realistic animation of clouds. *Siggraph*, pages 19–28, 2000.
- [5] John Irwin. Full-Spectral Rendering of the Earth's Atmosphere using a Physical Model of Rayleigh Scattering. *Eurographics*, 1996.
- [6] Henrik Wann Jensen, Frédo Durand, Julie Dorsey, Michael M Stark, Peter Shirley, and Simon Premoze. A physically-based night sky model. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 399–408. ACM, 2001.
- [7] Tomoyuki Nishita and Takao Sirai. Display of The Earth Taking into Account Atmospheric Scattering. *Siggraph*, (2):175–182, 1993.
- [8] Tomoyuki Nishita, Yasuhiro Miyawaki, and Ei-hachiro Nakamae. A shading model for atmospheric scattering considering luminous intensity distribution of light sources. *ACM SIGGRAPH Computer Graphics*, 21(4):303–310, 1987.
- [9] Jose M Noguera and J Roberto Jimenez. Mobile Volume Rendering: Past, Present and Future. *Ieee Transactions on Visualization and Computer Graphics*, 22(2):1164–1178, 2016.
- [10] Cliff (University of Virginia) Woolley. *GPU Gems 2*, volume 2. 2005.
- [11] José Miguel Santana, Jochen Wendel, Agustín Trujillo, José Pablo, Alexander Simons, and Andreas Koch. Multimodal Location Based Services - Semantic 3D city data as Virtual and Augmented Reality. In G Gartner and H Huang, editors, *Progress in Location-Based Services 2016*, chapter 17. Springer, 2016. ISBN 978-3-319-47289-8. doi: 10.1007/978-3-319-47289-8{\\_}17.