Physically-Based Atmosphere Rendering

Peter Z. Kutz

Advisor: Norman I. Badler

University of Pennsylvania

ABSTRACT

In computer generated imagery, as in photography, the sky is very ubiquitous and important—as a source of light, as a backdrop, and as a subject in and of itself. However, realistic and aesthetically appealing skies are also very difficult to render. Models exist to approximate the appearance of the sky, but they make many assumptions and have many limitations. To create truly realistic images a full physical simulation of atmospheric scattering, including clouds, is necessary. For this project, I have simulated and produced photorealistic and aesthetically pleasing renders of atmospheres, lit by the sun.

Project Blog: http://skyrenderer.blogspot.com/

For the clearest and most detailed description of this project, and for many more pictures, please see my blog.

1. INTRODUCTION

Realistic and dynamic skies are very difficult to render. Analytic sky models can be used to model the appearance of the sky, but they make assumptions and have many limitations, such as only working from the point of view of ground, not taking ground albedo into account, not working for twilight, not modeling clouds, etc. Image-based lighting in the form of HDR environment maps can also replicate real-life lighting, however they are static—only valid for a very specific time and location. Some researchers have tried simulating the atmosphere, which can be very accurate but very slow.

In real life, the sky is very ubiquitous and important. It serves as a source of light. It acts as a backdrop for everything we do. It indicates the time of day, the season, and the weather. It affects our moods. And it serves as a subject of contemplation, admiration, and artwork. The sky can also serve these same purposes in computer graphics. In particular, realistic sky light can make the difference between a flat, unrealistic image and a photorealistic image.

I have written an atmosphere simulation that balances physical accuracy, aesthetic beauty, flexibility, and performance. The simulation works from any location and altitude (including from space), with various input parameters, and, if time permits, with clouds. I will use real-life data and physics where applicable. I have not simulated all of the complex effects of light (including polarization, diffraction, dispersion, and atmospheric refraction)—instead, I have focused on the effects that have the most visual impact, such as Rayleigh and Mie scattering from air, water vapor, dust, and pollutants.

This project makes the following contributions:

• I have developed a new sky rendering system that produces physically accurate and aesthetically pleasing images.

• I have learned about a variety of subjects that I will need to understand to do further research, and subjects that will be highly relevant and useful in the movie and game industries.

1.1 Design Goals

The target audience for my project was anyone who wants to be able to utilize realistic, computer-generated skies and atmospheric effects. The user is able to create realistic and pretty pictures with this technology.

1.2 Projects Proposed Features and Functionality

I have implemented the following features and functionality for my design project:

- Simulated atmospheric scattering using Monte Carlo path tracing, unbiased distance sampling (in place of ray marching), direct sun sampling, various other forms of importance sampling, and various optimizations.
- Used real-life data for the composition of the Earth's atmosphere, and real-life physics for the scattering properties of the atmosphere.
- Included preferential absorption by ozone, which is typically not included in sky models, though it has a very significant effect on the color of the sky, especially when the sun is low or below the horizon.
- Wrote a spectral rendering system, that works for all wavelengths of visible light, as well as ultraviolet and infrared.
- Employed colorimetry to convert spectral data to images that can be displayed on computer displays.
- Generated and saved panoramic HDR images that can be used as light sources in other renderers.

2. RELATED WORK

People in various disciplines have been interested in characterizing the appearance of the sky for many many years. Summarized below are some of the significant past works that I have drawn on while developing my sky renderer.

Over a period of decades in the late 1800s, Lord Rayleigh published a series of papers that explained his theories about atmospheric scattering and how it caused the color and appearance of the sky. In [Ray71] and [Ray99] he formulates and refines his theory of the scattering of light by air molecules, now known as Rayleigh scattering.

[AKP74] examined the effect of ozone absorption and aerosols on the color of the twilight sky using computer simulations.

[McC76] describes turbidity, the scattering of light by haze in the atmosphere. Turbidity is an important aspect of atmospheric scattering.

In 1982, Jim Blinn developed methods for simulating light scattering in volumes in [Bli82]. He brought science from physics literature to computer graphics.

In [Kla87], Klassen simulated the color of the sky using a planar model of the atmosphere and single scattering.

Kaneda et al. simulated the color of the sky using a spherical model of the atmosphere and atmospheric density that decreases exponentially with altitude.

Later, [NDKY96] extended the previous paper to take multiple scattering into account.

Preetham et al. presented an analytic sky model in 1999, which has been widely used ever since. The paper contains many descriptions, data, and references.

[LL01] is a well-reviewed book that covers a large range of light and color phenomenon. It contains both high-level, intuitive descriptions of these phenomenon, as well as technical details. It also contains many photographs and diagrams.

At SIGGRAPH 2012, Hosek and Wilkie introduced a new and improved analytic sky model [HL12] which is more accurate than the Preetham model. They derived their analytic model from a series of brute force simulations. The model and their simulations have limitations though.

3. PROJECT PROPOSAL

Below are my plans for the project, written months ago at the beginning of the project. To summarize, I planned to build a spectral renderer, model the Earth's atmosphere, and simulate atmospheric scattering. I also planned to add clouds if time permitted.

3.1 Anticipated Approach

The atmosphere simulation will work from any location and altitude (including from space), with various input parameters.

For the simulation I will use Monte Carlo path tracing and ray marching. Ray marching will be necessary because the Earth's atmosphere varies as a function of location—altitude in particular. I will use real-life data for the composition of the Earth's atmosphere, using interpolation to create a continuous model of atmospheric composition between data points. I will keep the system as general as possible so that it can also be used for other planets.

I will use real-life physics for the scattering properties of the atmosphere and clouds (under a geometric optics approximation), including Rayleigh scattering for light scattering from air molecules, which is what gives the sky its blue color during the day under normal conditions.

I will model the Earth as a sphere. I will give let the user specify the color of the ground, and I will include diffuse reflection from the ground in the simulation because it has an effect on sky color.

I will implement a spectral rendering system, that will work for all wavelengths of visible light, as well as ultraviolet and infrared. I will also use real radiometric units and data for the simulation. I will employ colorimetry to convert this spectral data to images that can be displayed on computer displays. To enable manipulation and deep compositing, I will save HDR images, along with deep alpha maps and other information.

I will sample the sun directly, otherwise almost all rays would get lost in space and images would take forever to converge. In order to sample the sun directly, I will likely have to ignore atmosphere refraction, which is aesthetically not very significant, except when the sun is very close to the horizon. However, I will think about ways that atmospheric refraction could be simulated efficiently, and add it if I come up with a good idea. I will use the sun's actual emission spectrum for the light that the sun emits, potentially using it for importance sampling wavelengths when ray-casting from the camera.

If I have time leftover after simulating the atmosphere (which I likely will not), I will add clouds. Storing volumetric data for a sky full of clouds with sufficient detail would use a lot of memory. To avoid doing this, I will use proceduralism where applicable. To generate plausible clouds, I will investigate various techniques and technologies including metaballs, advanced data structures (non-uniform grids, frustum-aligned grids, or no grids at all), and Perlin noise. I will try to mimic the appearance of real-life clouds, however they will not be dynamic, physically-based, or the result of a simulation. They should nevertheless add to the realism of the resultant images, and seamlessly integrate with the physically-based atmosphere simulation. The structures of these clouds will be more advanced and fine-tuned than those of the volumetric renderer written in CIS 460, and the clouds will be lit much more realistically by taking into account multiple scattering and realistic phase functions.

3.2 Target Platforms

I will write the simulation in C++. As a starting framework, I will use some pieces of my own 3D renderer (Photorealizer, which I wrote from scratch): a basic Qt GUI, some image processing features (including anti-aliasing, gamma correction, and writing bitmap image files to disk), and code for ray-casting from the camera, and code for generating sampling points on the sun. I will consider using GLM as a linear algebra library.

3.3 Evaluation Criteria

I will render images that are generally very difficult to render, including pretty sunsets, crepuscular rays, and the shadow of the earth that is visible in the atmosphere at twilight. I will compare my sky simulation to existing simulations and to photographs. I will explain all of the science that went into creating the simulation.

4. RESEARCH TIMELINE

Below are some planned project milestones. I have also included a Gantt chart with more detail (it's near the end of the document). This timeline was designed months ago at the beginning of the project.

Alpha Review (October 19, 2012)

· Completed all background reading
· Proposed software framework is functioning with simple base case
· Built a spectral rendering system
· Collected all necessary data

Beta Review (November 20, 2012)

· Modeled the Earth's atmosphere
· Implemented atmospheric scattering and absorption

Before Final Presentation (Mid-December, 2012)

· Clouds, time permitting

Final Presentation and Final Deliverables (Mid-December, 2012)

· High quality rendered images showing various effects
· A timelapse video of a full day
· Live demo of renderer
· Completed report
· Completed renderer

Project Future Tasks

Given 6 more months to work on this project, I would do the following:

· Add clouds or improve my existing cloud implementation.
· Port the program to the GPU using CUDA. The atmosphere simulation will be highly parallelizable.
· Create an analytic version of my sky model using spherical harmonics and interpolation.
· Integrate my sky simulator into my 3D renderer, Photorealizer.
· Compare my results to existing analytic sky models.

5. METHOD

In this section I'll summarize the work that went into creating my sky renderer. See my blog (link on the first page) for even more detail.

5.1 Framework

I began by putting together a C++ framework using some pieces from Photorealizer, my own 3D renderer which I wrote from scratch. The framework included a basic Qt GUI, some image processing features (including anti-aliasing, gamma correction, and writing bitmap image files to disk), a basic linear algebra library, code for ray-casting from the camera, code for generating sampling points on the sun, multithreading support, and various other utilities.

5.2 Geometry

I modeled the Earth and the edge of the atmosphere using two giants spheres. I decided that I would work in SI units for this project to keep everything consistent and achieve physically accurate results. I made the Earth's surface to have an albedo of 31%, which seems to be the accepted average albedo of Earth's surface.

5.3 Spectral Rendering System

I wrote a spectral rendering system that associates each ray with a wavelength instead of an RGB color. Initially, in order to convert the spectral data to RGB primaries, I made up response curves for the camera's RGB sensors, but this was just a rough placeholder system. I considered implementing the response curves of a real-life camera, however, I decided on a more scientific approach. I decided to convert my spectral data to CIE XYZ by integrating it with the CIE XYZ color matching functions (specifically, the CIE 1931 2° XYZ CMFs modified by Judd and Vos) [Vos78], then transforming XYZ to linear RGB (Rec. 709 primaries), then finally transforming that to sRGB. This approach yields colorimetrically accurate results and is very flexible. I'm very interested in color, so doing this was also fun and good practice.

5.4 Rayleigh Scattering

The coloration of the sky is primarily a result of Rayleigh scattering, at least on clear days. Rayleigh scattering describes the scattering of light by particles much smaller than the wavelength of the light. In the case of the sky, light is scattered by the molecules of air themselves. The amount of scattering is inversely related to the fourth power of the wavelength of the light. This means that bluer light is scattered much more than redder light, which results in the blue color of the sky. And when the sun is near the horizon, its light needs to pass through more atmosphere to reach you, so even more blue light is scattered out of the path, and mostly just longer wavelengths remain, giving the sun it's vivid orangish color.

I implemented Rayleigh scattering from the ground up, including scattering cross sections, scattering coefficients, and phase functions. I am aiming for physical accuracy (and trying to learn about the physics) so I even included subtle factors such as dispersion of both the index of refraction and depolarization factor of air. The main resource I used in implementing Rayleigh scattering was [Buc95].

5.5 Modeling Earth's Atmosphere

In order to realistically simulate the appearance of the sky, I needed an accurate model of the composition of the Earth's atmosphere. To this end, I implemented the U.S.

Standard Atmosphere 1976, using the original paper from NASA, NOAA, and the USAF [NNU76]. I could have used a simple exponential density falloff, but that wouldn't have been as accurate, or as interesting to implement. I also could have used an existing implementation of the U.S. Standard Atmosphere, but I wouldn't have learned as much from that, and it wouldn't have been very fun.

The U.S. Standard Atmosphere uses SI units, and I converted to SI any values that were provided in other units. I also had to spot a few errors in the paper, where incorrect values were provided. It was fun working on a project that uses real-world units and measurements. Not just fun, but also necessary to achieve the correct results.

5.6 Unbiased Distance Sampling

Because the properties of the atmosphere vary with altitude, it's not possible to simply use the scattering and absorption properties at a single point to compute scattering and absorption along a ray.

At first, I implemented a ray-marching system, with adaptive step sizes based on a few criteria, however that system was biased and somewhat slow so I replaced it.

To replace ray marching, I implemented an algorithm for unbiased distance sampling in heterogeneous media. It's a Monte Carlo algorithm, "Algorithm 1" in [KRS07]. I learned about that paper from Hosek and Wilkie's sky model paper [HW12].

The algorithm samples a random distance based on the highest extinction (or scattering or absorption) coefficient along the ray (or line segment in my case), or any coefficient greater than or equal to that (such as the highest in the entire medium, which is often easier to find). Then it probabilistically takes another leap based on the actual coefficient at the sampled location.

Using the lowest max coefficient allows larger steps to be taken, which decreases the number of steps needed and increases performance. Since Rayleigh and ozone scattering and absorption coefficients vary predictably based on wavelength, I was able to find the lowest max coefficient for any given wavelength, and with Rayleigh scattering, I realized that I could find the lowest coefficient along the line segment by using the coefficient at the point on the line segment that is closest to the center of the Earth (assuming density is monotonically decreasing with altitude).

5.7 Sampling the Sun Directly

All of the light in my simulator comes from the sun. The sun covers only a tiny fraction of the sky, about 0.00047%, so with naive path tracing, most rays would get lost in space and the image would take forever to converge. By sampling the Sun directly, I was able to speed up my simulation tremendously. At each scattering point, I fire a ray directly towards the Sun, weighted by the Rayleigh scattering phase function in that direction, and weighted by the probability that a ray fired in a random direction would hit the sun. I also still fire a scattered ray in a direction drawn from the Rayleigh scattering phase function, however this ray cannot "see" the Sun. If I were to only sample the Sun, I would lose multiple scattering, which plays an important role in the appearance of the sky. I compute sample directions that are uniformly distributed over the solid angle subtended by the Sun (the Sun is not a point light or a directional light).

5.8 Spectral Solar Irradiance

I implemented the spectral irradiance of the sun based on modern AM0 data. AM0 (Air Mass Zero) means that the data is for sunlight that hasn't passed through any atmosphere, which is what I needed because I am simulating the atmosphere myself. This extraterrestrial spectral solar irradiance data is often used for space applications. AM1.5 data is also available, for ground level applications such as solar energy. I found the data on NREL's website, and it originally came from a 2003 paper by Guemard [Gue03]. In my renderer I convert irradiance to radiance instead of using the data directly. I could have alternatively used a black body radiation curve to approximate the spectral irradiance, however spectral solar irradiance does not exactly follow a black body radiation curve.

5.9 Ozone Absorption

In real life, the twilight zenith sky appears a deep blue. This blueness is due to preferential ozone absorption over long paths through the atmosphere [Hul53][AKP74]. In particular, the Chappuis absorption band of ozone absorbs light of longer wavelengths in the visible spectrum, thus making the sky appear more blue. Without ozone, the twilight sky would appear almost gray. Rayleigh scattering contributes relatively little to the blueness of the twilight sky.

I decided that I needed to include ozone absorption in my sky renderer. To accurately model ozone's absorption spectrum, I found recent (2011), high-precision, absorption cross-section measurements [BCGSW12]. The data was provided in 0.01 nanometer increments, which was too precise for my purposes, so I converted the data to 5 nanometer increments, averaging 500 data points to create each new data point. The data is provided for 11 different temperatures. They're similar enough that I probably could have just used one, or averaged them all together, but I decided to use all of them anyway. When a query is made, I just look up the data at the closest temperature and wavelength—I decided that interpolation was probably overkill in this case.

To convert the absorption cross-sections (units of cm^2/molecule) to absorption coefficients (units of cm^-1), I multiply by the molecular number density of standard air (units of molecules/cm^3), multiply by the concentration of ozone in the atmosphere (unitless), and then correct for altitude by multiplying by the relative density (unitless) at the altitude in question.

I implemented a realistic distribution of ozone. Average data for the Earth's atmosphere was difficult to find, and the exact distribution of ozone is pretty variable anyway, so I estimated the distribution based on a NASA graph, and made a few tweaks based on other information I had found.

5.10 Panorama Cameras

In addition to a standard rectilinear projection camera, I made two special cameras that allow the entire sky to be viewed in a single image.

The first uses latitude-longitude panoramic format, also known as an equirectangular mapping. This is the same format that I use for HDR environment maps in Photorealizer, and my sky renderer already saves images to EXR (in addition to PNG), which allows me to use my sky renderer renders as HDR environment maps for my Photorealizer renders. Plus, I have HDR environment map importance sampling in Photorealizer, so I can leave the tiny, bright, influential sun in the sky images and Photorealizer will automatically know to heavily sample the sun.

The second camera uses an angular fisheye projection. This is a pretty intuitive way to view the entire sky (and part of the ground in my implementation).

5.11 Aerosols

I added aerosols to the atmosphere. Aerosols have a significant impact on the appearance of the sky, but simulating them isn't particularly straightforward: the types and levels of aerosols vary widely by region (e.g., city vs. forest vs. ocean) and other factors, and computing their scattering properties is quite complicated.

I used the representative profile of aerosols in [Elt68]. That paper provides aerosol extinction coefficients that vary based on altitude and wavelength. It's the same data used in [AKP74], which is what led me to the paper in the first place. I used the provided scale height relationship (scale height of 1.2 km) to extrapolate the data beyond 50 km.

For the scattering phase function, I used the Henyey-Greenstein phase function, including analytic scattering direction sampling. I used a constant asymmetric parameter of 0.7 (which means that the mean cosine of the scattering angle is 0.7, which implies strong forward scattering), which seems to be a pretty good average value based on my research.

I used a constant single scattering albedo of 0.9, which, like my asymmetry parameter, seems to be a pretty good average value based on my research. At the sampled extinction distance, I scatter the photon/ray with 90% probability, and absorb it with 10% probability.

6. FUTURE WORK

6.1 Aerosols

My aerosols system could be improved in many ways. In particular, I could use a realistic particle size distribution, use realistic particle type proportions (each type having a certain index of refraction, with real and imaginary parts), and then compute scattering and absorption properties using the Mie solution to Maxwell's equations. That sounds like it might be overkill for now, although I'm sure I would learn a lot in the process. It would also be good to implement Cornette and Shanks's modified version of the Henyey-Greenstein phase function [CS94], which better approximates actual Mie scattering phase functions and converges to the Rayleigh phase function as the asymmetry parameter approaches zero.

6.2 Terrain

At some point, I would like to make a procedural ground, or use a map of the actual Earth. Procedural mountains would be particularly nice for showing off atmospheric effects.

6.3 GPU Acceleration

Because my sky renderer is based on highly parallelizable Monte Carlo path tracing, because it is very computation-heavy, and because it only needs to access a relatively small amount of information from memory, its performance could be greatly enhanced using the GPU.

6.4 Analytic Sky Model

I would be interested in creating an analytic sky model based on my atmosphere simulations.

6.5 Clouds

Clouds can have a large affect on the appearance of the sky. It would be cool to add clouds to my sky renderer. In addition to increasing the robustness and realism of the renderer, clouds would allow me to render especially pretty images of sunset and twilight.

7. RESULTS

My sky renderer is capable of generating highly realistic images of skies based on real-life data and physics. It performs unbiased Monte Carlo path tracing simulations of atmospheric scattering (including multiple scattering) and absorption. It can render images at any time of day, even during twilight when ozone strongly influences the color of the sky, and when the Earth's shadow is visible in the atmosphere. It can render from any vantage point, including outer space. Because of its high level of accuracy, it isn't the fastest sky rendering system, but it does include many optimizations that make it very fast for an unbiased path tracer, and it would be useful for a variety of offline applications.

I've included some images from my sky renderer at the end of this document, as well as some renders that use sky renderer images as light sources.

8. CONCLUSIONS

I created a very robust, unbiased, accurate, and physically-based sky renderer. In the process I learned a lot about many fields including computer graphics, physics, mathematics, atmospheric sciences, color, and more. I was also able to utilize and incorporate much lots of my existing knowledge and past experience. Finally, I gained valuable thinking skills and experience by working on a project that involved a large amount of research and creativity.

REFERENCES

[AKP74] ADAMS, C. N., KATTAWAR, G. W., PLASS, G. N.: The Influence of Ozone and Aerosols on the Brightness and Color of the Twilight Sky. Journal of the Atmospheric Sciences 31 (April 1974).

[BCGSW12] BURROWS, J. P., CHEHADE, W., GORSHELEV, V., SERDYUCHENKO, A., WEBER, M.: High spectral resolution ozone absorption cross-sections. IUP, MolSpec Lab (2012).

[Bli82] BLINN, J. F.: Light reflection functions for simulation of clouds and dusty surfaces. ACM SIGGRAPH Computer Graphics 16 (July 1982), 21–29.

[Buc95] BUCHOLTZ, A.: Rayleigh-scattering calculations for the terrestrial atmosphere. (1995).

[CS94] CORNETTE, W., SHANKS, J. G.: Physically reasonable analytic expression for the single-scattering phase function. Appl. Opt., 31 (1994), 3152-3160.

[Elt68] ELTERMAN, L.: UV, Visible, and IR Attenuation for Altitudes to 50 km. (1968).

[Gue04] GUEYMARD, C.: The Sun's total and spectral irradiance for solar energy applications and solar radiation models. Solar Energy, 764, 4 (April 2004), 423-453.

[Hul53] HULBERT, E. O.: Explanation of the Brightness and Color of the Sky, Particularly the Twilight Sky. Journal of the Optical Society of America 43, 2 (February 1953).

[HW12] HOSEK, L., WILKIE A.: An Analytic Model for Full Spectral Sky-Dome Radiance. ACM TOG 31, 4 (2012).

[Kla87] KLASSEN, R. V.: Modeling the effect of the atmosphere on light. ACM Transactions on Graphics 6, 3 (1987), 215–237.

[KNN91] KANEDA, K., OKAMOTO, T., NAKAME, E., NISHITA, T.:: Photorealistic image synthesis for outdoor scenery under various atmospheric conditions. The Visual Computer 7, 5 and 6 (1991), 247–258.

[KRS07] KELLER, A., RAAB, M., SEIBERT, D.: Unbiased Global Illumination with Participating Media. (2007).

[LL01] LYNCH, D. K., LIVINGSTON, W.: Color and Light in Nature. Cambridge University Press, 2001.

[McC76] MCCARTNEY, E. J. Optics of the Atmosphere. Wiley publication (1976).

[NDKY96] NISHITA, T., DOBASHI, Y., KANEDA, K., YAMASHITA, H.: Display method of the sky color taking into account multiple scattering. Pacific Graphics ’96 (1996), 117–132.

[NNU76] NASA, NOAA, USAF: U.S. Standard Atmosphere 1976. (1976).

[Pre03] PREETHAM, A. J.: Modeling Skylight and Aerial Perspective. (2003).

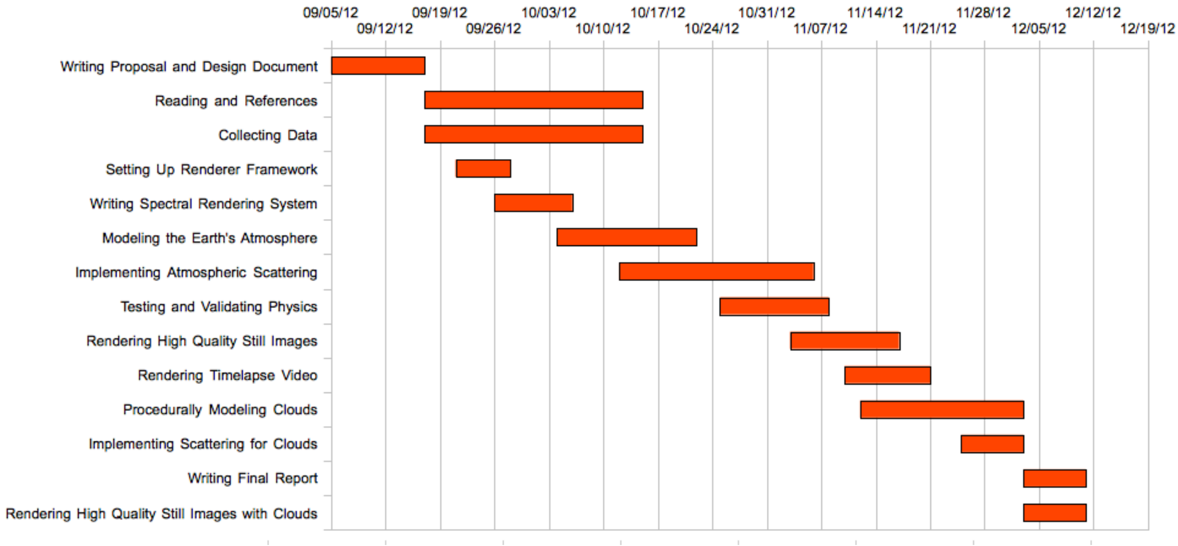
[PSS99] PREETHAM, A. J., SHIRLEY, P., SMITS, B.: A practical analytic model for daylight. Proceedings of the 26th annual conference on computer graphics and interactive techniques (1999), 91–100.

[Ray71] RAYLEIGH, L.: On the scattering of light by small particles. Philosophical Magazine 4, 41 (1871), 447–454.

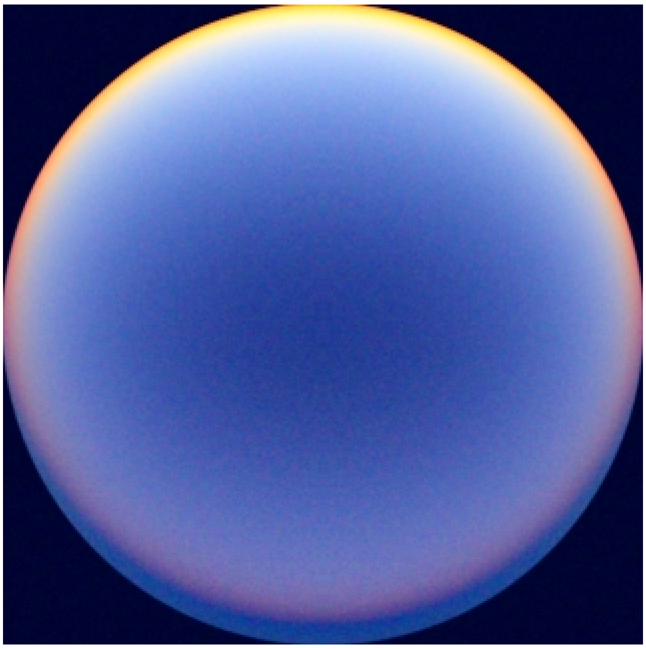
[Ray99] RAYLEIGH, L.: On the transmission of light through an atmosphere containing small particles in suspension, and on the origin of the blue of the sky. Philosophical Magazine 5, 47 (1899), 375–394.

[Vos78] VOS, J. J.: Colorimetric and photometric properties of a 2-deg fundamental observer. Color Research and Application, 3 (1978), 125-128.

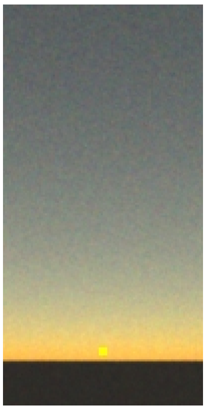
IMAGES



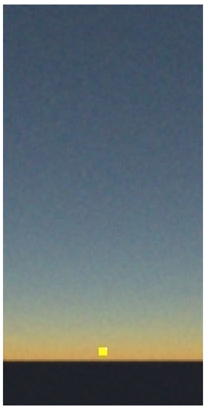
Gantt chart showing project schedule as I originally planned it.



A fisheye shot of twilight. This was rendered before I added aerosols to the atmosphere. The dark blue area at the bottom is the shadow of the Earth which is cast into the atmosphere when the sun is below the horizon.



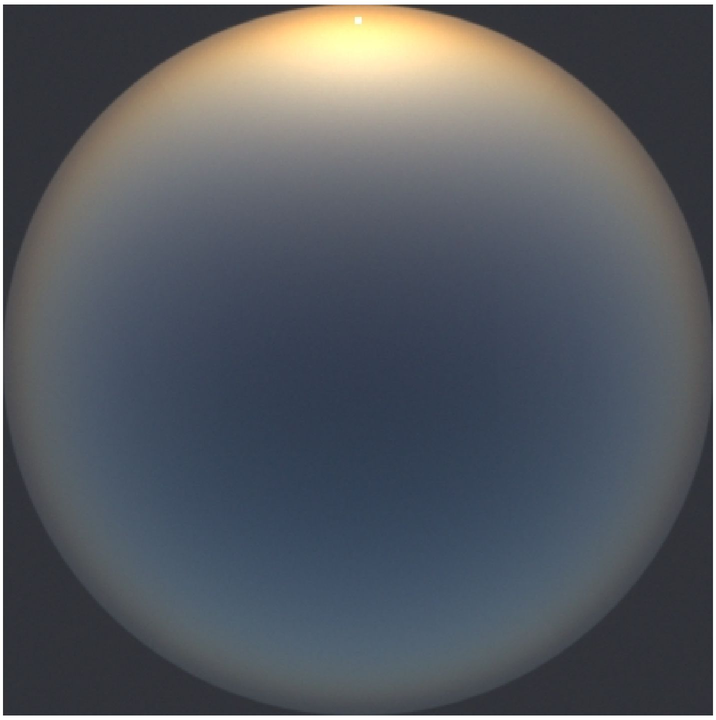
A render of sunrise/sunset before adding ozone absorption.



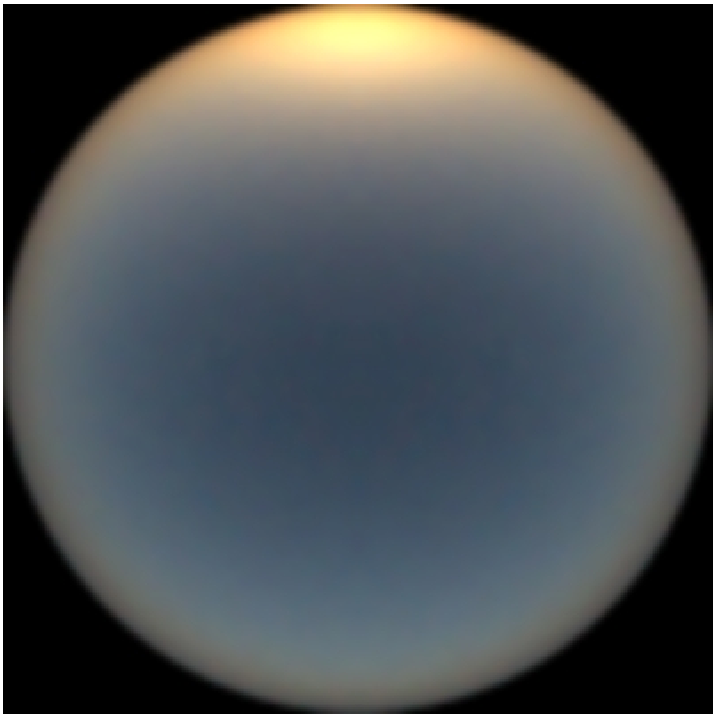
A render of sunrise/sunset after adding ozone absorption.

A render of the Earth's atmosphere, as seen from space.

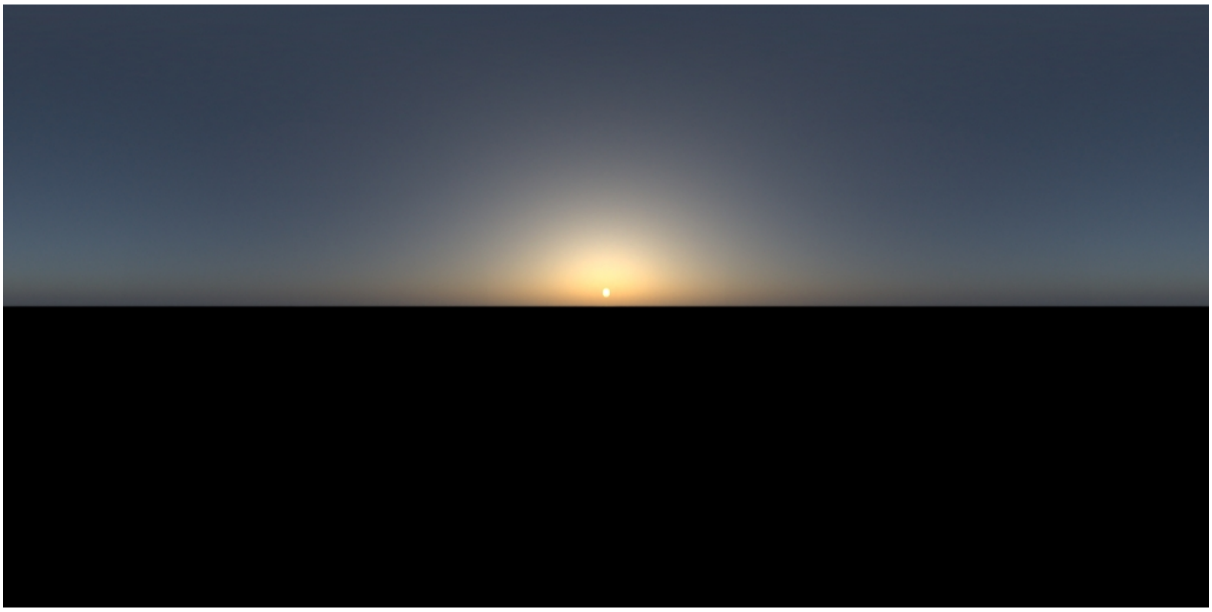
Another render of the Earth's atmosphere, as seen from space. This time the sun is below the horizon.



A sunrise/sunset sky with aerosols, rendered in my atmosphere path tracer.

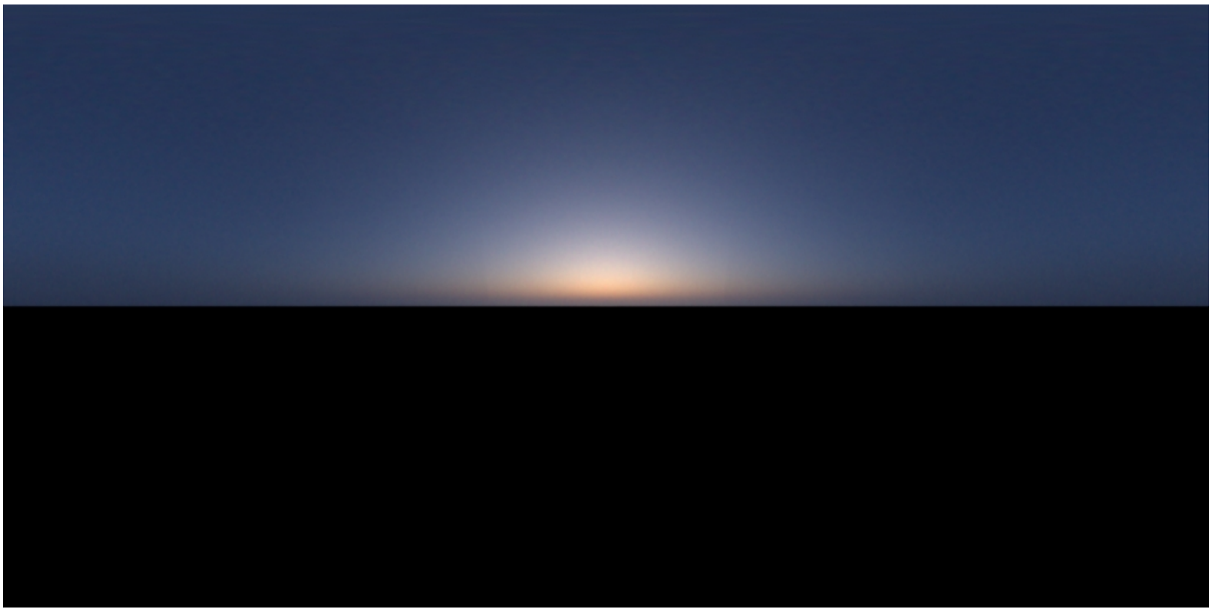


A sunrise/sunset sky with aerosols, rendered in Hosek and Wilkie's atmosphere path tracer (image found in [HW12]).



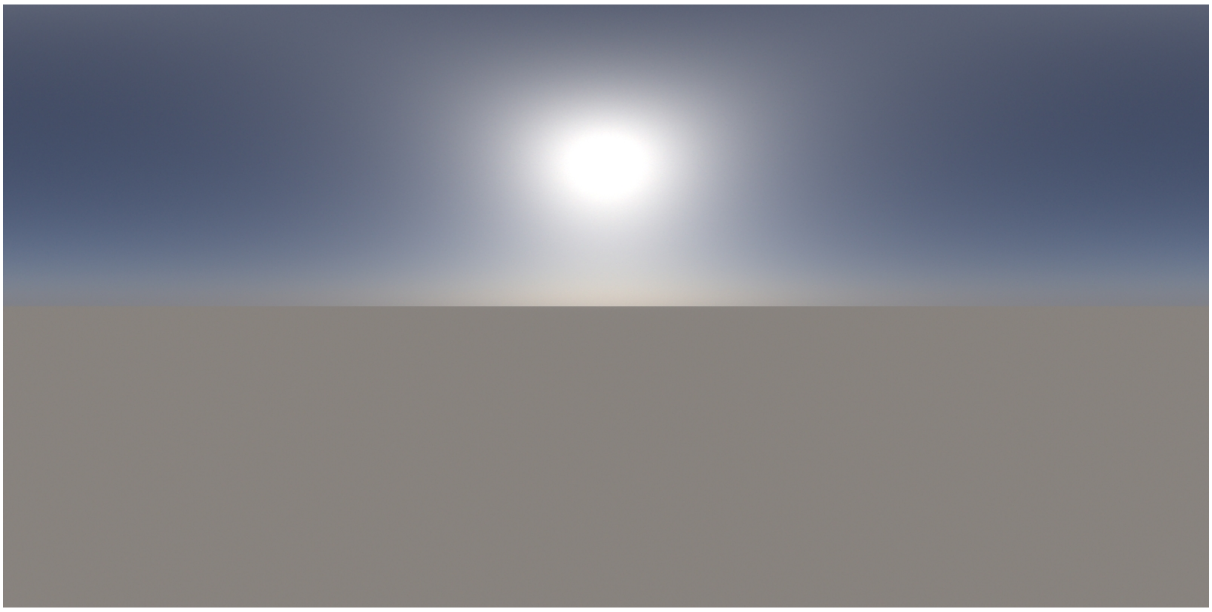
An equirectangular version of a sunrise/sunset render.

A bunny rendered in Photorealizer using the HDR version of the image above as a light source.



An equirectangular version of a twilight render.

A bunny rendered in Photorealizer using the HDR version of the image above as a light source.



An equirectangular midday render.

A bunny rendered in Photorealizer using the HDR version of the image above as a light source.