Illustration 1: Sunrise

Daylight Environment Mapping

a CS497 implementation

by Patrick Lacz

I chose to implement "A Practical Analytic Model for Daylight" by A. Preetham, Peter Shirley, and Brian Smits. My main motivation in doing so (in implementing a paper that was not presented), was a desire to create better outdoor scenes, and to potentially reuse code from my interference ray-tracer, which also handles spectrum operations and color space manipulations.

I wrote the application as two programs: a renderer and a display program. As an output of the renderer, I wanted all the required precomputation for displaying a sky at a given time, location, and atmosphere. The program then output a tiff file of tone-mapped radiances for a given θ and φ.

The program could also output four additional precomputations: attenuation and in-scattering constants over θ and φ, and more in-scattering computation over distance and θ. For the demo, I did not use these tables; instead they were included in the output environment map as part of the radiance computation.

The implementation of the paper uses more wavelength samples than I do, and have implemented a renderer that is able to apply so-called aerial effects (such as the attenuation and in-scattering) to items in the scene other than the sky itself. I also did not include the effects of "mixed gasses" and water absorption, because these particles only scattered at wavelengths I was not sampling (greater than 700nm). These are features not implemented in my version.

In the sample images, several important features are noticeable: The haze around the bottom of the image is entirely modeled as a combination of in-scattering and attenuation parameters. The orb of the sun itself is modeled by a interpolated fall-off function based on the angular difference from the sun's direction. All other hues are computed from the zenith color.

One major parameter of the image that was not addressed by the paper is a proper conversion to RGB. I used a straight-forward tone-mapping scheme by Ward that used an average luminosity to scale all brightnesses to a convenient range. I suspect a more sophisticated function may preserve relative brightnesses between different lighting conditions better.

The second program had high hopes of using the precomputed data as lookup tables to do lighting calculations per-pixel. Due to a few missing instructions (arccos, arctan), I was temporarily reduced to simply showing the computed result as an environment map.


Illustration 2: *Early Morning on a hazy day.*

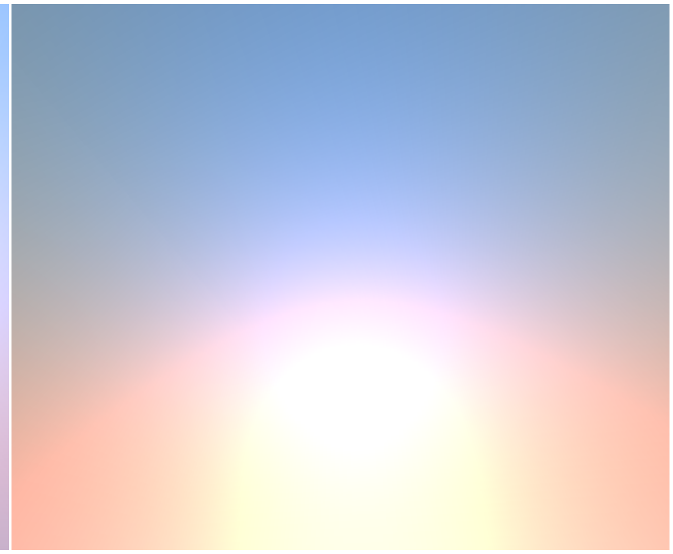
Illustration 3: *Sunset on a hazy day.*


Illustration 4: Looking north at sunset.


Illustration 5: Noon on a clear day.

Obviously, the model could be improved by the inclusion of cloud modelling. With higher turbidities, uniformly overcast days can be modelled, however this currently causes problems with my renderer, so no images are available.