Volumetric Scattering

Steve Rotenberg

CSE168: Rendering Algorithms

UCSD, Spring 2014

Volumetric Scattering

- *Volumetric scattering* refers to light scattering in volumetric phenomena like clouds, smoke, fire, and translucent materials like wax and human skin
- Sometimes, the medium responsible for the scattering is called a *participating medium*
- Without volumetric scattering, the radiance along a straight ray between two surfaces is constant
- With participating media, the media affect the radiance along the ray

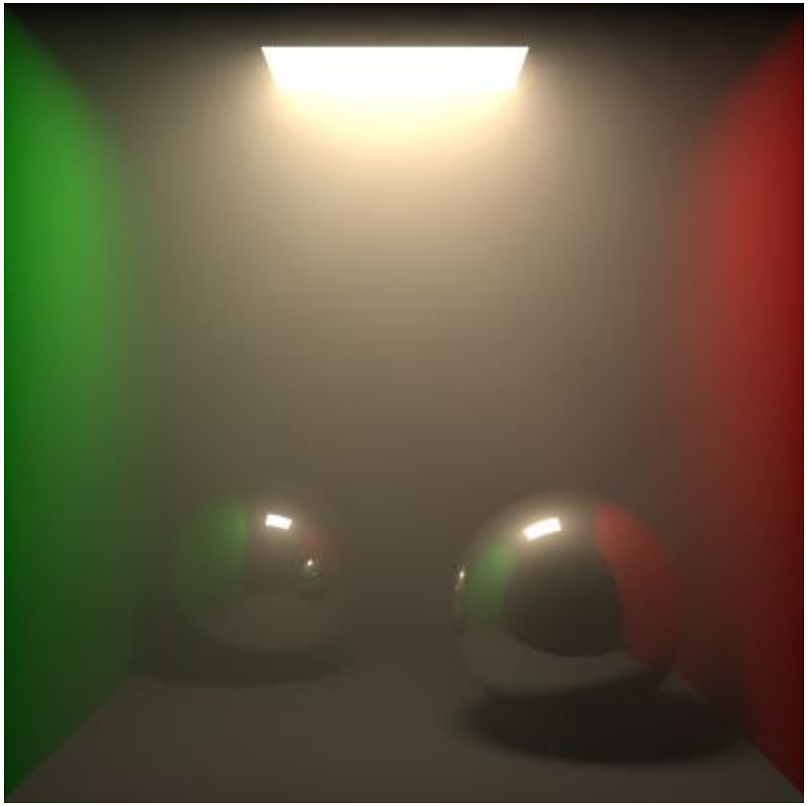
Volumetric Scattering

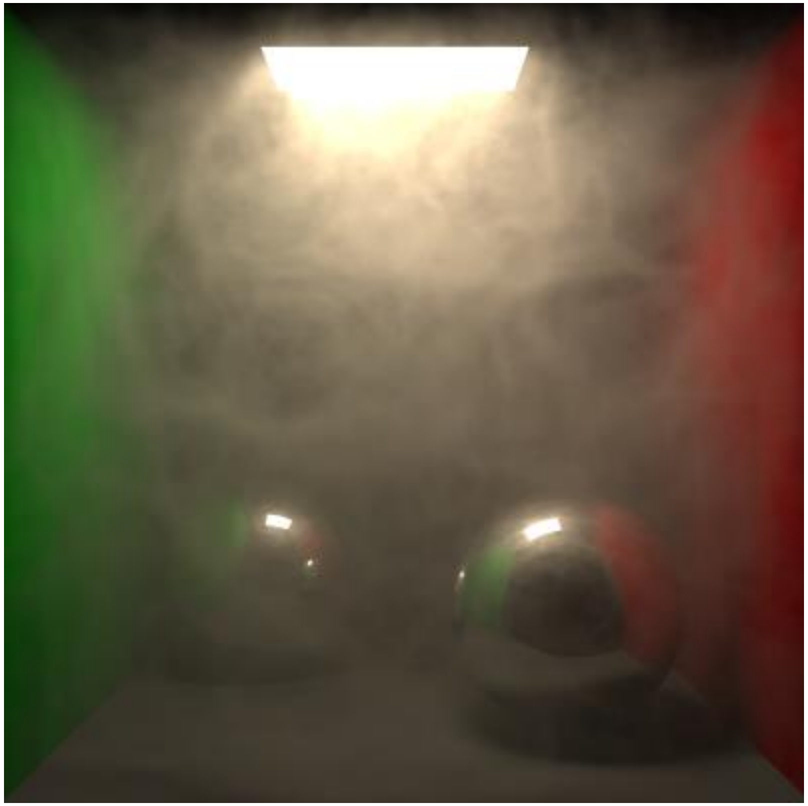
Volumetric Scattering

Homogeneous vs. Heterogeneous

- With some volumes, we assume that their properties are constant everywhere
- We refer to these as *homogeneous*
- Fog can often be represented as a homogeneous medium
- Other volumetric phenomena vary across space
- These are called *heterogeneous* or *inhomogeneous*
- Smoke, fire, and many interesting volumes have spatially varying properties and are heterogeneous
- Homogeneous volumes are a little easier to deal with and allow for some computational short cuts

Homogeneous vs. Heterogeneous





Volumetric Interactions

- There are 4 different processes that affect the radiance of a beam through a participating medium
  - Emission
  - Absorption
  - Out-scattering
  - In-scattering

Emission

- Some volumes emit light, such as fire and neon gas
- Emission is usually isotropic (uniform in all directions)

$$dL(\mathbf{x}, \boldsymbol{\omega}) = \frac{1}{4\pi} E(\mathbf{x}) ds$$

- $dL(\mathbf{x}, \boldsymbol{\omega})$ is the rate of change of radiance of a beam passing through location **x** in direction **ω** along the beam of length *ds*
- *E*(**x**) is a function that describes the isotropic emission per length at the point **x**, which gets divided by 4π to describe the emission per solid angle

Absorption

- Some media absorb light and convert it to other forms of energy (such as heat)
- We encountered this process already when we considered the absorption of light in dielectrics according to the Beer-Lambert law
- Absorption in a volume is described by the absorption coefficient σ_a, which describes the rate of absorption in units of 1/meter

$$dL(\mathbf{x},\boldsymbol{\omega}) = -\sigma_a(\mathbf{x})L(\mathbf{x},\boldsymbol{\omega})ds$$

- The rate of loss of radiance for a beam of light through point **x** in direction **ω** is equal to absorption coefficient times the radiance *L* of the beam times the differential length *ds*

Out-Scattering

- Some of the light that passes through a medium may be scattered away in various directions
- The rate of scattering is described by the scattering coefficient σ_s

$$dL(\mathbf{x}, \boldsymbol{\omega}) = -\sigma_s(\mathbf{x}) L(\mathbf{x}, \boldsymbol{\omega}) ds$$

Extinction Coefficient

- When a beam of light passes through a volume, some of the light is lost to absorption and some is lost to out-scattering
- Often these two coefficients are combined into a single extinction coefficient

$$\sigma_t = \sigma_a + \sigma_s$$

Transmittance

- We define the *transmittance* $T(s)$ along a path of length s as the percentage of light that is passed through the medium and not absorbed or out-scattered away

$$dL(\mathbf{x}, \boldsymbol{\omega}) = -\sigma_t(\mathbf{x}) L(\mathbf{x}, \boldsymbol{\omega}) ds$$

$$\frac{dL}{ds}(\mathbf{x}, \boldsymbol{\omega}) = -\sigma_t(\mathbf{x}) L(\mathbf{x}, \boldsymbol{\omega})$$

$$\tau(s) = \int_0^s \sigma_t(\mathbf{x} + s'\boldsymbol{\omega}) ds'$$

$$L(\mathbf{x} + s\boldsymbol{\omega}, \boldsymbol{\omega}) = e^{-\tau(s)} L(\mathbf{x}, \boldsymbol{\omega}) = T(s) L(\mathbf{x}, \boldsymbol{\omega})$$

- $\tau(s)$ is called the *optical thickness*

Beer-Lambert Law

- For homogeneous media the extinction coefficient is constant and the optical thickness becomes

$$\tau(s) = \int_0^s \sigma_t(\mathbf{x} + s'\boldsymbol{\omega})ds' = \int_0^s \sigma_t ds' = \sigma_t s$$

- The light loss due to extinction in a homogeneous medium reduces to the Beer-Lambert law:

$$T(s) = e^{-\sigma_t s}$$

In-Scattering

- In-scattering refers to the increase in radiance along the beam due to light coming from other directions that gets scattered into the beam direction

$$dL(\mathbf{x},\boldsymbol{\omega}) = \sigma_s(\mathbf{x})ds \int_{\Theta} p(\boldsymbol{\omega}_i \cdot \boldsymbol{\omega})L(\mathbf{x},\boldsymbol{\omega}_i)d\boldsymbol{\omega}_i$$

- It is very similar to the radiance equation that describes the reflected radiance off of a surface in a particular direction
- It is an integral of the incoming radiance *L* over the sphere of directions $\boldsymbol{\omega}_i$, scaled by the phase function *p*()
- The phase function *p*() describes the distribution of scattered light around the sphere, and behaves very similarly to a BRDF

Light Interaction

Emission:

$$dL(\mathbf{x}, \boldsymbol{\omega}) = \frac{1}{4\pi} E(\mathbf{x}) ds$$

Absorption:

$$dL(\mathbf{x}, \boldsymbol{\omega}) = -\sigma_a(\mathbf{x}) L(\mathbf{x}, \boldsymbol{\omega}) ds$$

Out-Scattering:

$$dL(\mathbf{x}, \boldsymbol{\omega}) = -\sigma_s(\mathbf{x}) L(\mathbf{x}, \boldsymbol{\omega}) ds$$

In-Scattering:

$$dL(\mathbf{x}, \boldsymbol{\omega}) = \sigma_s(\mathbf{x}) ds \int_{\Theta} p(\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}) L(\mathbf{x}, \boldsymbol{\omega}_i) d\boldsymbol{\omega}_i$$

Scattering Phase Functions

Scattering Phase Functions

- When a beam of light interacts with a volume, it is scattered in some spherical distribution
- This process is very similar to how light scatters off of a surface
- For volumetric scattering, we use a *phase function* to describe the shape of the scattering distribution
- This is analogous to a BRDF that we use for surface scattering
- The phase function varies based on the angle between the incoming and outgoing direction, and is usually parameterized by the cosine of that angle

$$p(\cos\theta) = p(\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_s)$$

- Like BRDFs, volumetric phase functions must be reciprocal and must conserve energy
- Also, like BRDFs, we will want to do both forward evaluation and importance sampling of the phase function

Isotropic Scattering

- The simplest phase function is just an isotropic scattering function

$$p(\cos\theta) = \frac{1}{4\pi}$$

- This scatters equally across all directions in the sphere

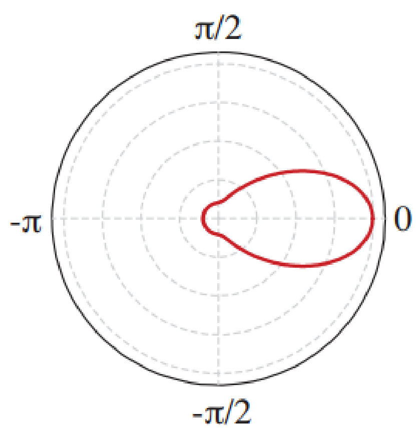
Mie Scattering

- The Lorenz-Mie theory describes the scattering of electromagnetic waves by spherical particles
- It can be used to describe the scattering functions that occur when the size of the particles are at the same scale of the wavelength of the light
- This happens with water droplets in the atmosphere as well as fat droplets in milk
- Names after Gustav Mie (1869-1957) and Ludvig Lorenz (1829-1891)

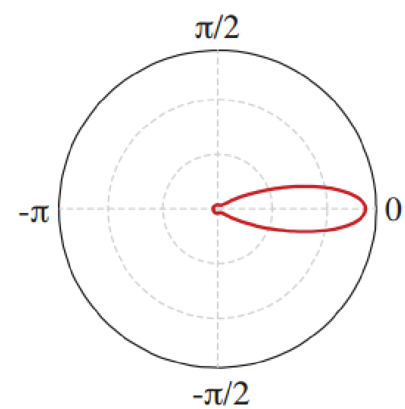
Empirical Mie Approximation

- The following empirical function is often used to approximate the shape of Mie scattering

$$p(\cos\theta) = \frac{1}{4\pi}\left(\frac{1}{2} + \frac{(z+1)}{2}\left(\frac{1+\cos\theta}{2}\right)^{z}\right)$$



z=8



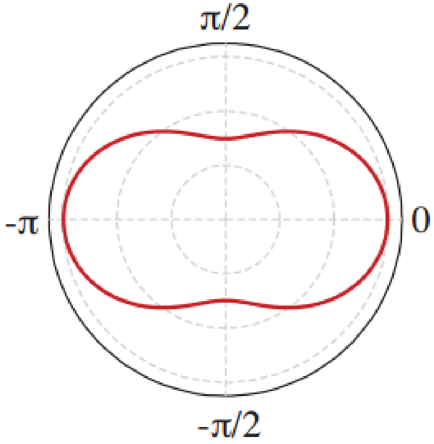
z=32

Rayleigh Scattering

- Rayleigh scattering describes the scattering of light by particles much smaller than the wavelength

$$p(\cos\theta) = \frac{3}{16\pi}(1 + \cos^2\theta)$$

$$\sigma_s = \frac{2\pi^5}{3}\frac{d^6}{\lambda^4}\left(\frac{n^2 - 1}{n^2 + 2}\right)^2$$

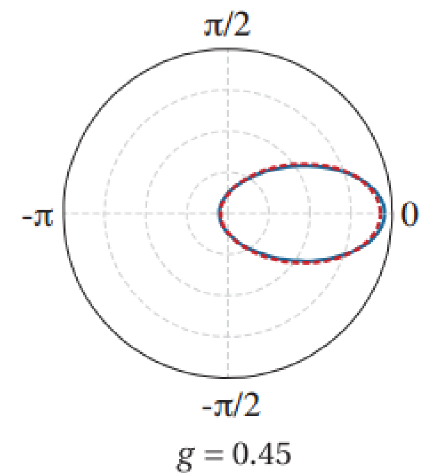


- Where λ is the wavelength of light, d is the diameter of the particle, and n is the index of refraction of the particle
- The blue color of the sky is caused by Rayleigh scattering of sunlight by air molecules
- The strong dependence on wavelength (λ^{-4}) causes greater scattering towards the blue end of the spectrum

Henyey-Greenstein Function

- The Henyey-Greenstein phase function is an empirical function originally designed to model the scattering in galactic dust clouds

$$p(\cos\theta) = \frac{1 - g^2}{4\pi(1 + g^2 - 2g\cos\theta)^{1.5}}$$



- It uses an anisotropy parameter *g* that ranges between -1 (full backscatter) and 1 (full forward scatter), and is isotropic for *g*=0

Volumetric Rendering

Volume Representation

- The first issue that we need to address in order to add volumetric scattering to a renderer is the issue of how the volume is represented
- We want a flexible system that allows us to add volume types, so we'll want a Volume base class similar to the existing Object base class
- The Volume would need a virtual function that evaluates the volume at some position $\mathbf{x}$, and returns the absorption and scattering coefficients, emission, and scattering phase function at that point
- The Volume will also require an intersection routine to determine if a ray passes through the volume
- We also need a scattering phase class similar to the Material class that has both forward evaluation and importance sampling (random sample generation) virtual functions

Ray Marching

- As a ray passes through a volume, we need to accumulate the changes in radiance along the ray due to emission, absorption, out-scattering and in-scattering
- As these changes vary along the ray, we need to divide up the ray into finite segments and approximate these values along the segments
- We call this process *ray marching*
- We need some maximum step size, which could be just set as a constant, or could be tunable per volume or even adapt based on local properties of the medium
- Each actual step we take as we march along the ray is a random number from 0 to 1 times the maximum step size
- This way, each ray randomly samples at several points in the volume

Ray Marching

- For each segment in the march, we need to compute the change in radiance due to the emission, absorption, out-scattering, and in-scattering
- Absorption and out-scattering can be combined into a single extinction calculation, where we assume a constant extinction coefficient along each segment
- Emission along the finite segment can be approximated as the emission at point **x** times the length *s* of the segment, times the transmittance of the segment
- In-scattering at point **x** is represented by $L_i(\mathbf{x})$

$$L(\mathbf{x} + s\boldsymbol{\omega}, \boldsymbol{\omega}) \approx e^{-\sigma_t(\mathbf{x})s}\left(L(\mathbf{x}, \boldsymbol{\omega}) + E(\mathbf{x})s + L_i(\mathbf{x})s\right)$$

In-Scattering

$$L(\mathbf{x} + s\boldsymbol{\omega}, \boldsymbol{\omega}) \approx e^{-\sigma_t(\mathbf{x})s}(L(\mathbf{x}, \boldsymbol{\omega}) + E(\mathbf{x})s + L_i(\mathbf{x})s)$$

- The change in radiance along a finite segment due to extinction and emission are straightforward computations
- However, the in-scattered radiance $L_i(\mathbf{x})$ is an integral of incoming radiance around a sphere, and must be approximated
- We would like to know the total light coming into the point from all directions around a sphere
- This is very much like the surface shading problem, where we want to know the total light coming in across a hemisphere
- Like surface shading, which is often broken into direct and indirect components, we compute the volumetric in-scattering in a similar way
- The in-scattering at a point is the sum of the scattering coming directly from the light sources and the scattering from indirect light coming from other directions
- We can evaluate this by tracing shadow rays to the light source and generating sample scattering rays that we trace into the environment

In-Scattering

$$L_i(\mathbf{x}, \boldsymbol{\omega}) = \sigma_s(\mathbf{x}) \int_{\Theta} p(\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}) L(\mathbf{x}, \boldsymbol{\omega}_i) d\boldsymbol{\omega}_i$$

$$L_i(\mathbf{x}, \boldsymbol{\omega}) \approx \frac{4\pi\sigma_s(\mathbf{x})}{N} \sum^{N} p(\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}) L(\mathbf{x}, \boldsymbol{\omega}_i)$$

Shadow Rays

- When rendering shadows cast by volumes, we start with the standard shadow ray approach
- To determine the shadows on a surface, trace a ray to the light, first testing for any opaque surfaces
- If nothing is blocking the light, then test for volumes
- If the ray passes through a volume, the light will be attenuated by the extinction (absorption and out-scattering) of the volume, resulting in a [0…1] scale factor applied to the light
- If the volume is homogeneous, we can attenuate it by the total length of the ray through volume
- For an inhomogeneous volume, we must ray march through the volume, applying the local extinction to each segment of the ray
- Shadow rays like this are used both for computing direct lighting on surfaces as well as direct light in-scattering in volumes

Single/Multiple Scattering

- Often volumetric scattering is simplified by only considering *single scattering* as opposed to *multiple scattering*
- Single scattering refers to using only one scattering event per light path, and multiple scattering considers any number of scattering events
- Single scattering is often sufficient for thin haze and will capture the effect of light beams
- It is also often sufficient for thicker volumes with a high coefficient of absorption like smoke
- However, it is not sufficient for dense, low-absorption volumes like clouds, where multiple scattering dominates

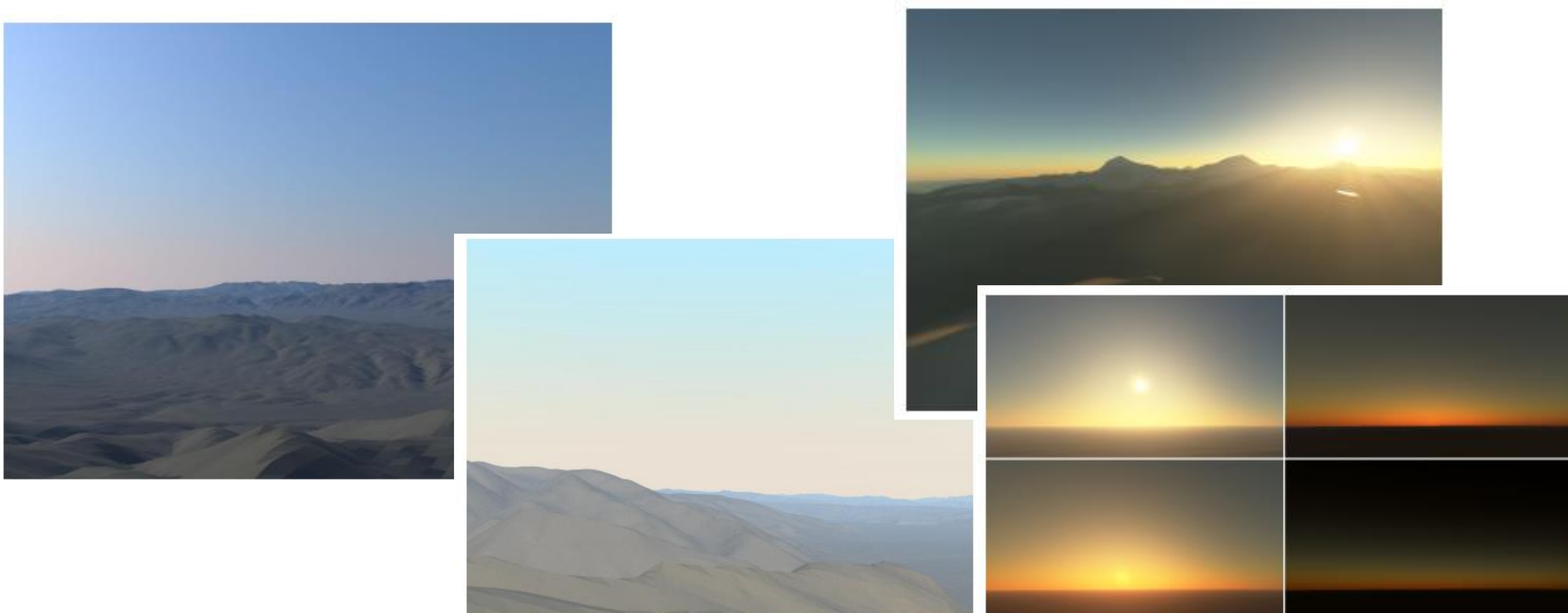
Volumetric Phenomena

Fire

- “Physically Based Modeling and Animation of Fire”, Nguyen, Fedkiw, Jensen, 2003

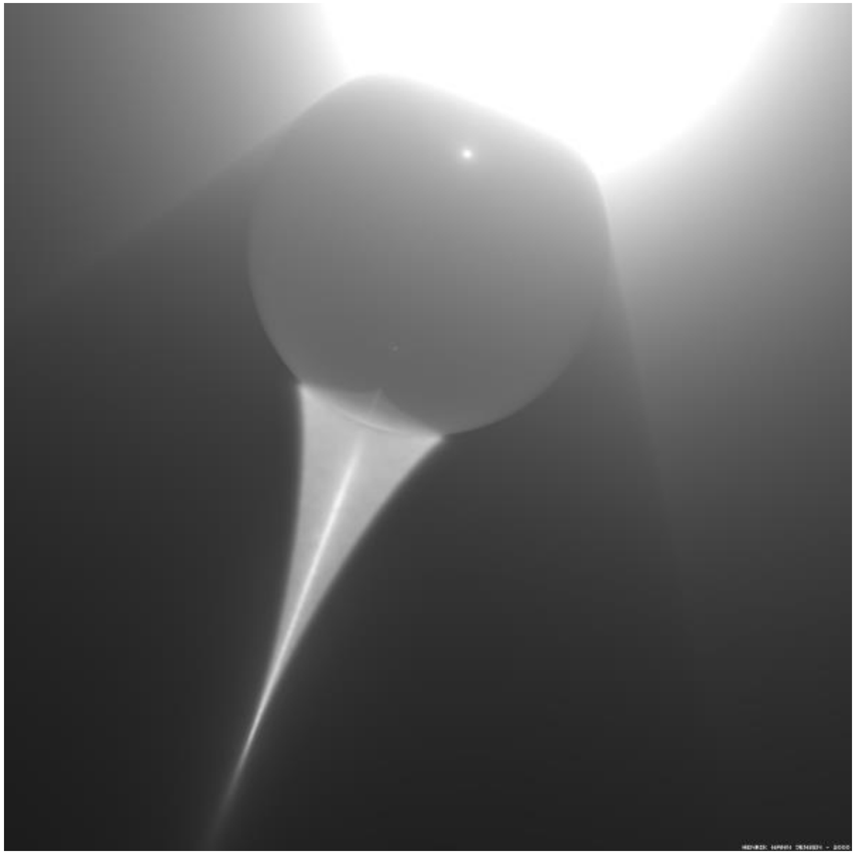
Sky Rendering

- “A Practical Analytical Model for Daylight”, Preetham, Shirley, Smits, 1999
- “A Physically Based Night Sky Model”, Jensen, Durand, Stark, Premoze, Dorsey, Shirley, 2001
- “Precomputed Atmospheric Scattering”, Bruneton, Neyret, 2008
- “An Analytic Model for Full Spectral Sky-Dome Radiance”, Hosek, Wilkie, 2012



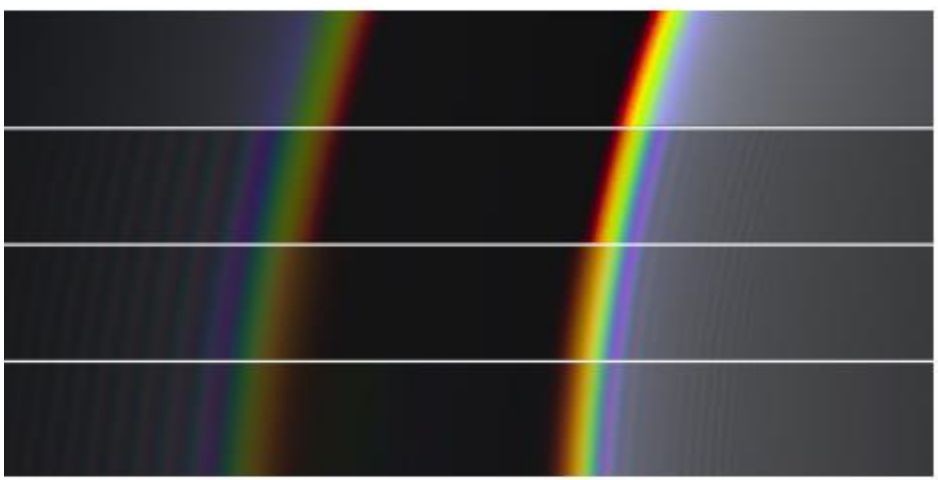
Volumetric Caustics

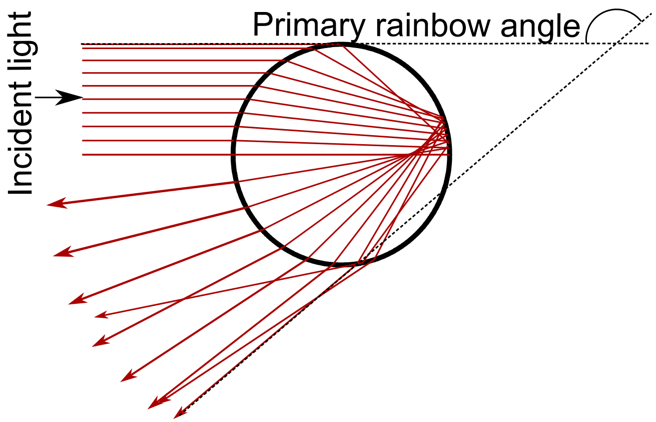
- "Efficient Simulation of Light Transport in Scenes with Participating Media using Photon Maps", Jensen, Christensen, 1998



Rainbows

- “Physically Based Simulation of Rainbows”, Sadeghi, Munoz, Laven, Jarosz, Seron, Gutierrez, Jensen, 2012

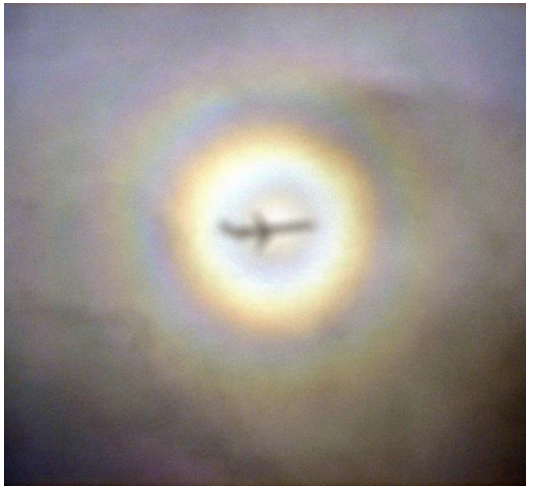




Atmospheric Phenomena

Corona

Ice Crystal Halo



Glory

Final Project

Final Project

- For the final project, you must add some features of your own choice to your renderer and render a final image
- Ideally, you could come up with a vision of what you want the final image to be and then implement the necessary features to achieve this, although you could also take the opposite approach and implement a bunch of features and then choose an idea that shows those off
- Also, it would be nice if it involved some different models than the standard test models
- This means you could search around online for some data and then write an appropriate importer, or even better- you could build something yourself in 3DStudio or Maya and then import the data to your renderer
- The project should demonstrate at least one ‘complex’ feature and a few ‘simpler’ features
- You can also generate more than one image to show off different features, or you cold render an animation
- Provide an HTML page that shows the image and has a description of the features
- Do a 5 minute live presentation during finals week

Final Project

- Some suggestions for ‘complex’ features:
  - Volumetric scattering
  - Procedural modeling
  - Displacement mapping
  - Dispersion
  - Photon mapping
  - Translucency
  - Procedural texture