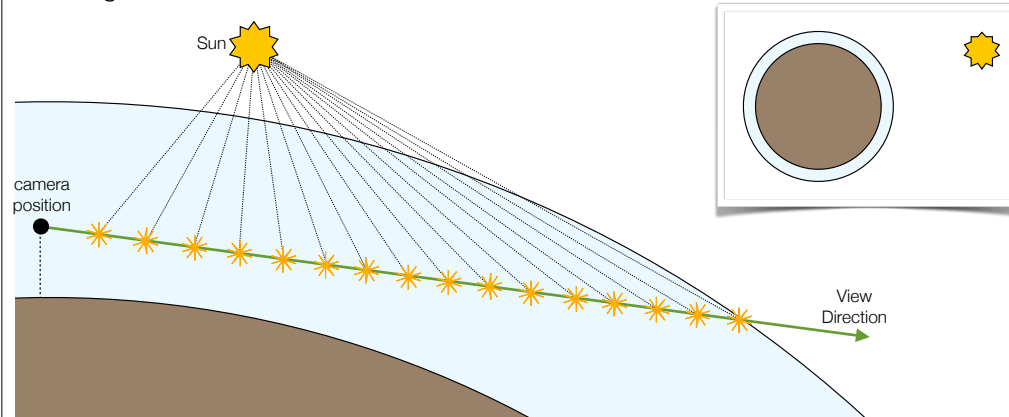


Material prepared from this chapter of GPU Gems 2:  
[http://http.developer.nvidia.com/GPUGems2/gpugems2\\_chapter16.html](http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter16.html)

# Atmospheric Model



Particles along the view ray reflect a **portion** (not 100% of it.. thus “scattering”) of the light toward the camera.



planet: earth radius + atmosphere radius

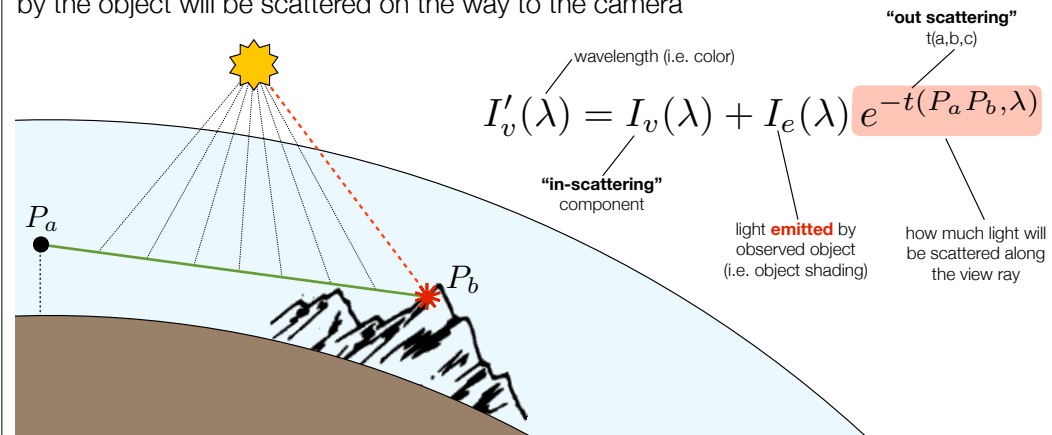
observer: height + view direction (angle)

sun: position (or direction assuming far-field approximation)

# Are you looking at an object?



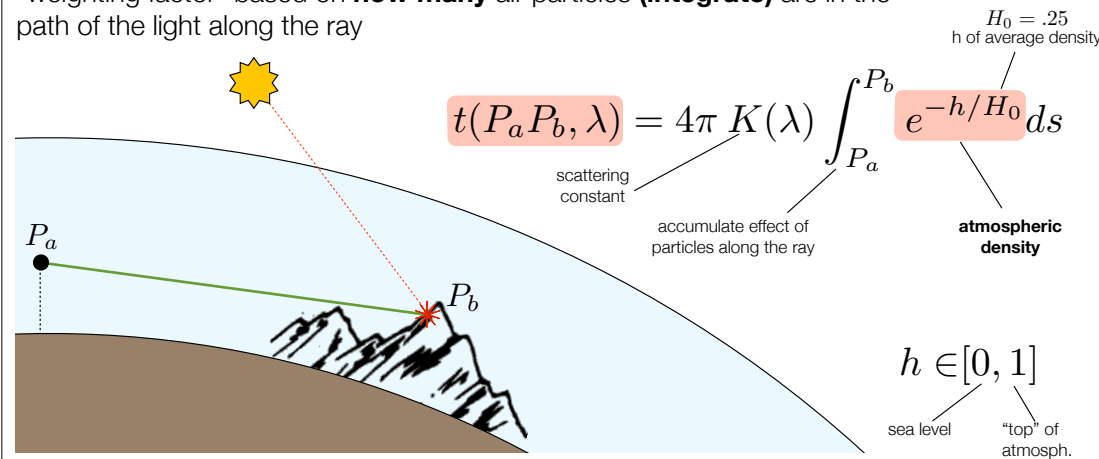
we must also take into account that some of the light reflected by the object will be scattered on the way to the camera



# “Out Scattering” Equation



“weighting factor” based on **how many** air particles (**integrate**) are in the path of the light along the ray

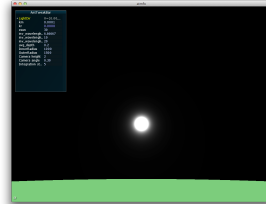


Essentially density of particles is assumed proportional to the height  
The scattering constant “K” is different for {Mie, Rayleigh} scattering

# Rayleigh v.s. Mie Scattering

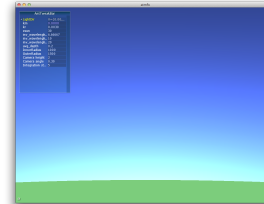


## Mie Scattering



Caused by aerosol particles (e.g. dust)  
Acts equally across wavelengths  
Effect: gray sky on rainy days  
Effect: sun halo on cloudy days

## Rayleigh Scattering



Scattering of (small) molecules in air  
Affects more short wavelengths  
Effect: blue daylight sky  
Effect: red sunset

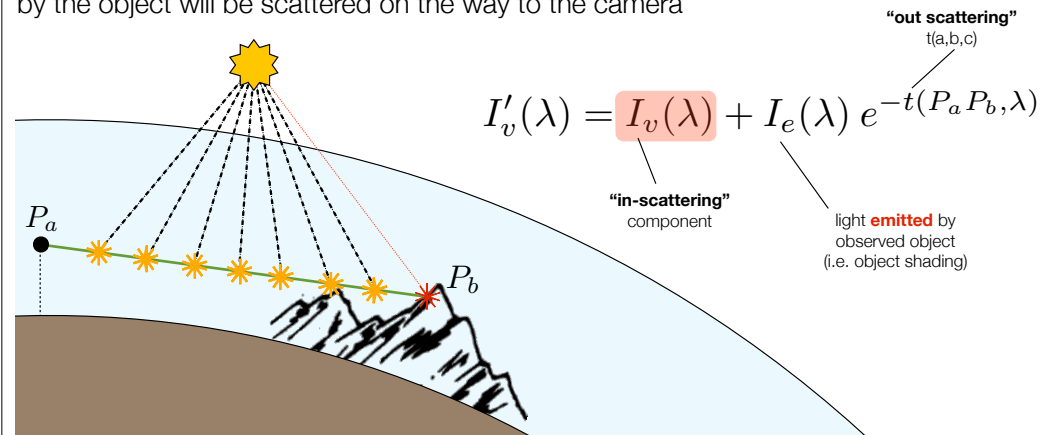
Mie: particles around size of wavelength

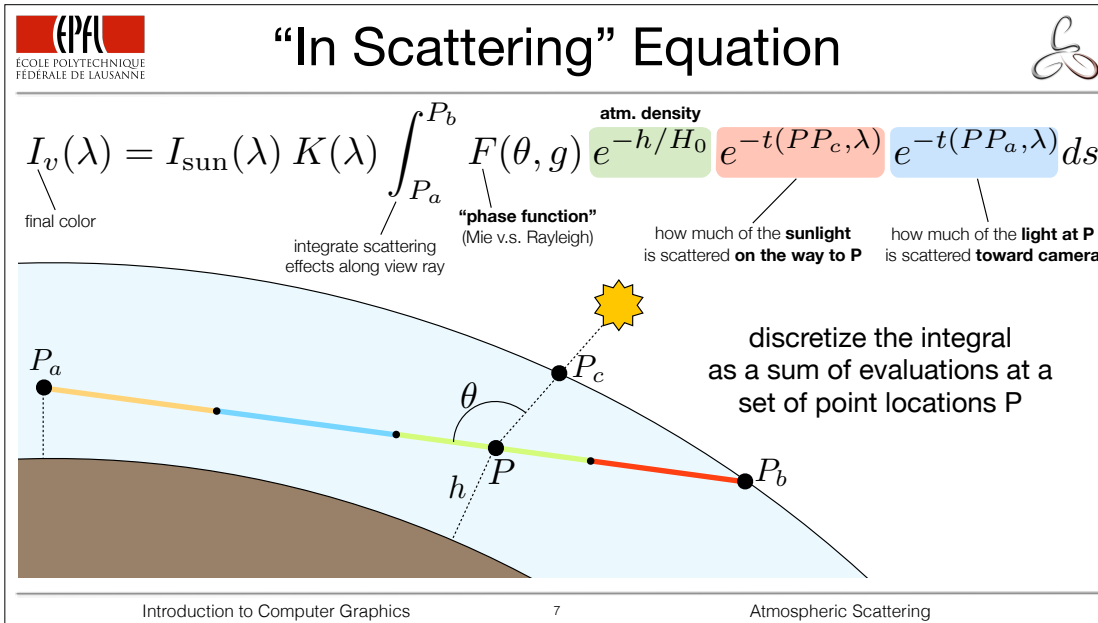
Rayleigh: particles much smaller than wavelength

Rayleigh (blue → green → red)

wiki: The reddening of sunlight is intensified when the sun is near the horizon, because the volume of air through which sunlight must pass is significantly greater than when the sun is high in the sky. The Rayleigh scattering effect is thus increased, **removing virtually all blue light** from the direct path to the observer. The remaining un-scattered light is mostly of a longer wavelength, and therefore appears to be orange.

we must also take into account that some of the light reflected by the object will be scattered on the way to the camera





The parameter "g" of the phase function changes for {Mie, Rayleigh} (see next slide)  
 You can think of "atm. density" here as the "shading" of air around P

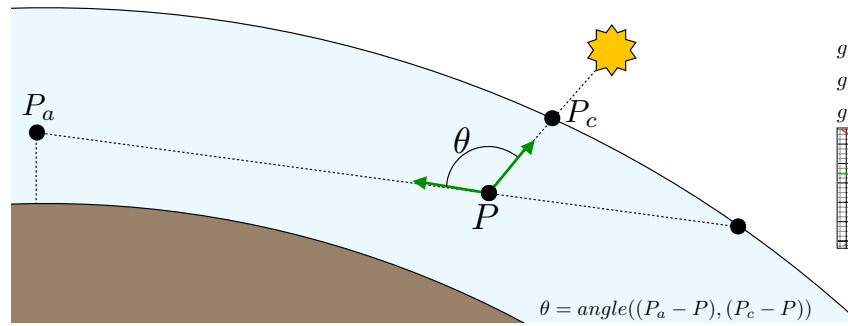
# The Phase Function “F”



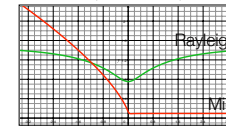
how much light is scattered in the direction of the camera (w.r.t. **angle**)

$$I_v(\lambda) = I_{\text{sun}}(\lambda) K(\lambda) \int_{P_a}^{P_b} F(\theta, g) e^{-h/H_0} e^{-t(P_c, \lambda)} e^{-t(P_a, \lambda)} ds$$

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




$g \neq \{-1, +1\}$   
 $g = [-.99, -.75]$  (Mie)  
 $g = 0$  (Rayleigh)









$$I_v(\lambda) = I_{\text{sun}}(\lambda) K(\lambda) \int_{P_a}^{P_b} F(\theta, g) e^{-h/H_0} e^{-t(PP_c, \lambda)} e^{-t(PP_a, \lambda)} ds$$

parallel sunlight

$$I_v(\lambda) = K(\lambda) F(\theta, g) e^{-4\pi K(\lambda)} \int_{P_a}^{P_b} e^{-h/H_0} T_1(h, \alpha) e^{-t(PP_a, \lambda)} ds$$

precompute and store in a lookup table

Parallel Sunlight Assumption [Nishita et al. '93]

The evaluation of the outer integral is “ok” (...necessary).  
But can we do something about the inner integral?  
Otherwise the cost is 300 computations per-fragment  
(and that just discretizing integral with 5 samples!).

**Parallel sun rays:** “This makes it possible to calculate a lookup table that contains the amount of out-scattering for rays going from the sun to any point in the atmosphere. This table replaces one of the out-scattering integrals with a lookup table whose variables are altitude and angle to the sun. Because the rays to the camera are not parallel, the out-scattering integral for camera rays still had to be solved at runtime.”

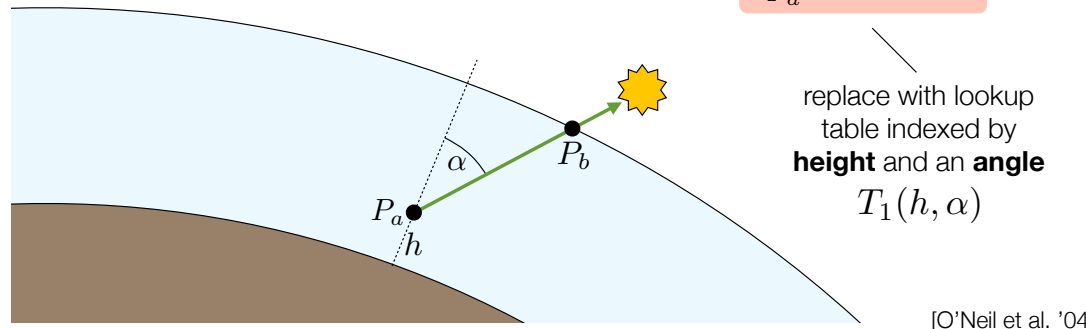
# Using a Lookup Table



likely the reasons for  
which this was named "t"  
(as in lookup-table)

$$t(P_a P_b, \lambda) = 4\pi K(\lambda)$$

$$\int_{P_a}^{P_b} e^{-h/H_0} ds$$



replace with lookup  
table indexed by  
**height** and an **angle**  
 $T_1(h, \alpha)$

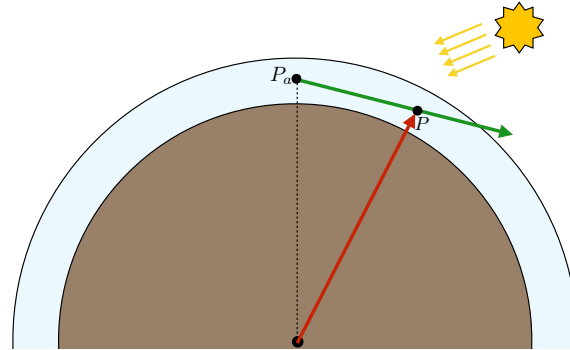
[O'Neil et al. '04]

Read VERY carefully the 2nd part of Section 16.3.  
You have to make a few important tweaks to use the lookup table

# Girod Scattering Equations :)



$$I_v(\lambda) = I_{\text{sun}}(\lambda) K(\lambda) \int_{P_a}^{P_b} F(\theta, g) e^{-h/H_0} \underbrace{e^{-t(P P_c, \lambda)}}_{T_{PP_c}} \underbrace{e^{-t(P P_a, \lambda)}}_{T_{PP_a}} ds$$



$$T_{PP_c} = H_0 e^{-\text{angle}(\text{red arrow}, \text{yellow arrow})^3}$$

$$T_{PP_a} = H_0 e^{-\text{angle}(\text{red arrow}, \text{green arrow})^3}$$

(technically they are incorrect... use for debug!)  
see correct approx. on [GPU Gems 2, Ch.16]

Read VERY carefully section 16.4.1 of GPU Gems

Disclaimer: check math as sun direction vector might be flipped!

