Rendering Outdoor Light Scattering in Real Time

Naty Hoffman
Westwood Studios
naty@westwood.com

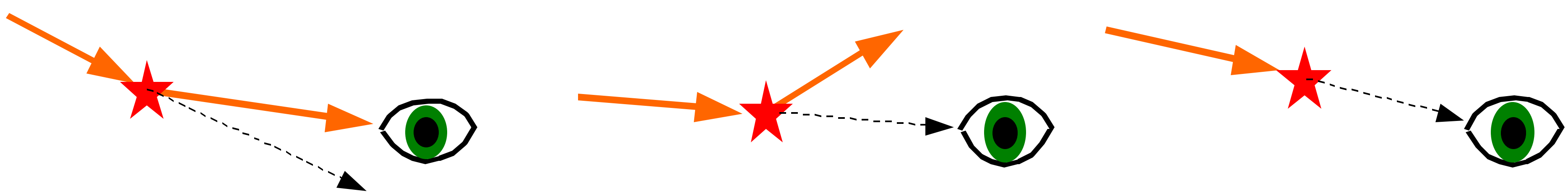
Arcot J Preetham
ATI Research
preetham@ati.com

Outline

- Basics
  - Atmospheric Light Scattering
  - Radiometric Quantities
  - From Radiance to Pixels
- Scattering Theory
  - Absorption, Out-Scattering, In-Scattering
  - Rayleigh and Mie Scattering
- Implementation
  - Aerial Perspective, Sunlight, Skylight
  - Vertex Shader
- Future Work

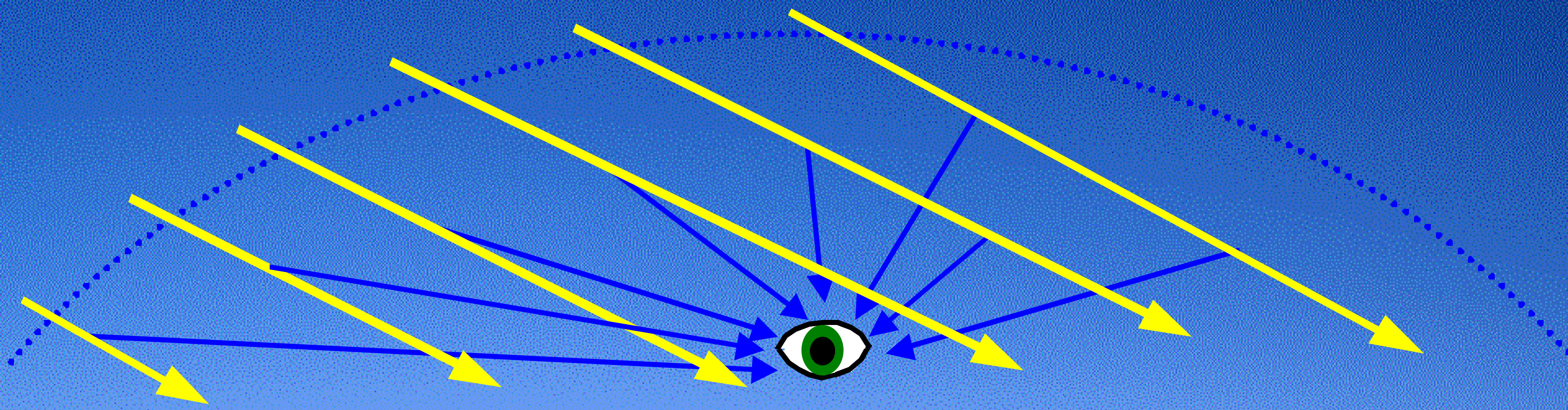
Atmospheric Light Scattering

- Is caused by a variety of particles
  - Molecules, dust, water vapor, etc.
- These can cause light to be:
  - Scattered into the line of sight (in-scattering)
  - Scattered out of the line of sight (out-scattering)
  - Absorbed altogether (absorption)



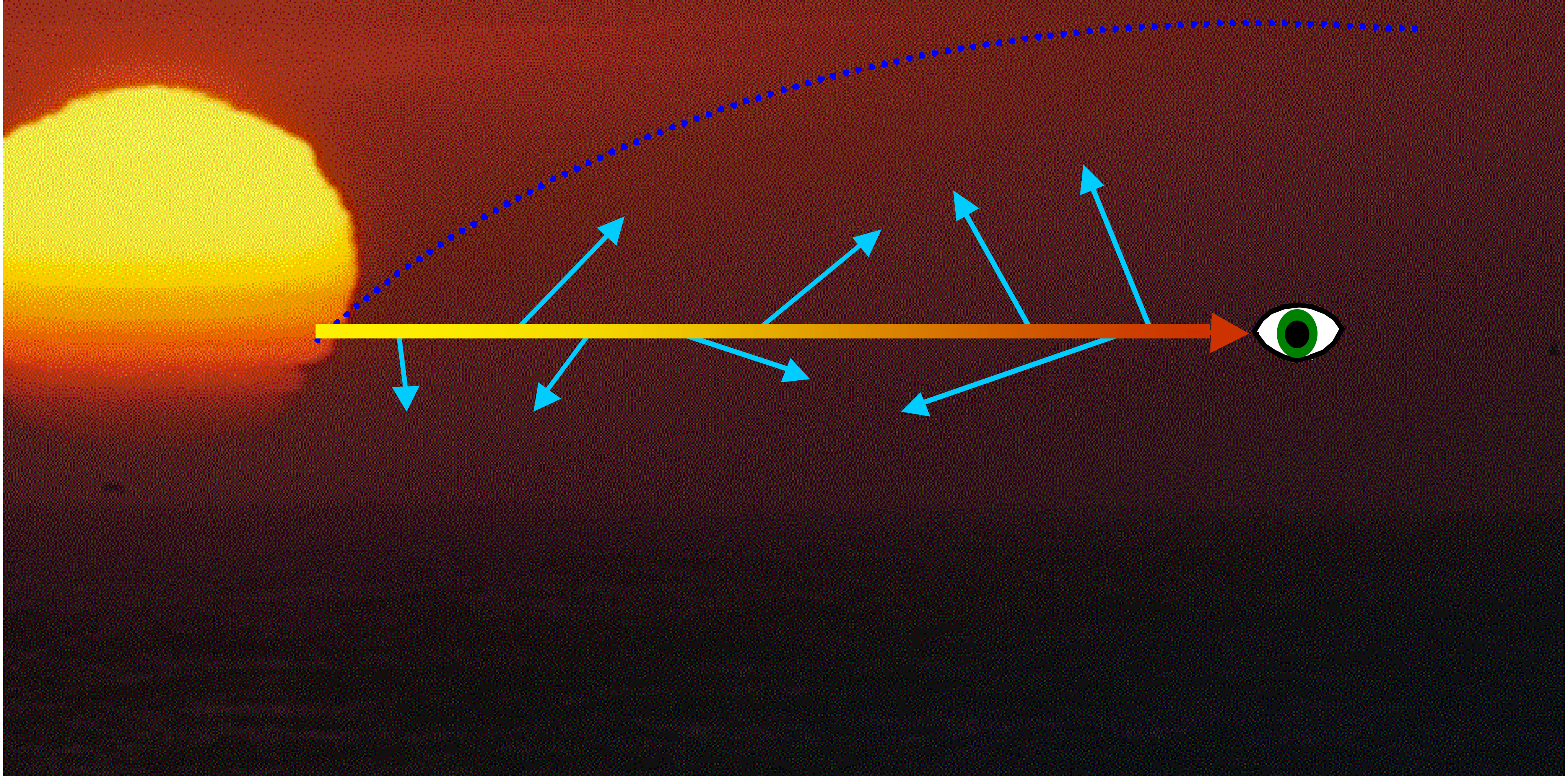
Atmospheric Light Scattering

- Illuminates the sky



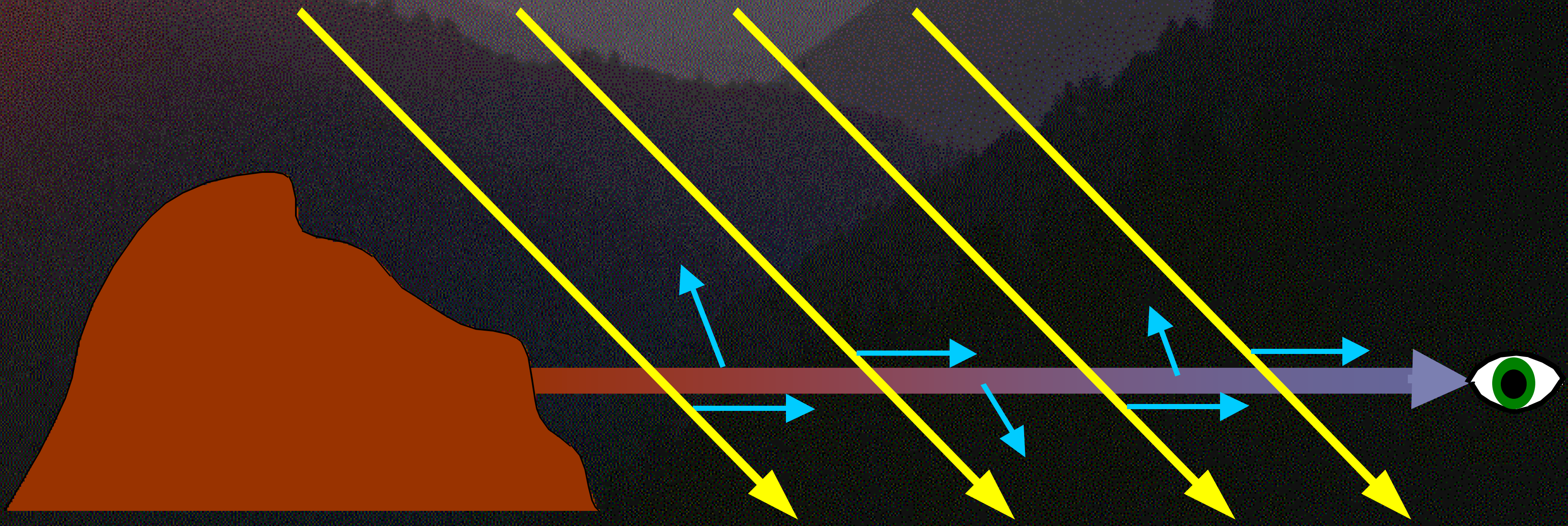
Atmospheric Light Scattering

- Attenuates and colors the Sun



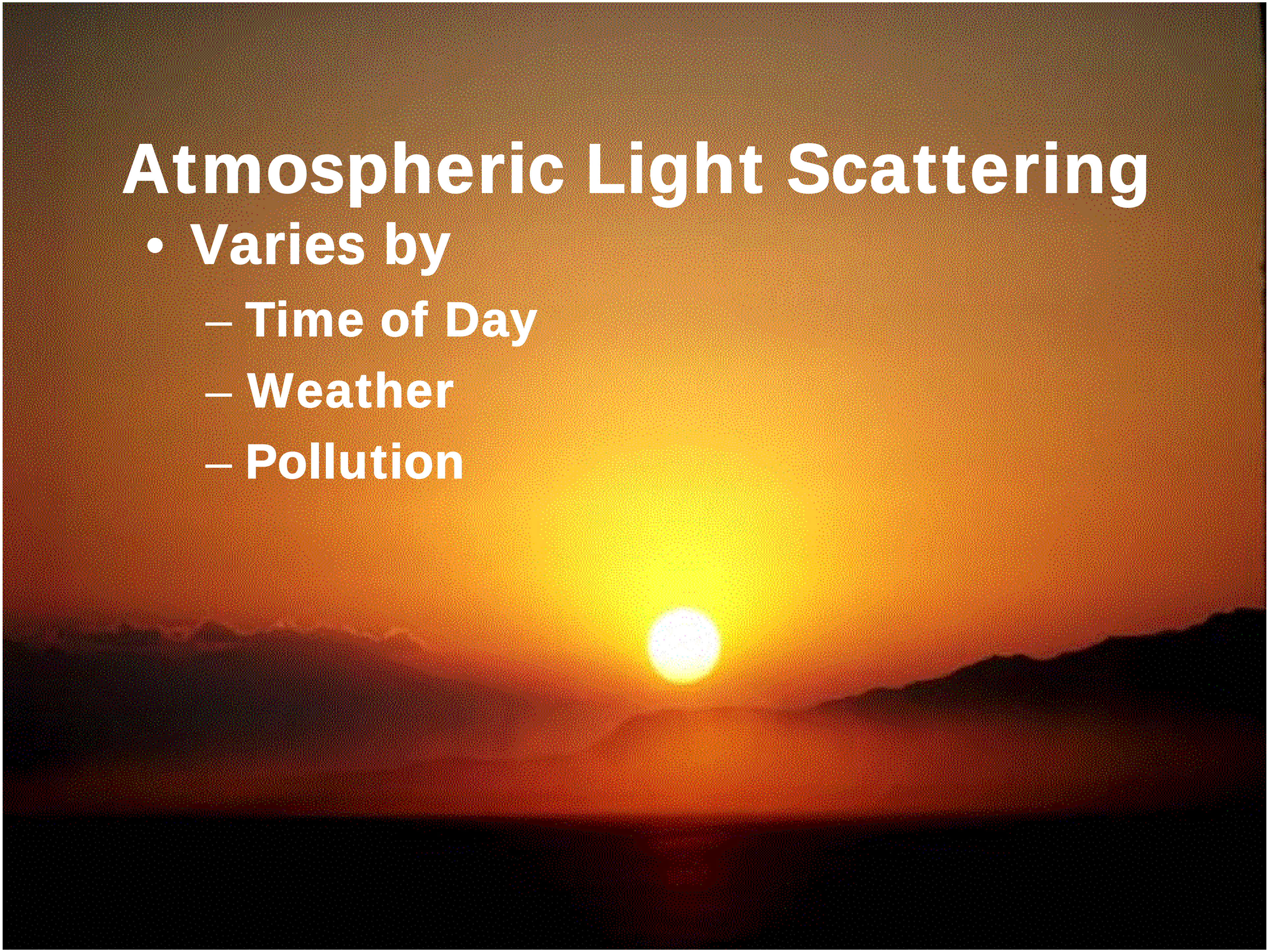
Atmospheric Light Scattering

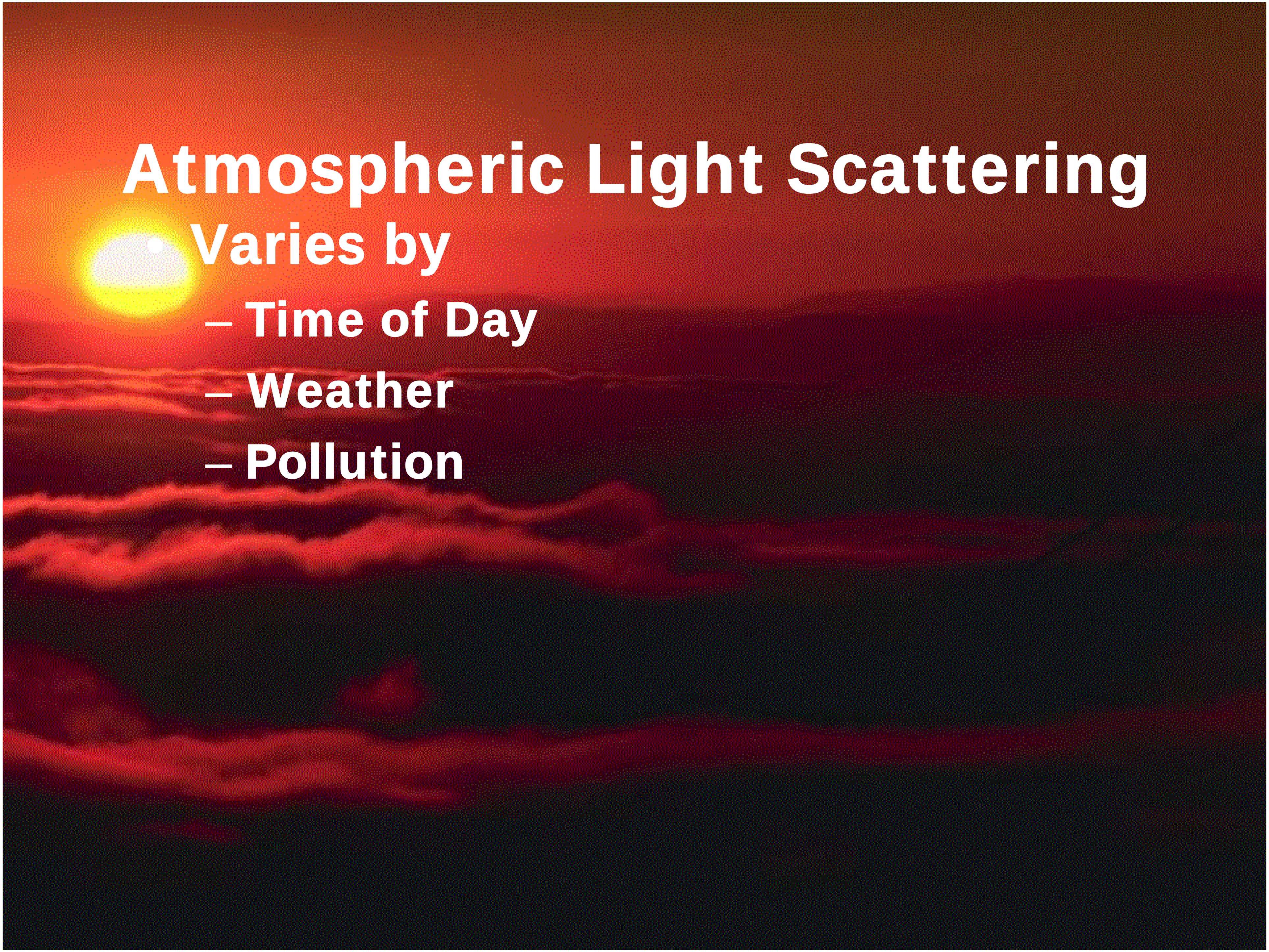
- Attenuates and colors distant objects



Atmospheric Light Scattering

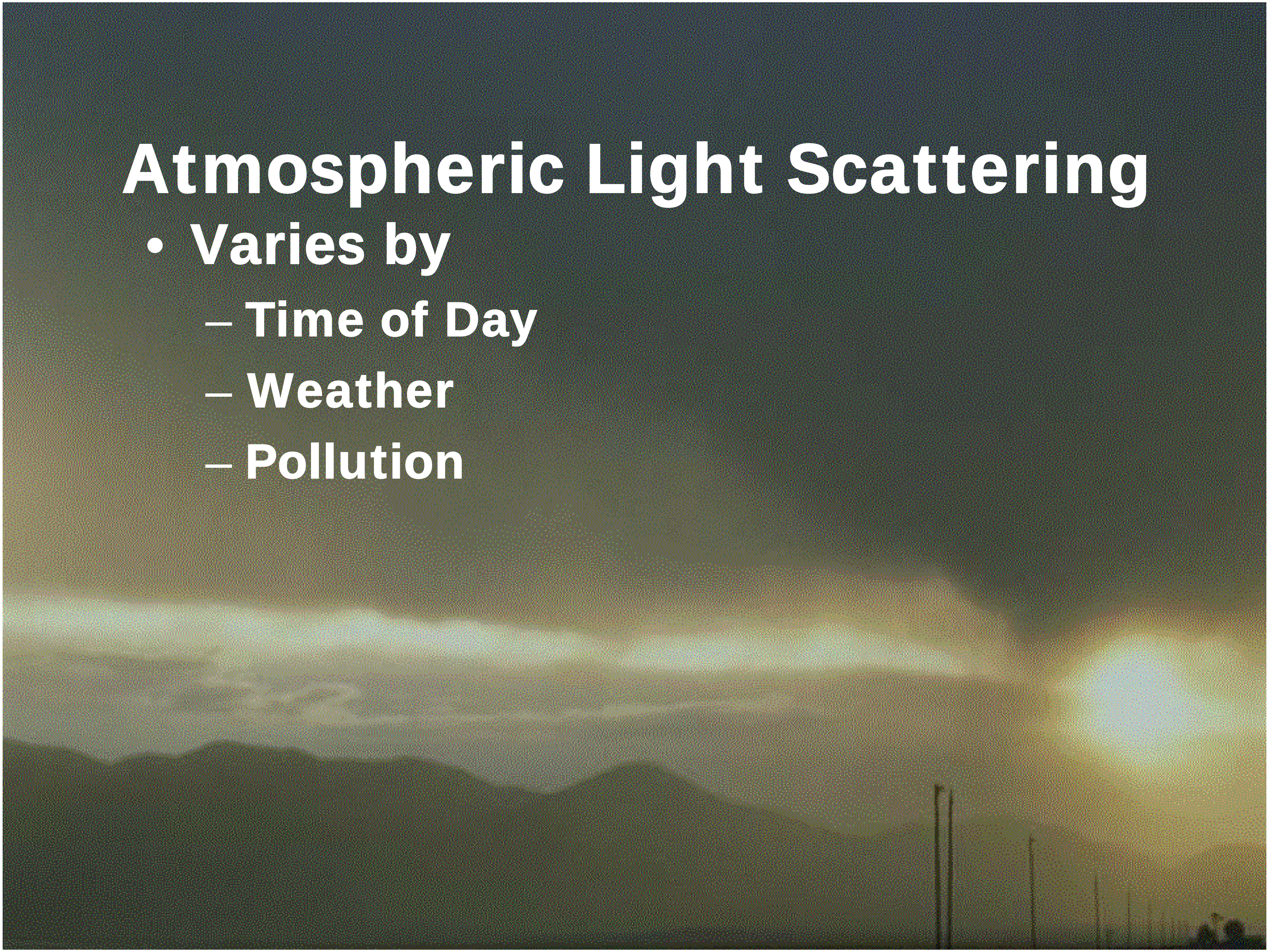
- Varies by
  - Time of Day
  - Weather
  - Pollution


Atmospheric Light Scattering
• Varies by
– Time of Day
– Weather
– Pollution



Atmospheric Light Scattering

- Varies by
  - Time of Day
  - Weather
  - Pollution

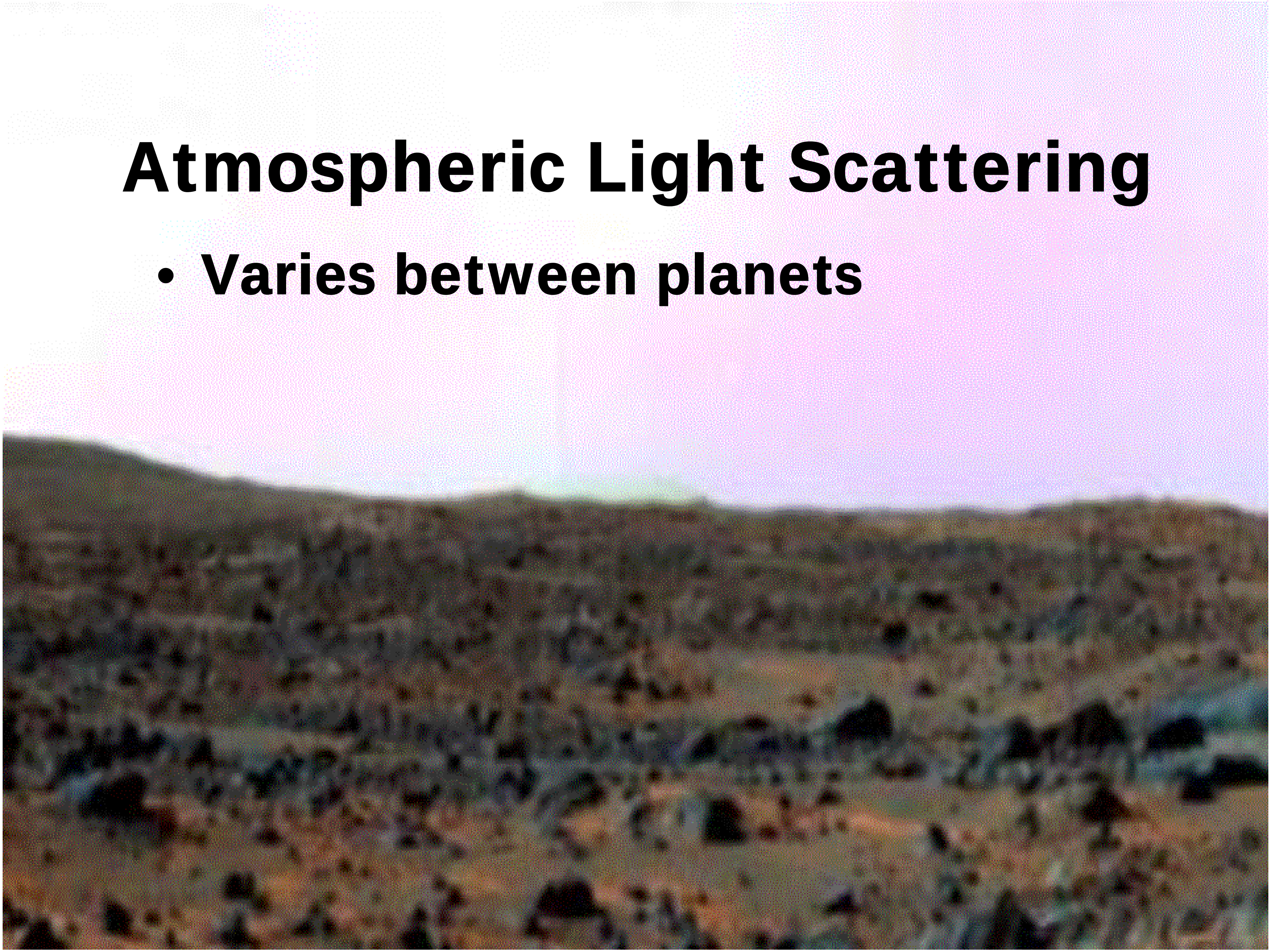


Atmospheric Light Scattering

- Varies by
  - Time of Day
  - Weather
  - Pollution

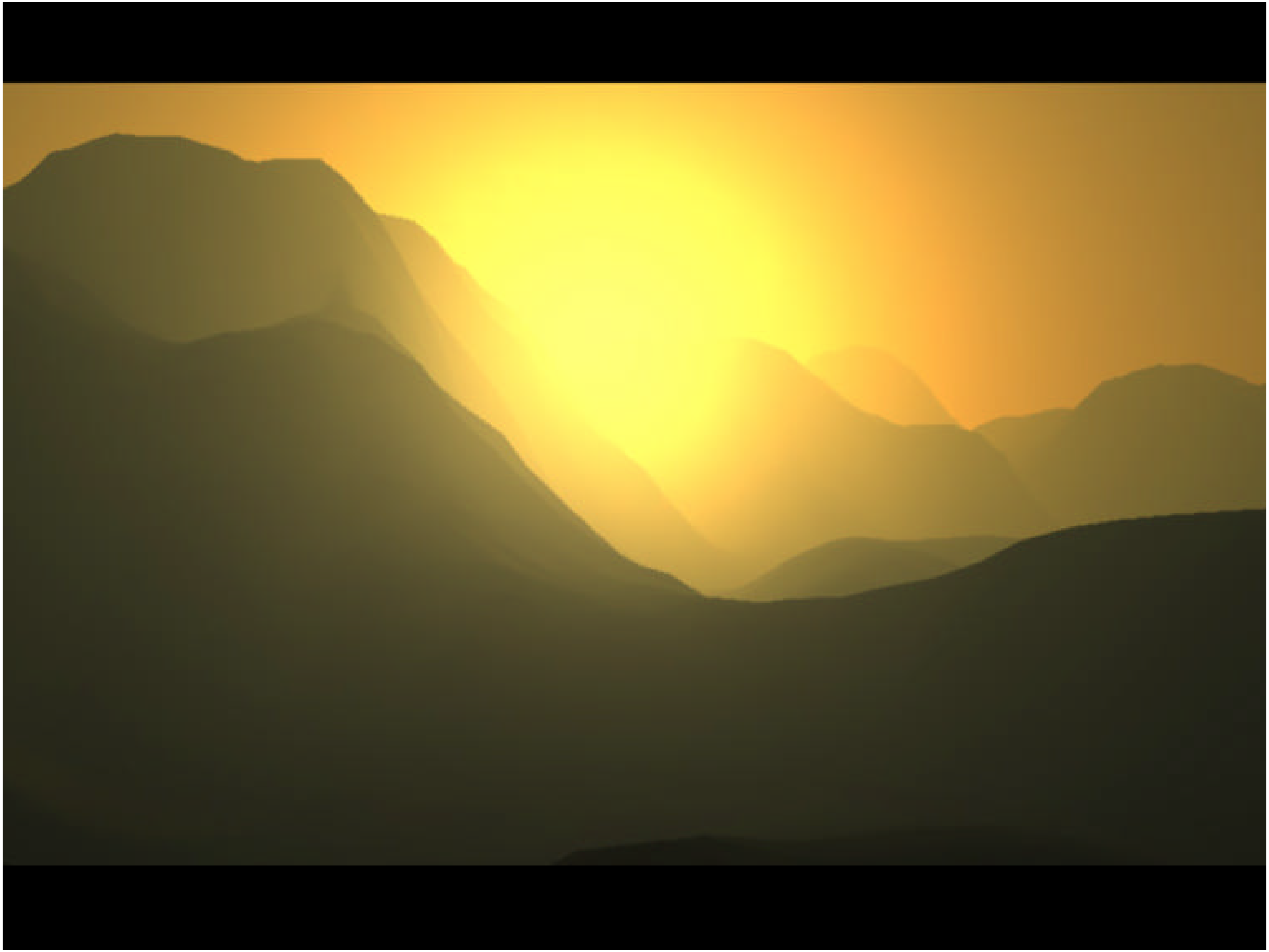
Atmospheric Light Scattering

- Varies between planets



Atmospheric Light Scattering

- Extinction (Absorption, Out-scattering)
  - Phenomena which remove light
  - Multiplicative: $L_{\text{extinction}} = F_{\text{ex}} L_0$
- In-scattering:
  - Phenomenon which adds light
  - Additive: L_{in}
- Combined: $L_{\text{scattering}} = F_{\text{ex}} L_0 + L_{\text{in}}$

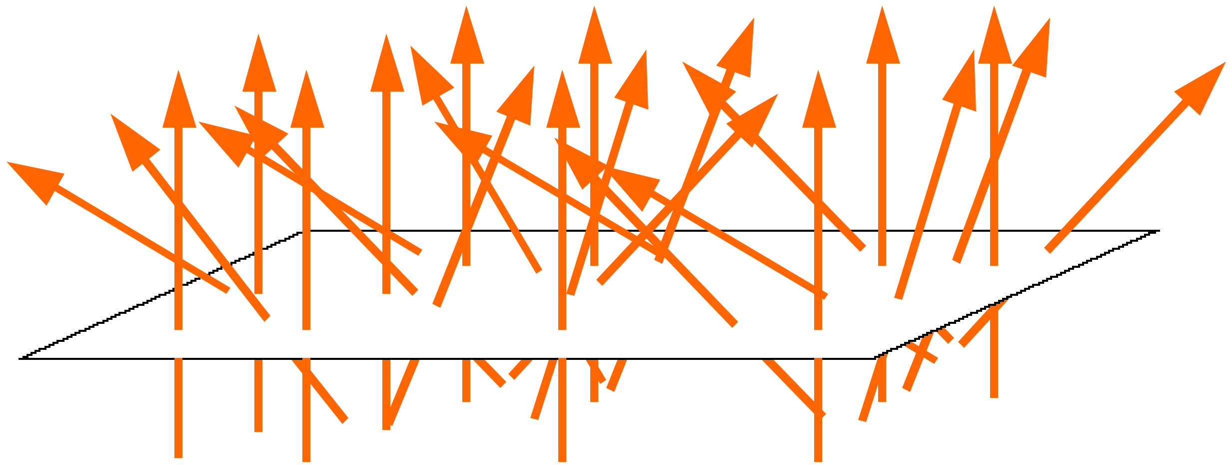


Radiometric Quantities

- Radiant Flux
- Radiance
- Irradiance

Radiometric Quantities

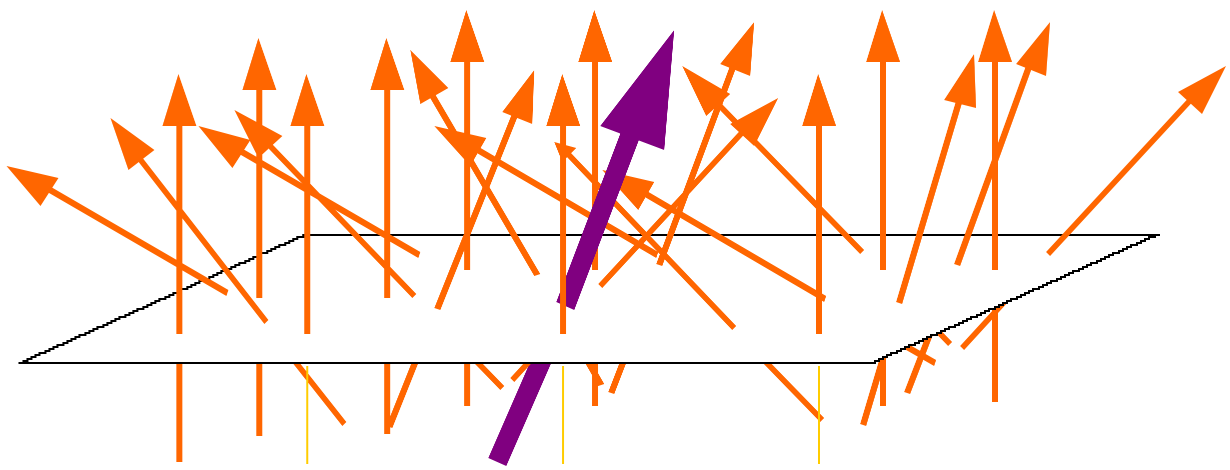
- **Radiant Flux Φ**
  - Quantity of light through a surface
  - Radiant power (energy / time)
  - Watt



make
better
games

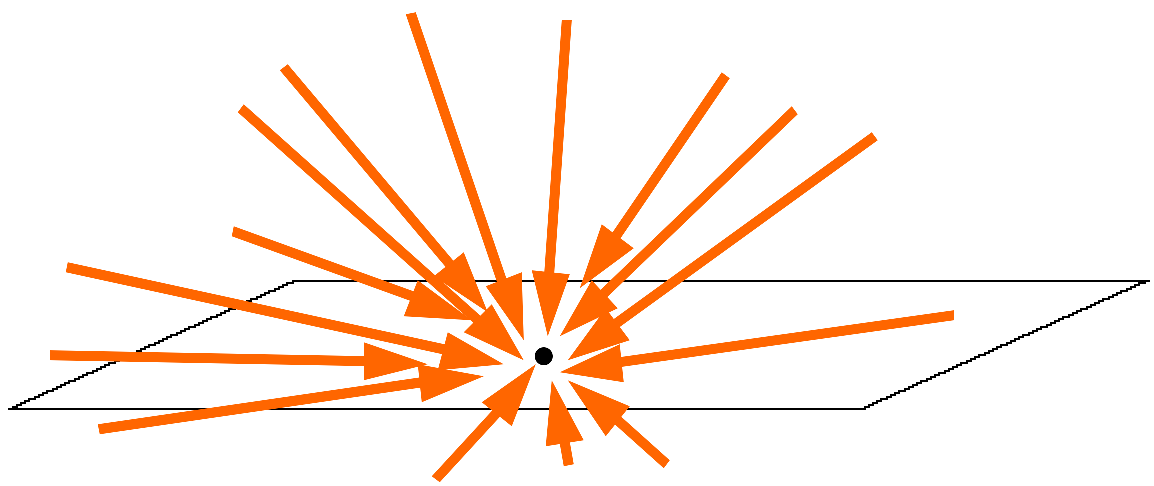
Radiometric Quantities

- **Radiance L**
  - **Quantity of light in a single ray**
  - **Radiant flux / area / solid angle**
  - **Watt / (meter^2 * steradian)**



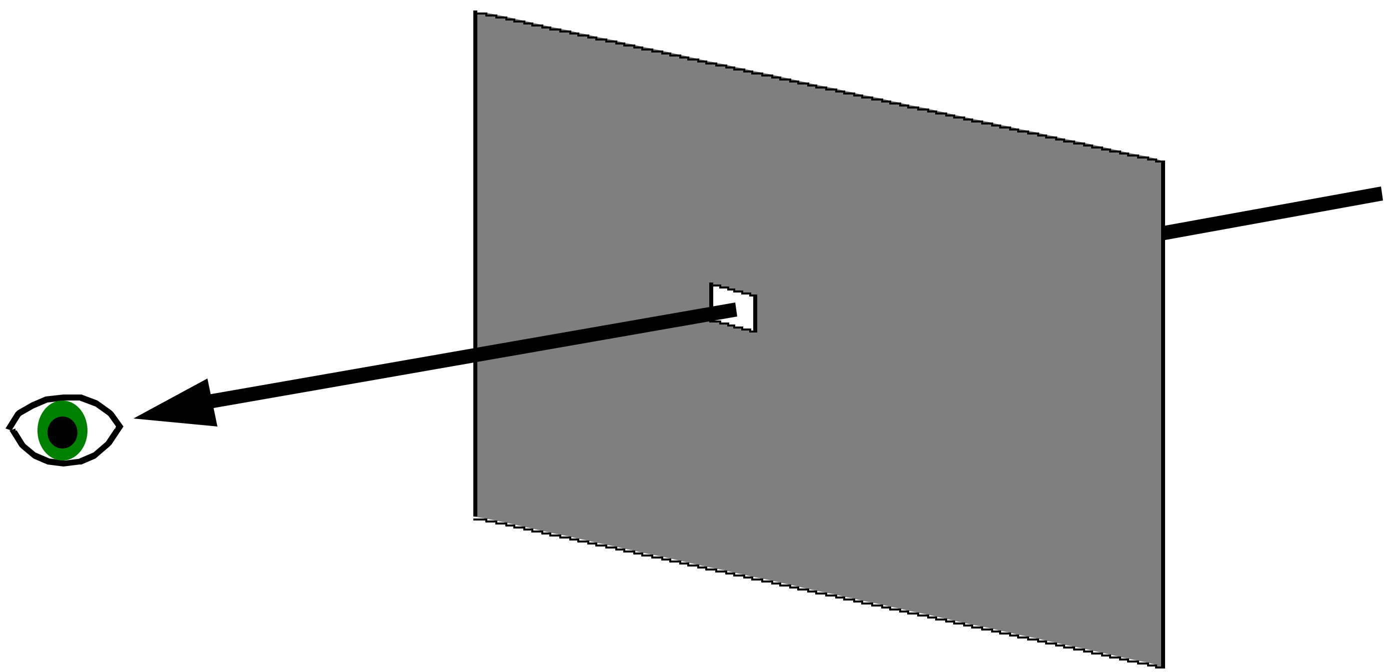
Radiometric Quantities

- **Irradiance** E
  - **Quantity of light incident to a surface point**
  - **Incident radiant flux / area (Watt / meter^2)**
  - **Radiance integrated over hemisphere**



From Radiance to Pixels

- Compute radiance incident to eye through each screen pixel

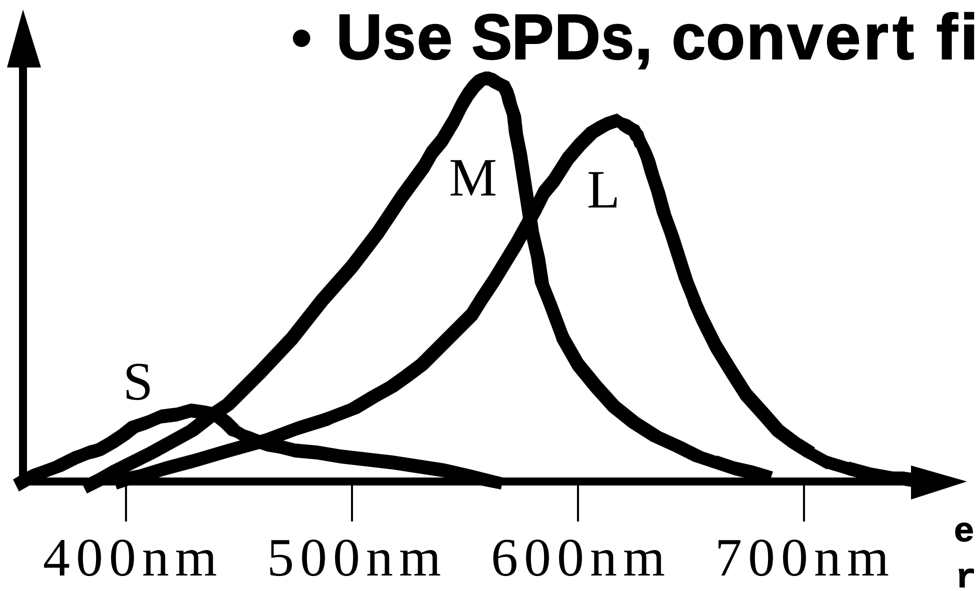


From Radiance to Pixels

- Pixel value based on radiance
- But radiance is distributed continuously along the spectrum
  - We need three numbers: R, G, B

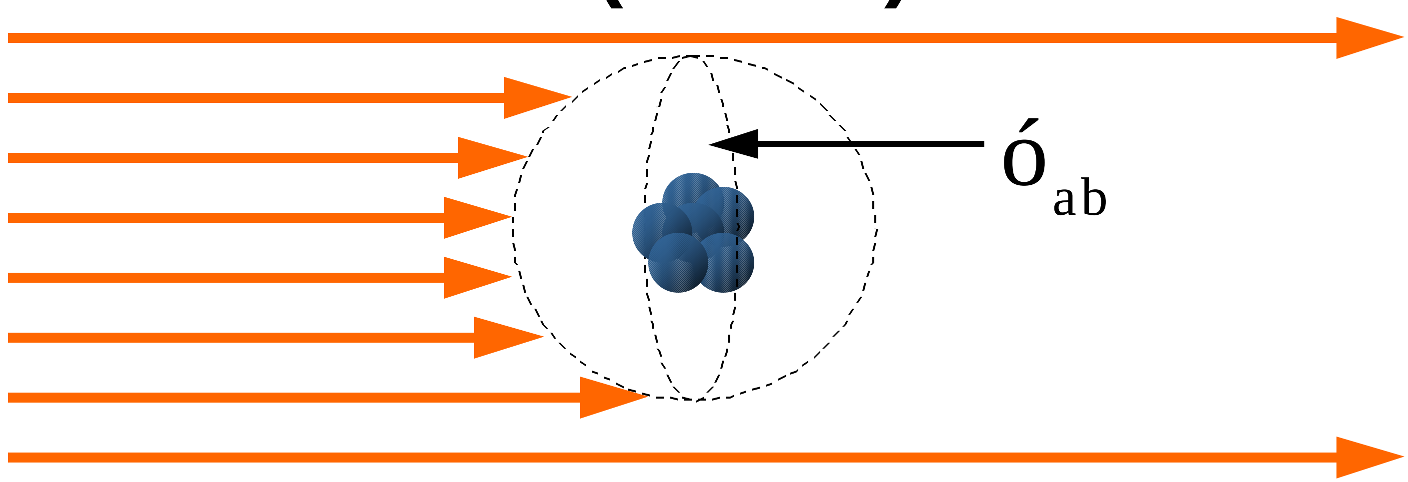
From Radiance to Pixels

- **SPD (Spectral Power Distribution) to RGB**
  - **Fast approach:**
    - **Do all math at R, G, B sample wavelengths**
  - **Correct approach:**
    - **Use SPDs, convert final radiance to RGB**



Absorption

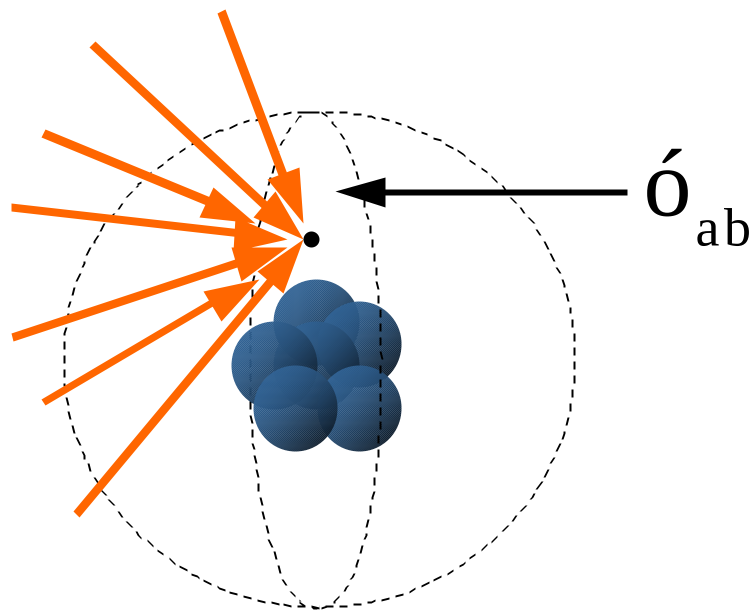
- Absorption cross section σ_{ab}
  - Absorbed radiant flux per unit incident irradiance Φ / E
  - Units of area ($meter^2$)



Absorption

- **Absorption cross section** $\acute{o}_{ab}$

$$\Phi = E * \acute{o}_{ab} \Longrightarrow \acute{o}_{ab} = \Phi / E$$



Absorption

- **Absorption coefficient** $\hat{a}_{\mathrm{ab}}$
  - **Particle density** $\tilde{n}_{\mathrm{ab}}$ **times absorption cross section** $\acute{o}_{\mathrm{ab}}$
  - **Units of inverse length (meter^{-1})**

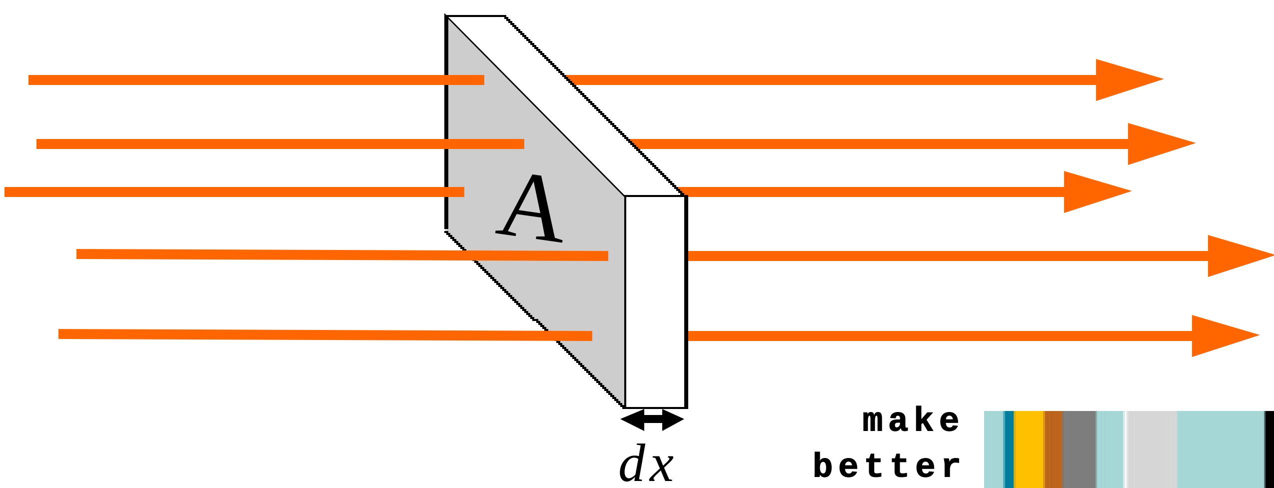
Absorption

- **Total absorption cross section:**

$$A_{ab} = ó_{ab}\ ñ_{ab}\ A\ dx$$

- **Probability of absorption:**

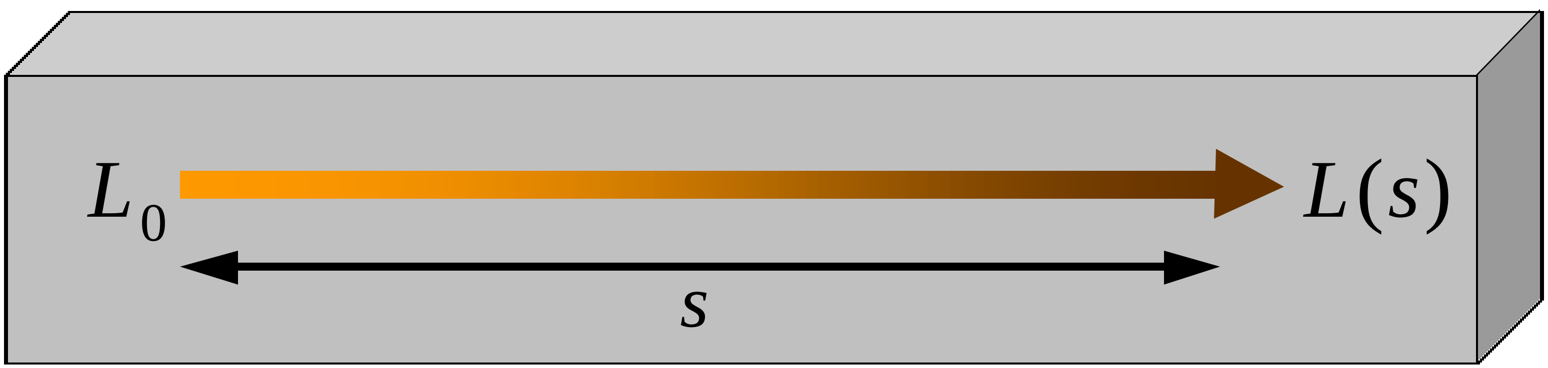
$$P_{ab} = A_{ab} / A = ó_{ab}\ ñ_{ab}\ dx = \hat{a}_{ab}\ dx$$



Absorption

- **Attenuation of radiance from travel through a <u>constant-density</u> absorptive medium:**

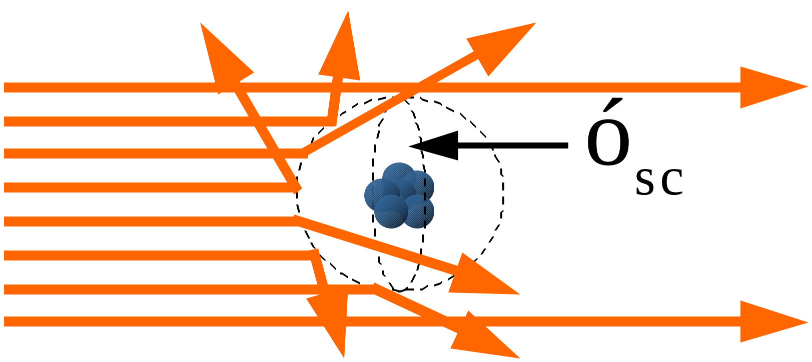
$$L(s) = L_0 e^{-b_{ab} s}$$



Out-Scattering

- **Exactly as in the absorption case**
  - **Scattering cross section** $\acute{o}_{sc}$
  - **Scattering coefficient** $\hat{a}_{sc} = \tilde{n}_{sc}\,\acute{o}_{sc}$
  - **Attenuation due to out-scattering in a constant-density medium:**

$$L(s) = L_0 e^{-b_{sc} s}$$



Extinction

- Both absorption and out-scattering attenuate light
- They can be combined as extinction
- Extinction coefficient $b_{\mathrm{ex}} = b_{\mathrm{ab}} + b_{\mathrm{sc}}$
- Total attenuation from extinction

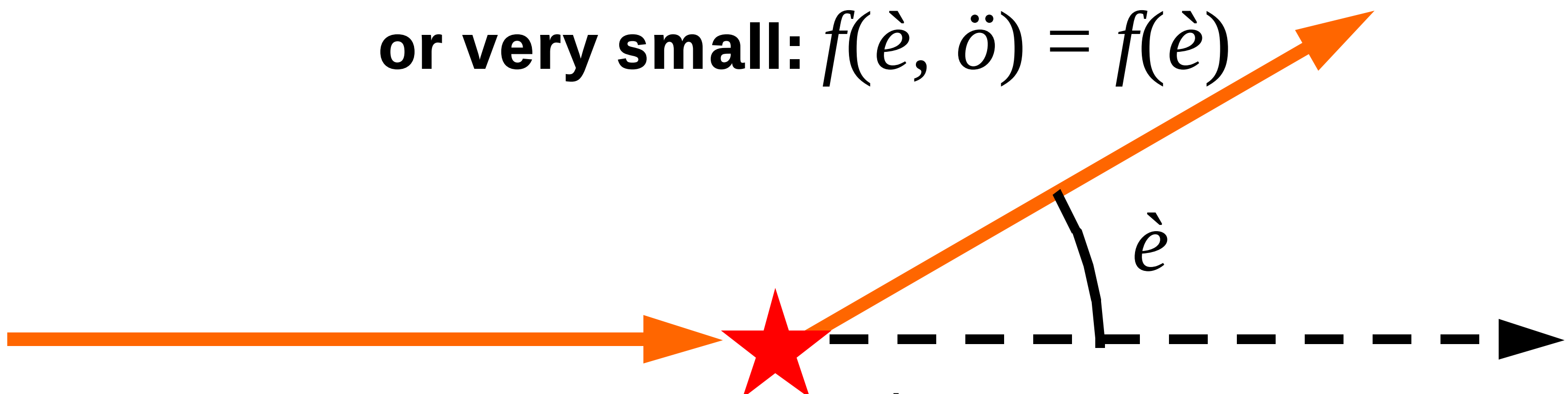
$$L(s) = L_0 e^{-b_{\mathrm{ex}} s} \Longrightarrow F_{\mathrm{ex}}(s) = e^{-b_{\mathrm{ex}} s}$$

In-Scattering

- Light is scattered into a view ray from all directions
  - From the sun
  - From the sky
  - From the ground
- We will only handle in-scattering from the sun

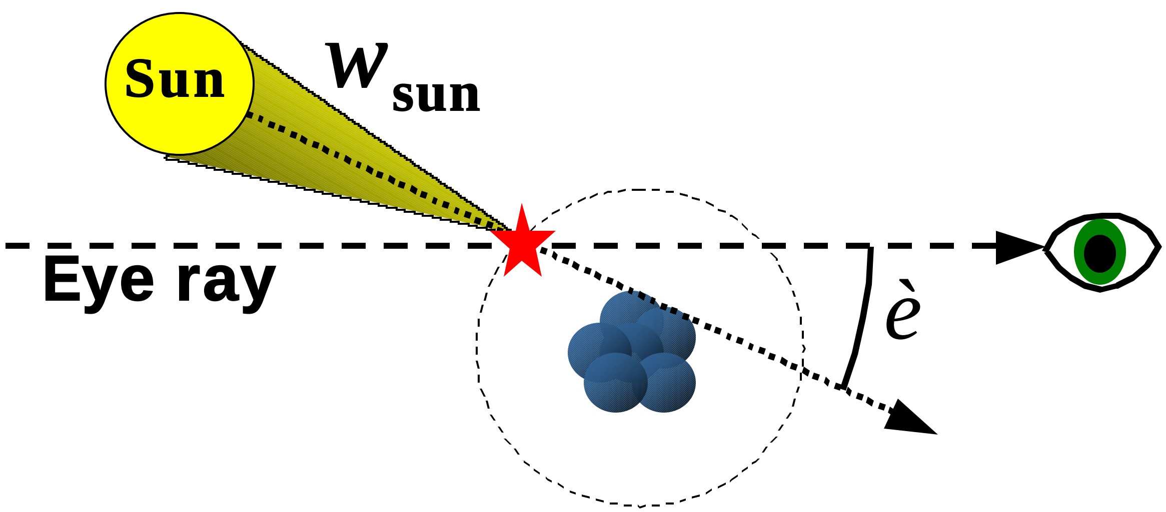
In-Scattering

- **Where does a scattered photon go?**
  - **Scattering phase function** *f*(è, ö)
    - **If a photon is scattered, gives the probability it goes in direction** è, ö
    - **Most atmospheric particles are spherical or very small:** *f*(è, ö) = *f*(è)



In-Scattering

- How do we use $f(\grave{e})$ for in-scattering?
  - In-scatter probability: $f(\grave{e})w_{\mathrm{sun}}$
  - In-scatter radiance : $f(\grave{e})w_{\mathrm{sun}}L_{\mathrm{sun}}=f(\grave{e})E_{\mathrm{sun}}$



In-Scattering

- **In-scattering over a path**
  - **Radiance from a single event:** $f(\grave{e})E_{\mathrm{sun}}$
  - **Over a distance** dx**:** $f(\grave{e})E_{\mathrm{sun}}\,\hat{a}_{\mathrm{sc}}\,dx$
- **Angular scattering coefficient**

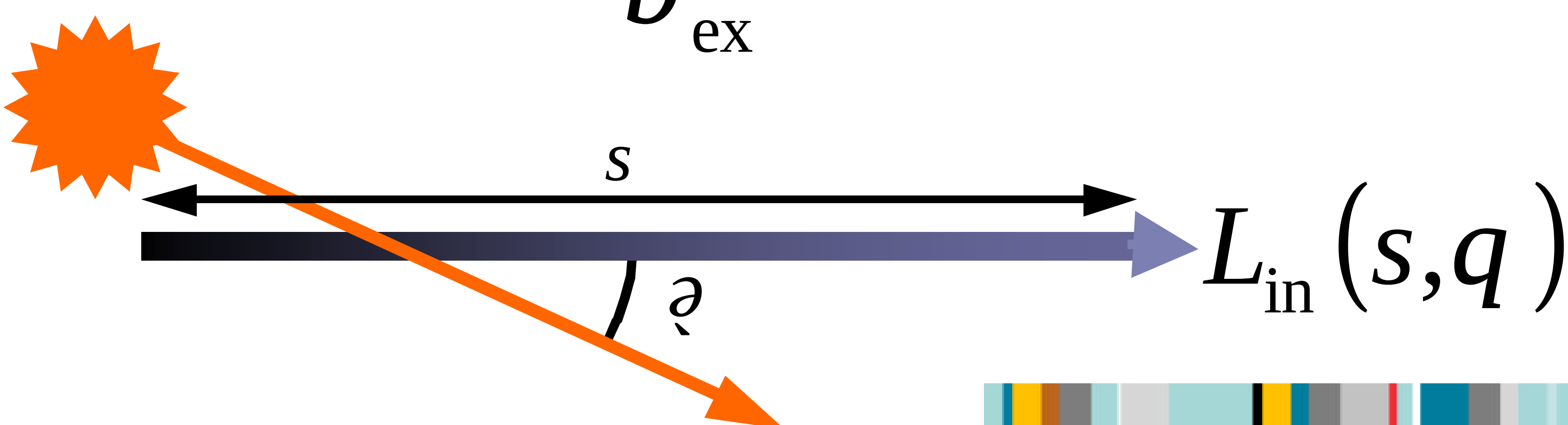
$$b_{\mathrm{sc}}(q) = b_{\mathrm{sc}}\,f(q)$$

  - **In-scattering over** dx**:** $E_{\mathrm{sun}}\,\hat{a}_{\mathrm{sc}}(\grave{e})\,dx$
  - **Units of** $\hat{a}_{\mathrm{sc}}(\grave{e})$ **: meter^{-1} * steradian^{-1}**

In-Scattering

- **Added radiance from solar in-scattering through a constant-density scattering medium:**

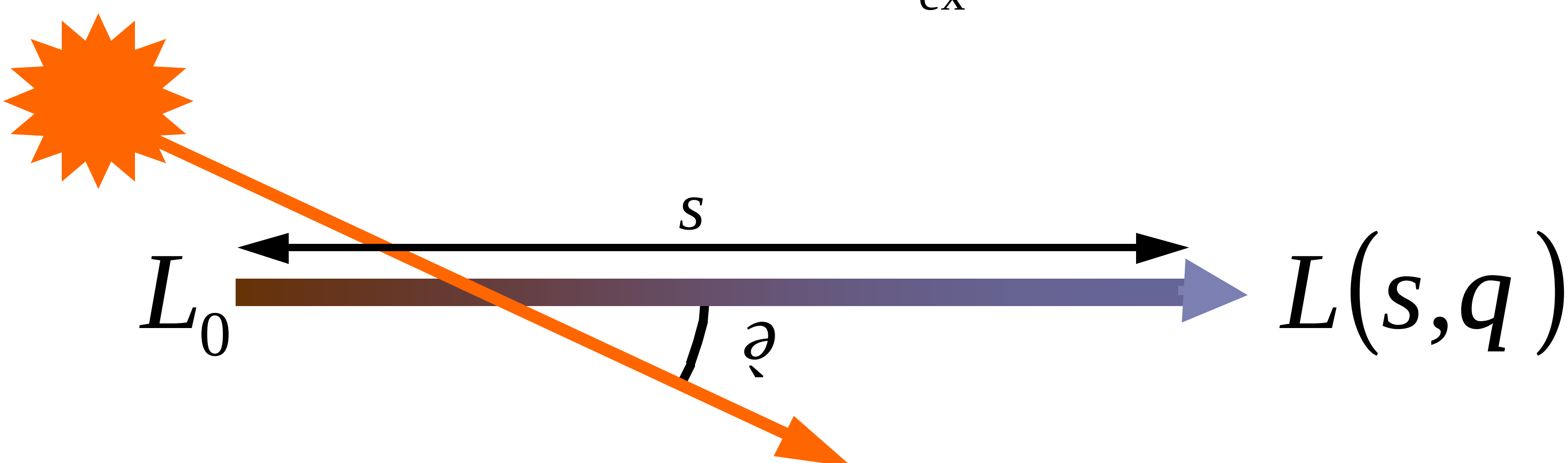
$$L_{\text{in}}(s,q) = \frac{1}{b_{\text{ex}}} E_{\text{sun}}\, b_{\text{sc}}(q)\left(1 - e^{-b_{\text{ex}} s}\right)$$



Extinction and In-Scattering

$$L(s,q) = L_0 F_{ex}(s) + L_{in}(s,q)$$

$$F_{ex}(s) = e^{-b_{ex} s} \qquad L_{in}(s,q) = \frac{1}{b_{ex}} E_{sun} b_{sc}(q)\left(1 - e^{-b_{ex} s}\right)$$



Extinction and In-Scattering

$$L(s,q) = L_0 \mathrm{F}_{\mathrm{ex}}(s) + L_{\mathrm{in}}(s,q)$$

- **Compare to hardware fog:**

$$L(s) = L_0(1 - \mathrm{f}(s)) + C_{\mathrm{fog}}\mathrm{f}(s)$$

  - **Monochrome extinction**
  - **Added color completely non-directional**

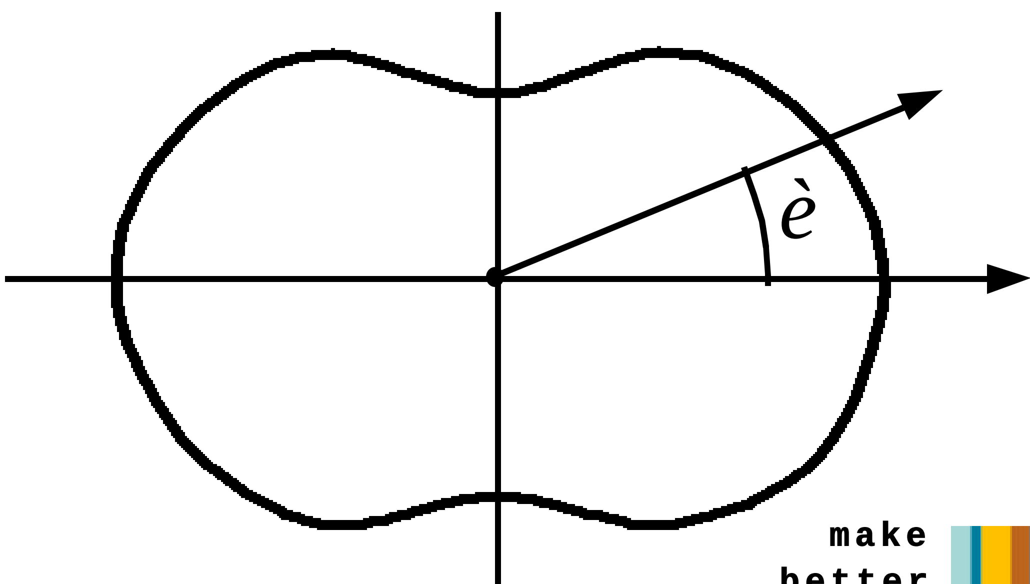
Rayleigh Scattering

- **Small particles (**$r < 0.05$ *ë***)**
- $\hat{a}_{sc}$ **is proportional to** $1/\lambda^4$

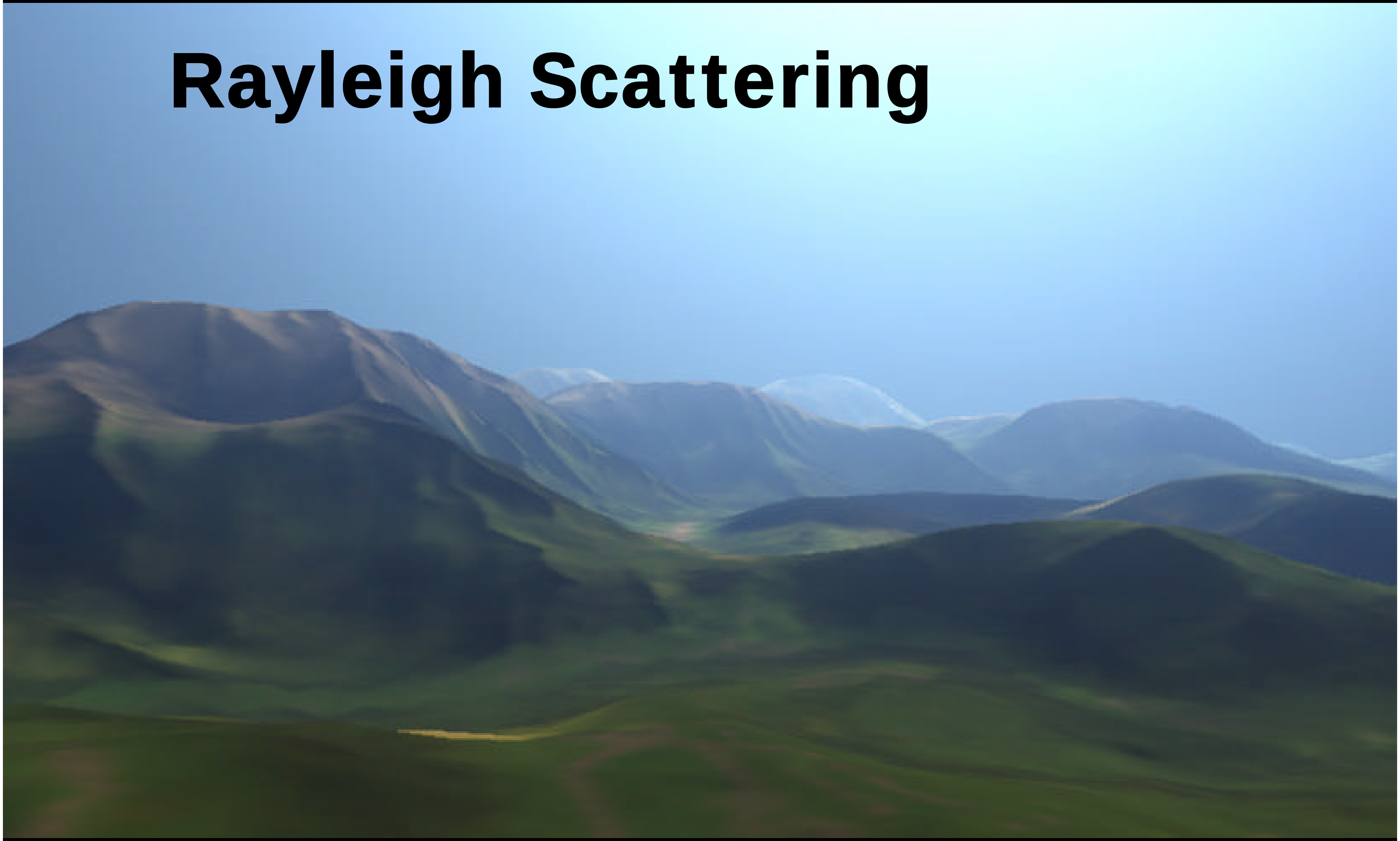
Rayleigh Scattering

- **Phase function:**

$$f_{\mathrm{R}}(q) = \frac{3}{16p}\left(1 + \cos^2 q\right)$$



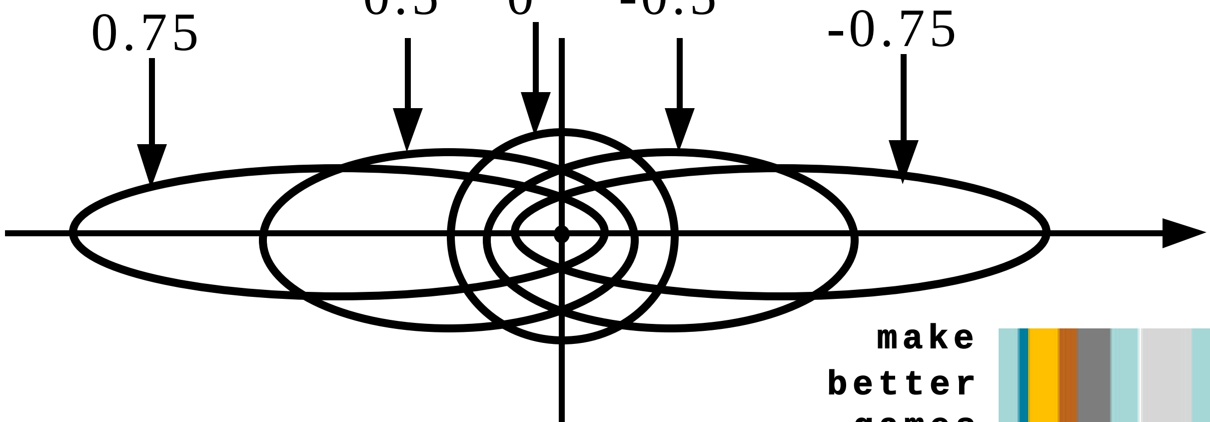
Rayleigh Scattering



Mie Scattering

- **Larger, spherical particles**
- **Phase function approximation:**
  - **Henyey-Greenstein**

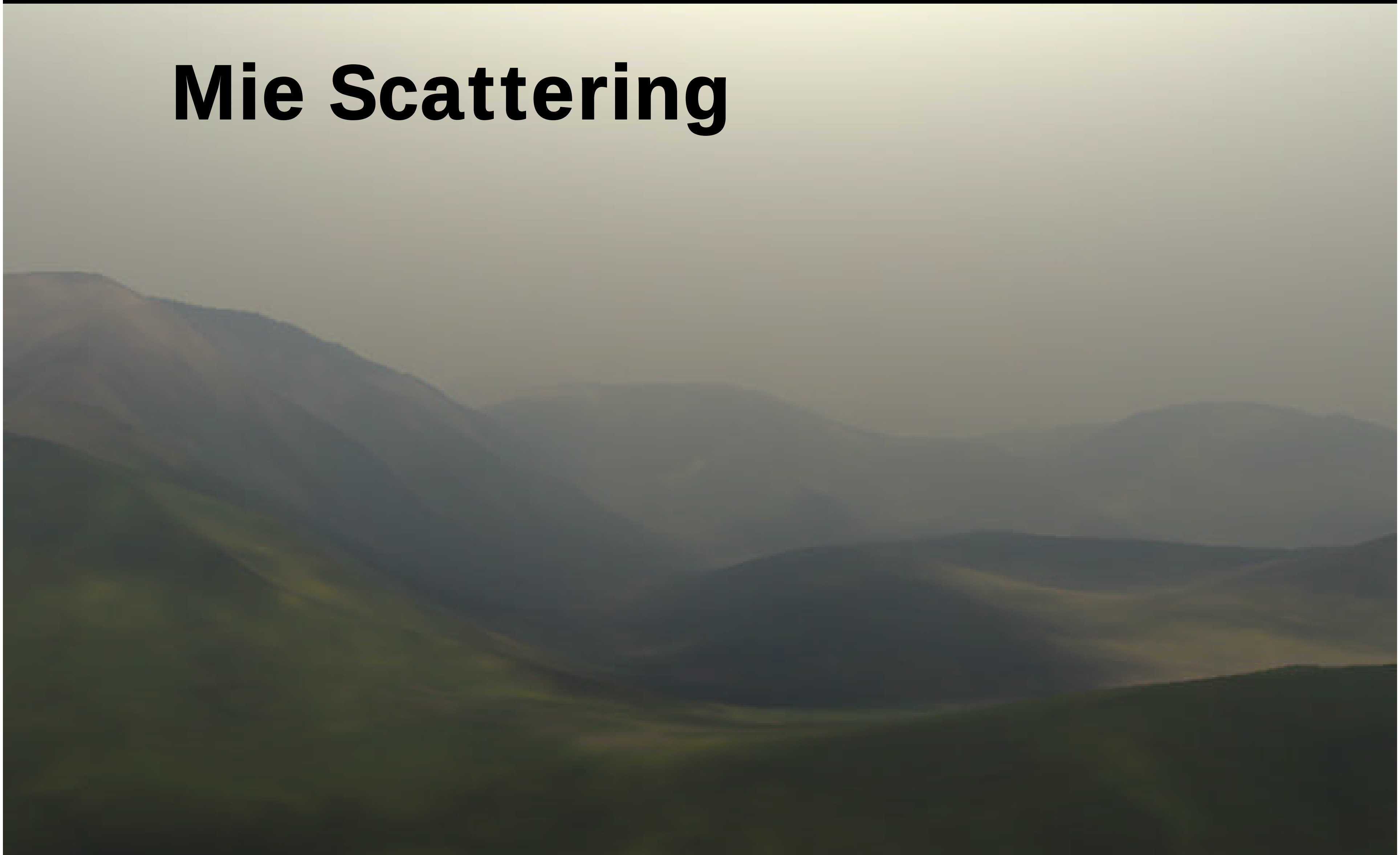
$$f_{\mathrm{HG}}(q) = \frac{(1-g)^2}{4p\left(1+g^2-2g\cos(q)\right)^{3/2}}$$



Mie Scattering

- Wavelength dependence
  - Complex and depends on exact size of particle
  - In practice, air usually contains a mix of various sizes of Mie particles
    - In the aggregate these tend to average out any wavelength dependence

Mie Scattering



Combined Scattering

- In practice, air contains both Rayleigh and Mie scatterers
- Absorption is usually slight
- We will use:

$$b_{\mathrm{ex}} = b_{\mathrm{sc}}^{\mathrm{Rayleigh}} + b_{\mathrm{sc}}^{\mathrm{Mie}}$$

$$b_{\mathrm{sc}}(q) = b_{\mathrm{sc}}^{\mathrm{Rayleigh}} f_{\mathrm{R}}(q) + b_{\mathrm{sc}}^{\mathrm{Mie}} f_{\mathrm{HG}}(q)$$

Parameters

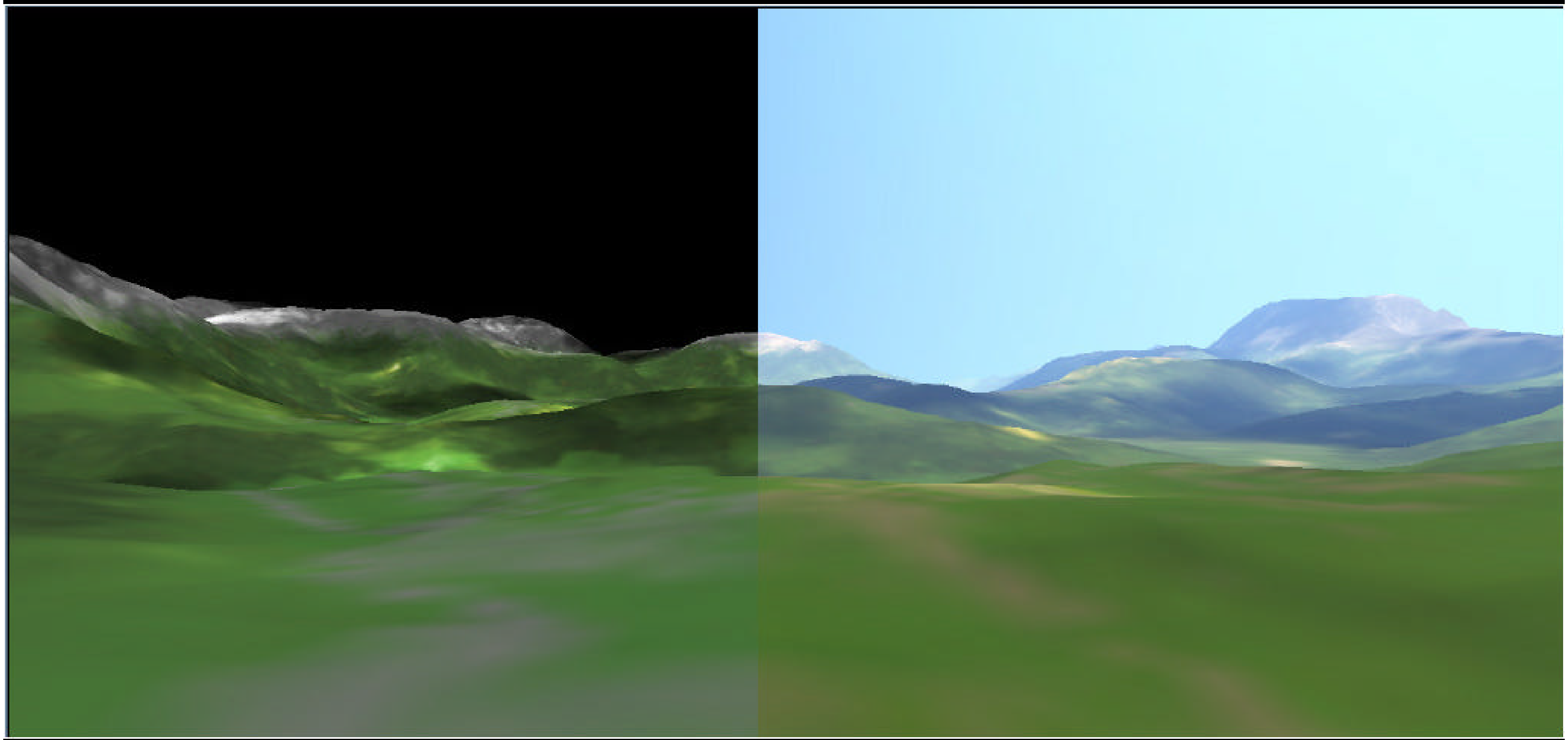
- **Atmospheric parameters:**

$$b_{sc}^{\mathrm{Rayleigh}} \qquad b_{sc}^{\mathrm{Mie}} \qquad g_{\mathrm{HG}}$$

- **Constant?** E_{sun}
  - **Affected by extinction**
- **Constant:** E_{sun}^{0}

Implementation

How ?

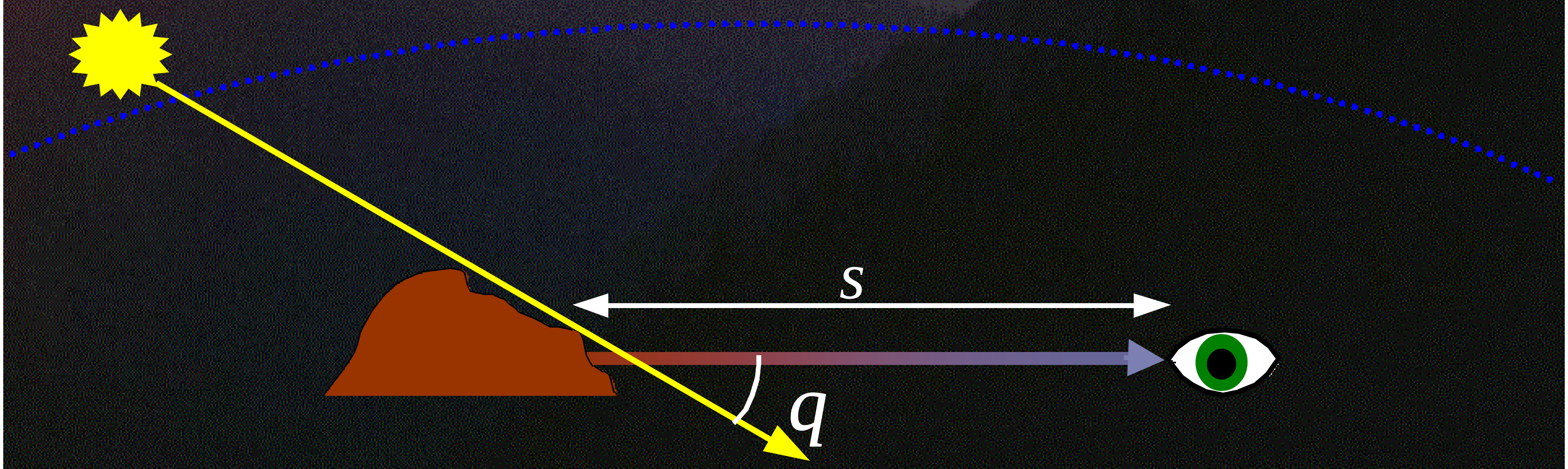


Without scattering

With scattering

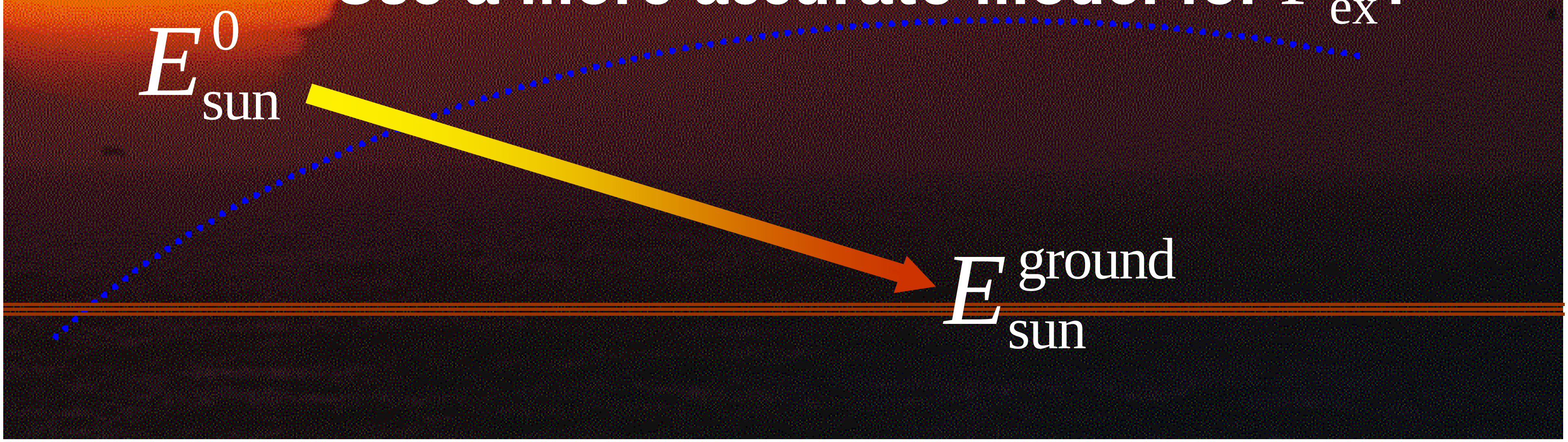
Implementation

- Aerial Perspective
  - Extinction & Inscattering
  - Rays low in atmosphere
  - Constant density good approximation



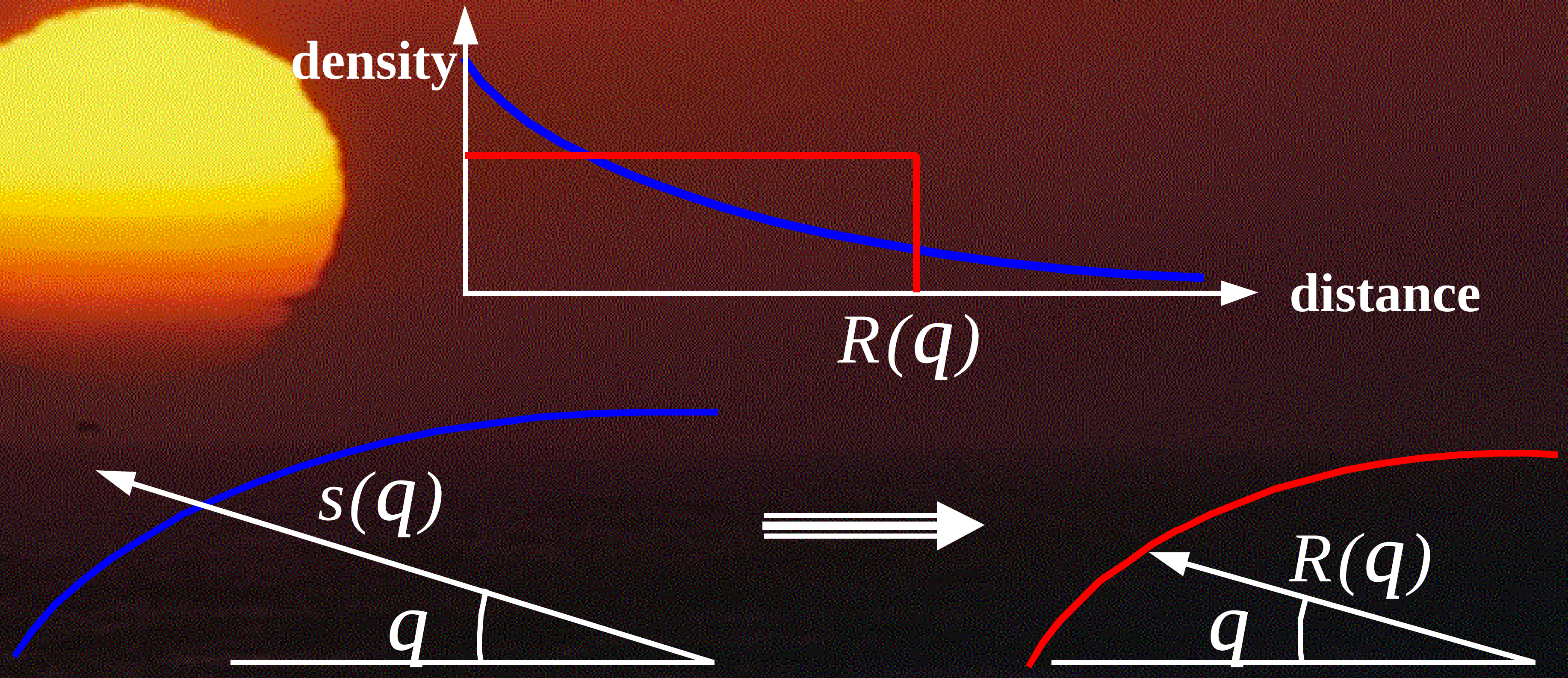
Implementation

- **Sunlight** $$E_{sun}^{ground} = E_{sun}^{0} F_{ex}$$
  - E_{sun}^{0} **is white**
  - **Density is <u>not</u> constant!**
  - **Use a more accurate model for F_{ex}?**



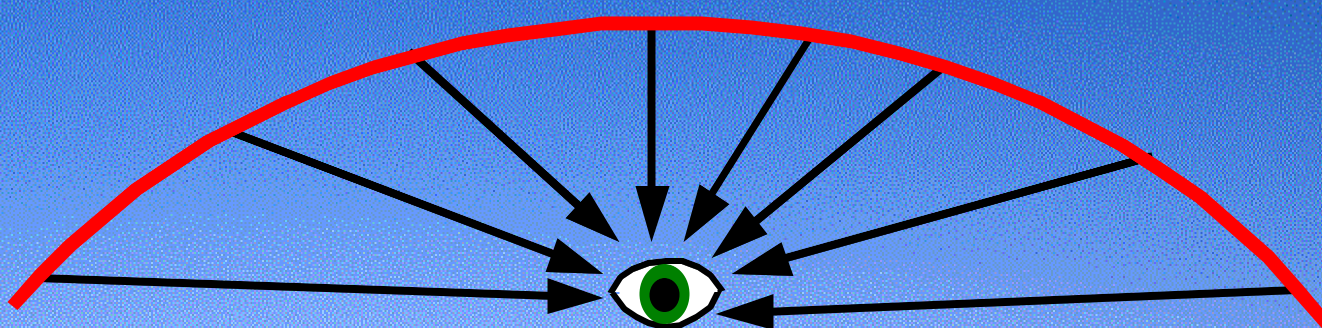
Implementation

- **Sunlight:**
  - Virtual sky dome, use simple model



Implementation

- **Sky color:** $L_{\mathrm{sky}}(q, j) = F_{\mathrm{in}}(q, j)$
  - **Density is <u>not</u> constant!**
  - **More accurate model too expensive**
    - **Many computations needed per frame**
- **Sky geometry**
  - **Virtual sky dome**



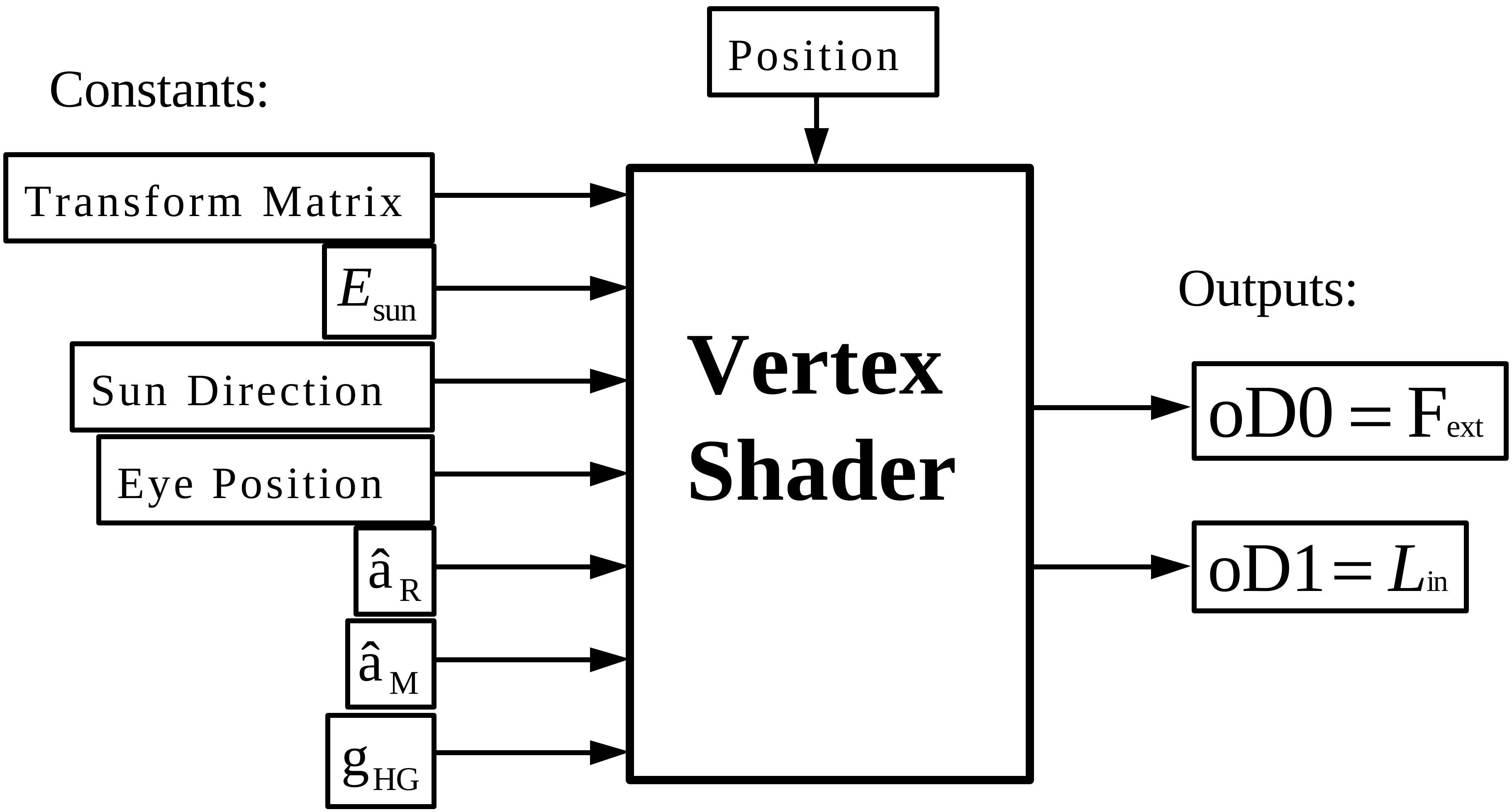
Implementation

- **Compute:** $L(s,q) = L_0 \mathrm{F}_{\mathrm{ex}}(s) + L_{\mathrm{in}}(s,q)$
  - **Can be done with textures**
    - **1D texture for** F_{ex}
      - **Texture coordinate is a function of** s
    - **2D texture for** L_{in}
      - **Texture coords are functions of** s, q
    - **Combine in pixel shader**
  - **We decided on a different approach**

Implementation

- **Compute:** $L(s,q) = L_0 F_{ex}(s) + L_{in}(s,q)$
  - **Use vertex shader to compute** F_{ex}, L_{in}
  - **Apply as vertex interpolated colors**
    - **In pixel shader, or even fixed pipeline**
  - **Pros:**
    - **Doesn’t use valuable texture slots**
    - **Can change atmosphere properties**
  - **Cons:**
    - **Somewhat dependent on tessellation**

Vertex Shader



Vertex Shader

$$L(s,q) = L_0 \mathrm{F}_{\mathrm{ex}}(s) + L_{\mathrm{in}}(s,q)$$

$$\mathrm{F}_{\mathrm{ex}}(s) = e^{-(b_{\mathrm{R}} + b_{\mathrm{M}})s}$$

$$L_{\mathrm{in}}(s,q) = \frac{b_{\mathrm{R}}(q) + b_{\mathrm{M}}(q)}{b_{\mathrm{R}} + b_{\mathrm{M}}} E_{\mathrm{sun}}\left(1 - e^{-(b_{\mathrm{R}} + b_{\mathrm{M}})s}\right)$$

$$b_{\mathrm{R}}(q) = \frac{3}{16p} b_{\mathrm{R}}\left(1 + \cos^2 q\right)$$

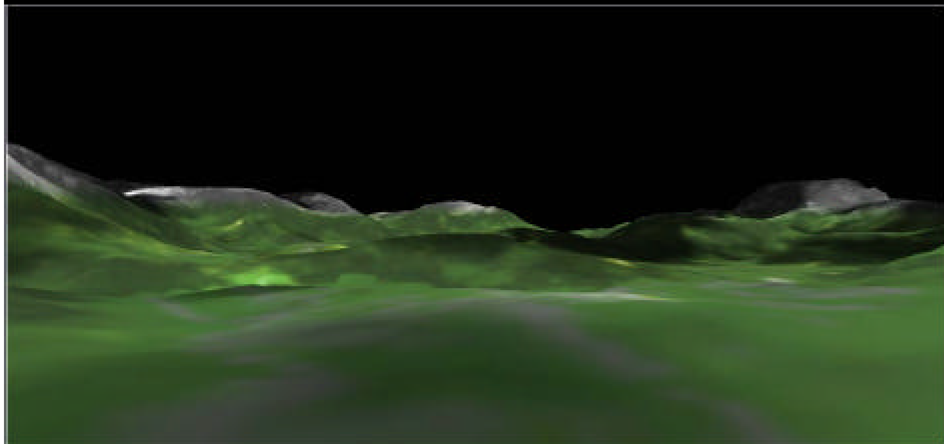
$$b_{\mathrm{M}}(q) = \frac{1}{4p} b_{\mathrm{M}} \frac{(1-g)^2}{\left(1 + g^2 - 2g\cos(q)\right)^{3/2}}$$

Vertex Shader

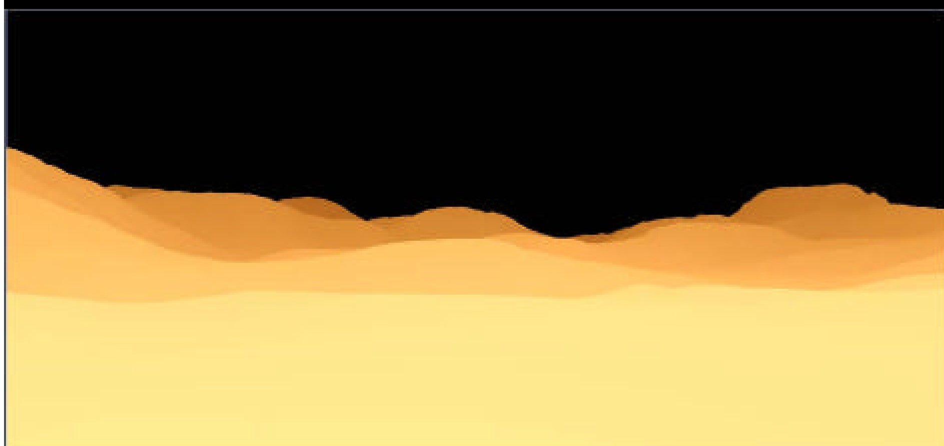
- Current Implementation:
  - 33 Instructions
    - Not including macro expansion
    - Could probably be optimized
  - 8 registers

Pixel Shader

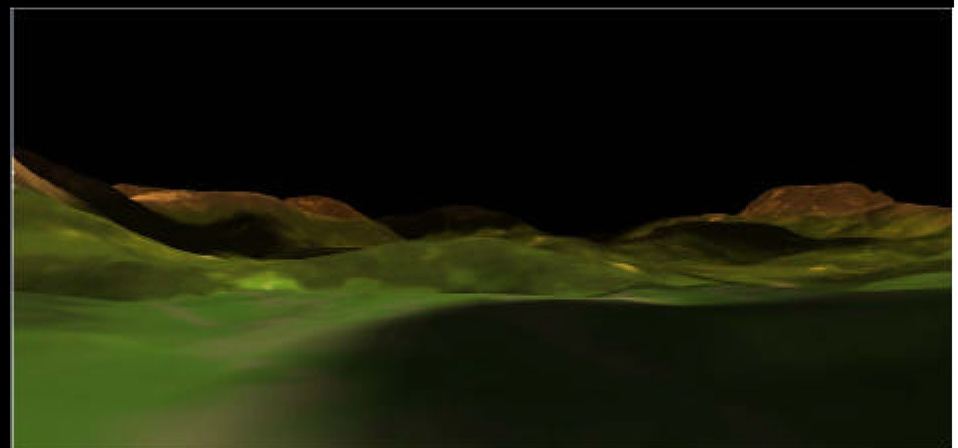
$$L = Lo * F\_ex + L\_in$$



Lo **x** **=**



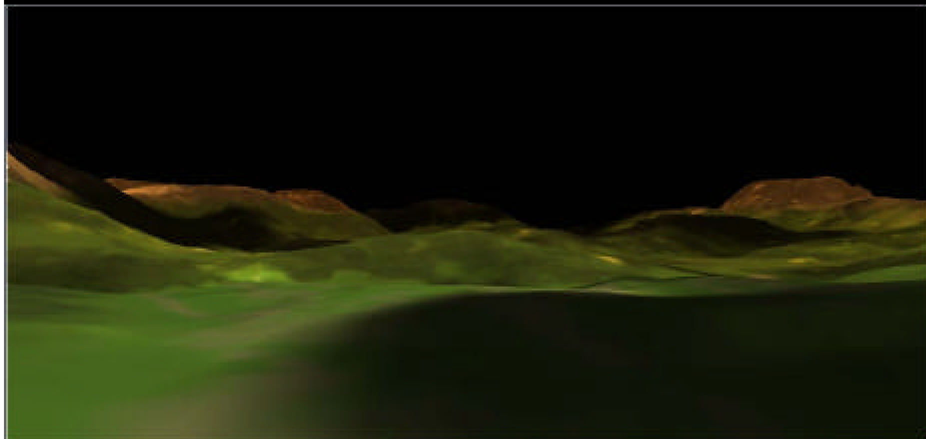
F_ex



Lo * F_ex

Pixel Shader

$$L = Lo * F\_ex + L\_in$$



Lo * F_ex

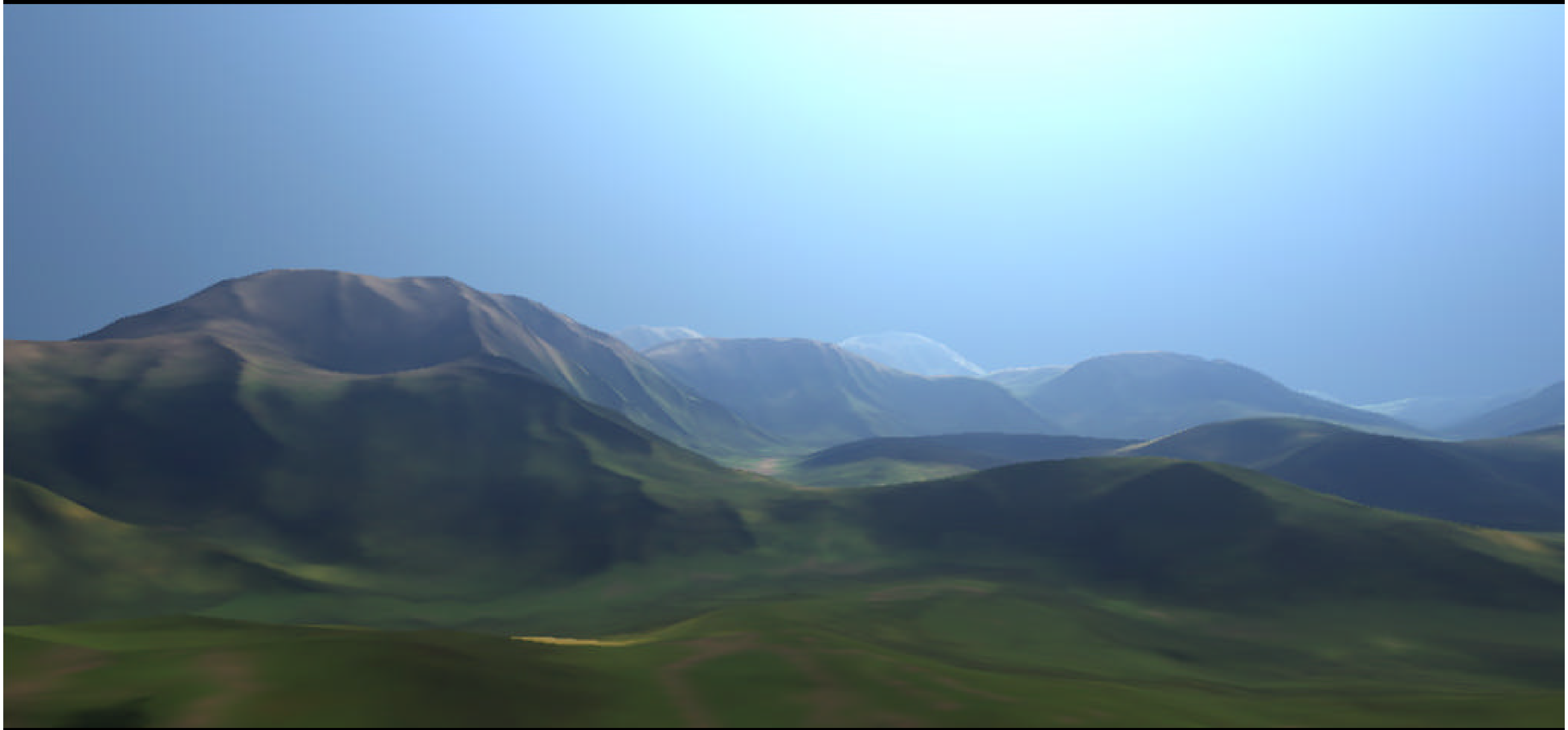
+

=

Lo * F_ex + L_in

L_in

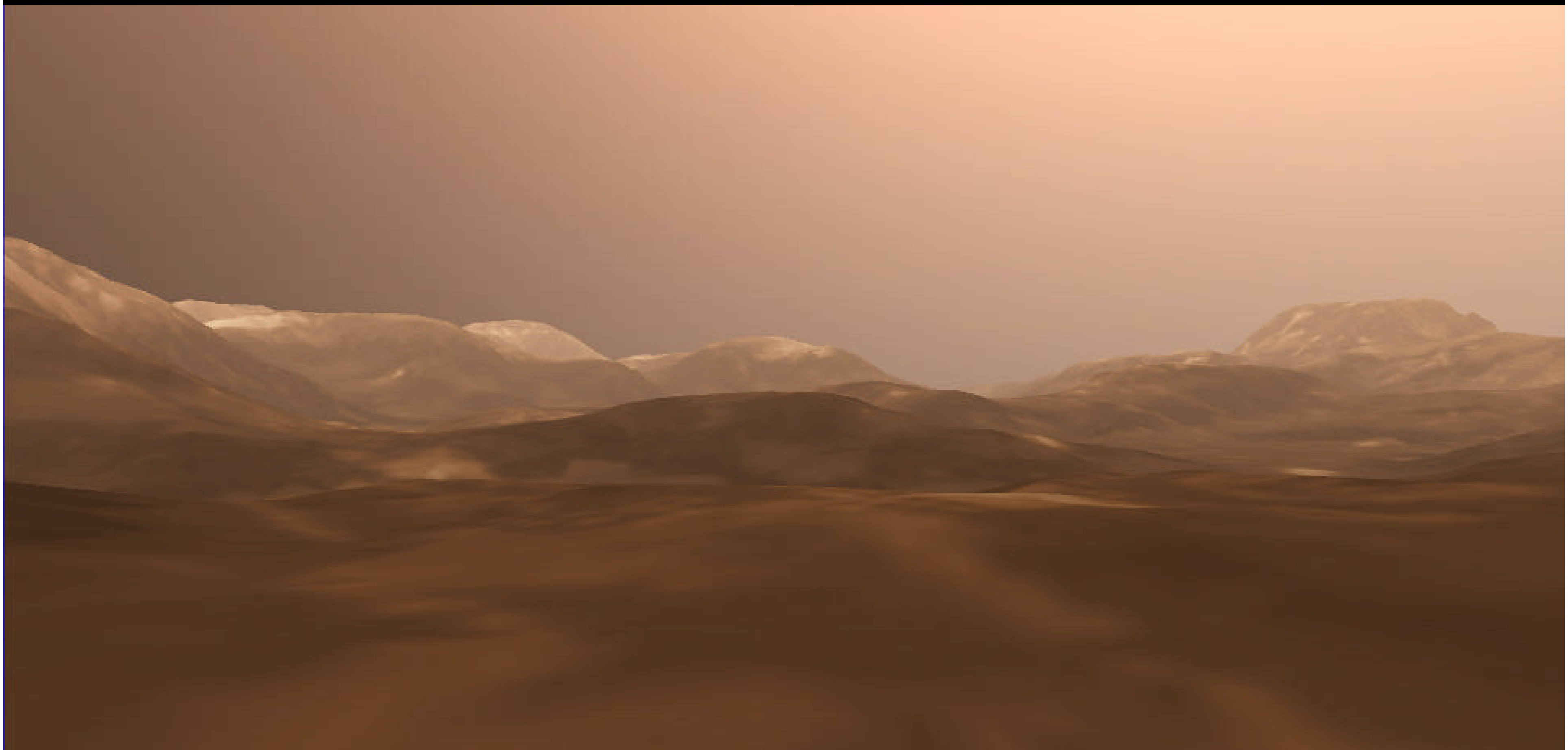
Results



Rayleigh Scattering - high
Mie Scattering - low
Sun Altitude - high

Rayleigh Scattering - low
Mie Scattering - high
Sun Altitude - high

Rayleigh Scattering - medium
Mie Scattering - medium
Sun Altitude - low



Planet Mars like scattering

Demo

Conclusion

- Scattering is easy to implement.
- Easy to add to an existing rendering framework
  - compute F_ex & L_in
  - apply these to existing color to get final color

Future Work

- In-scattering from sky
- Clouds (scattering and extinction)
- Volumetric cloud shadows
- Non-uniform density distributions
- Full-spectrum math?

GameDevelopers Conference

Acknowledgements

- We would like to thank
  - Kenny Mitchell for the terrain engine used in our demo
  - Solomon Srinivasan for help with the demo movie

References

[Blinn1982] J. F. Blinn. *Light Reflection Functions for Simulation of Clouds and Dusty Surfaces.*

[Dutré2001] P. Dutré. *Global Illumination Compendium.*

[Henyey1941] L. G. Henyey and J. L. Greenstein. *Diffuse Reflection in the Galaxy.*

[Hoffman2001] N. Hoffman and K. J. Mitchell. *Photorealistic Terrain Lighting in Real Time.*

[Klassen1987] R. V. Klassen. *Modeling the Effect of the Atmosphere on Light.*

[Mie1908] G. Mie. *Bietage zur Optik truber Medien Speziell Kolloidaler Metallosungen.*

[Preetham1999] A. J. Preetham, P. Shirley, B. E. Smits. *A Practical Analytic Model for Daylight.*

[Rayleigh1871] J. W. Strutt (Lord Rayleigh). *On the light from the sky, its polarization and colour.*

[Yee2002] H. Yee, P. Dutré, S. Pattanaik. *Fundamentals of Lighting and Perception: The Rendering of Physically Accurate Images*.

THANK YOU