Advanced Rendering

Motion Blur

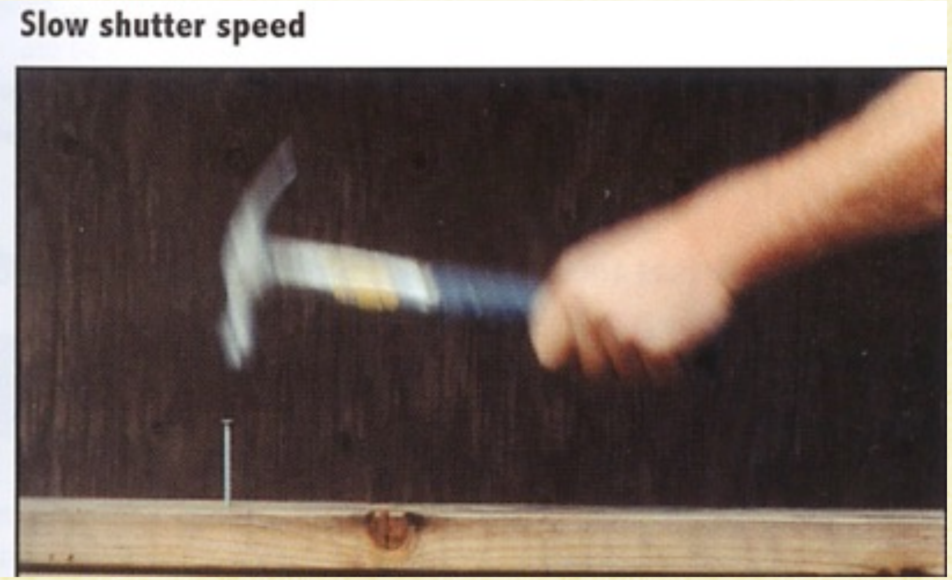

Olivier Cefai - Realflow,3dsmax,finalrender - render time 6m 9s

Shutter Speed

- The <u>shutter</u> limits the amount of light that hits the sensor
- While the shutter is open, moving objects create streaks on the sensor
- A faster shutter prevents motion blur, but limits the amount of entering light (often making the image too dark)

Ray Tracing Animated Geometry

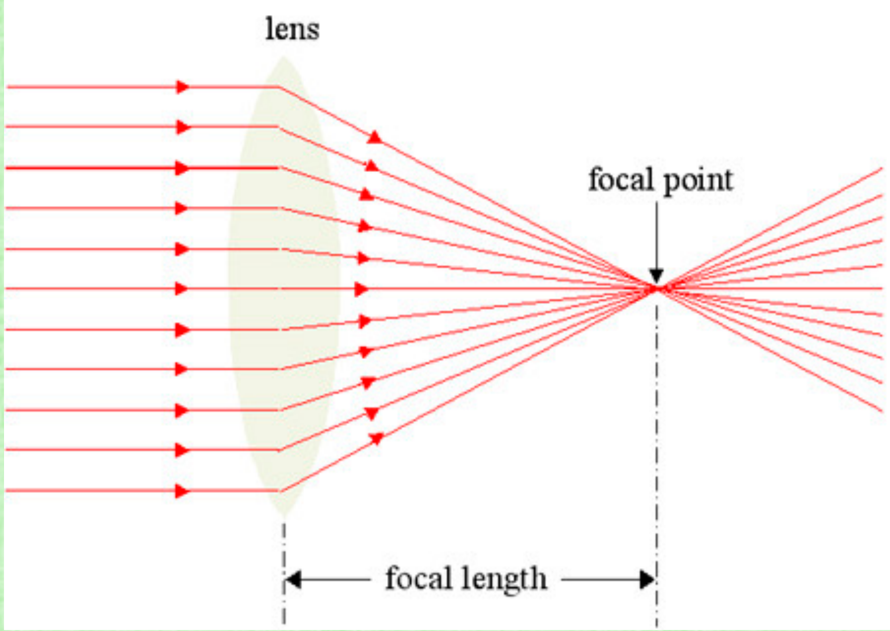
- Animate objects during the time interval $[T_0, T_1]$, when the shutter is open
  - Specify the object's transform as a function $F(t)$ for time $t \in [T_0, T_1]$
- Then, for each ray:
  - Assign a random time: $t_{ray} = (1 - \alpha)T_0 + \alpha T_1$ with $\alpha \in [0,1]$
  - Place the object into its time t_{ray} location, given by the transform $F(t_{ray})$
  - Trace the ray (as usual) to get a color
- Works significantly better when using many rays per pixel (to combat temporal aliasing)

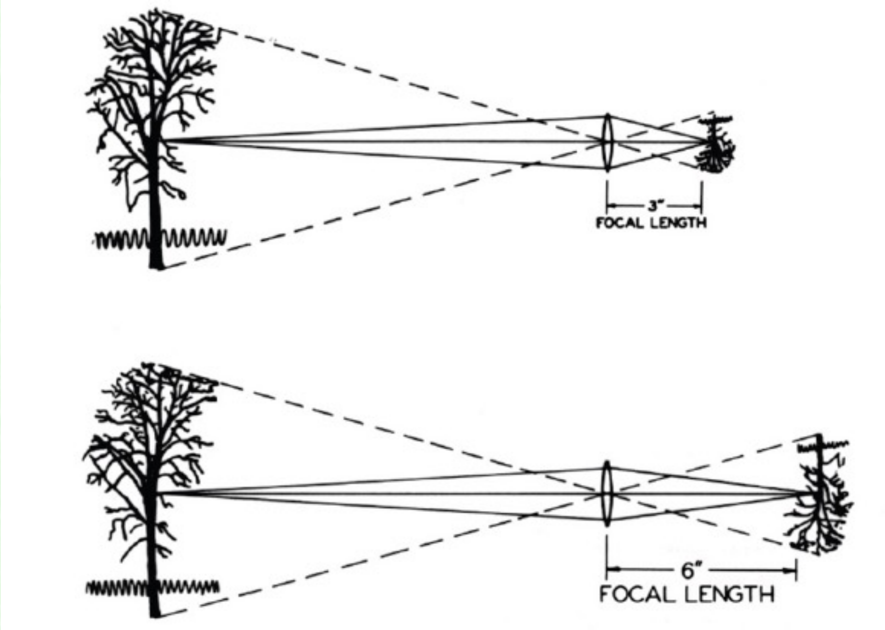


Depth of Field

Focal Length

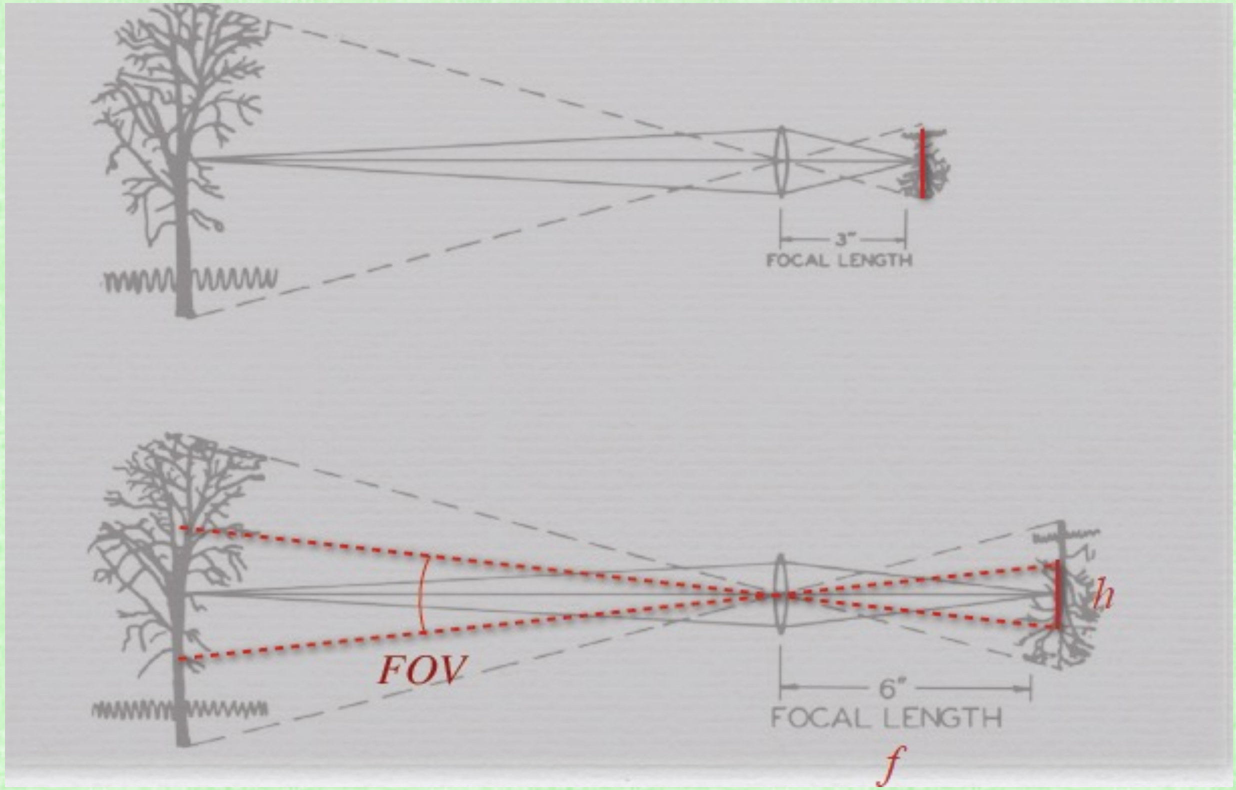
- The distance it takes for a lens to bring parallel rays into focus
- A stronger lens (or lens system) has a shorter focal length (bending rays more easily)
  - Individual elements of a lens system can be adjusted to change the overall focal length (but each individual lens has a fixed focal length)
- The farther away an object is, the closer the image plane should be placed to the focal point (for the object to be in focus)





Field of View

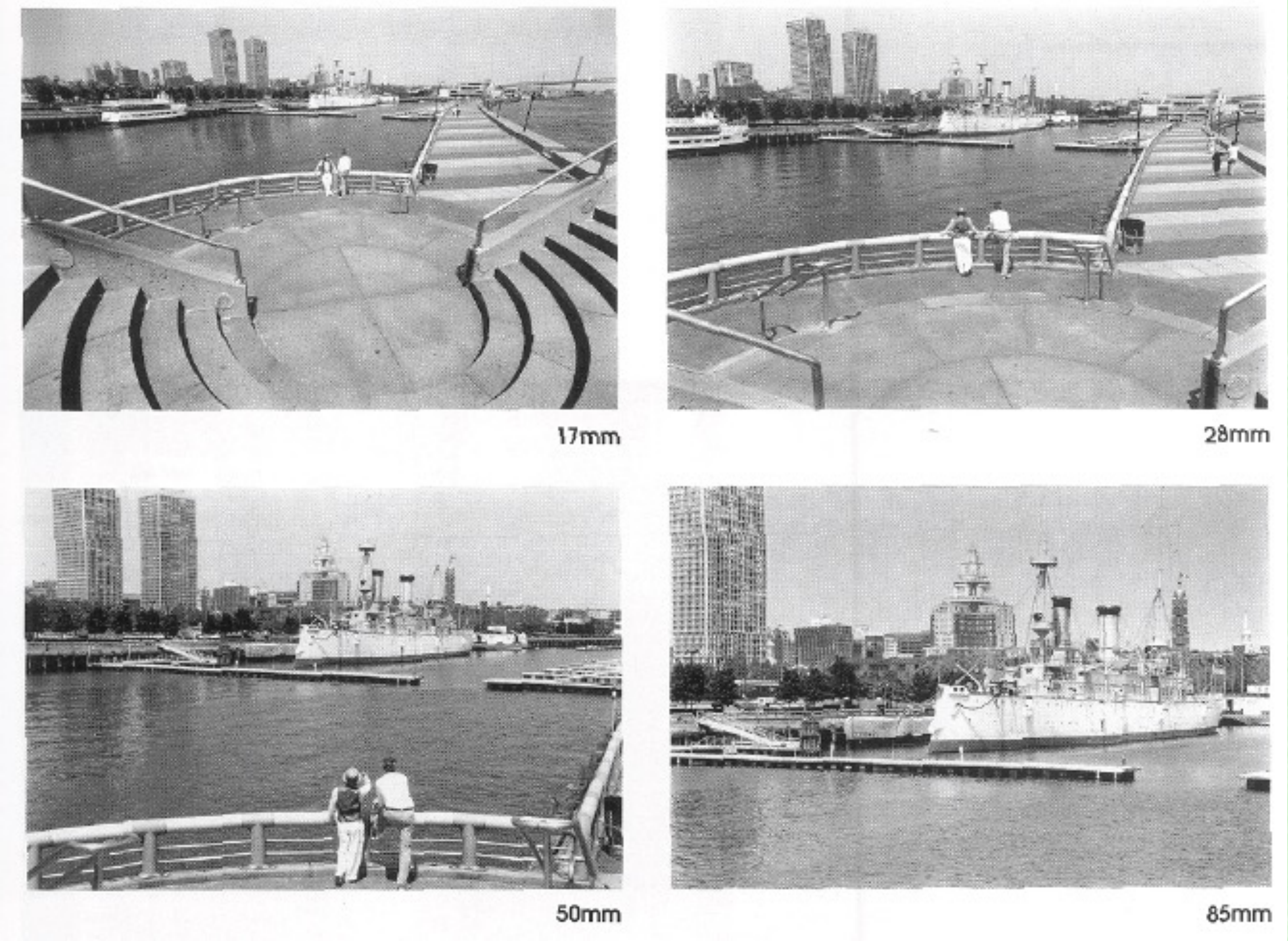
- Portion of the world visible to the sensor
- Zoom out/in by decreasing/increasing the focal length of a lens system
- Move the sensor in/out to adjust for the new focal length
- Since the sensor size doesn't change, the field of view expands/shrinks
- Get less/more pixels per feature (less/more detail)

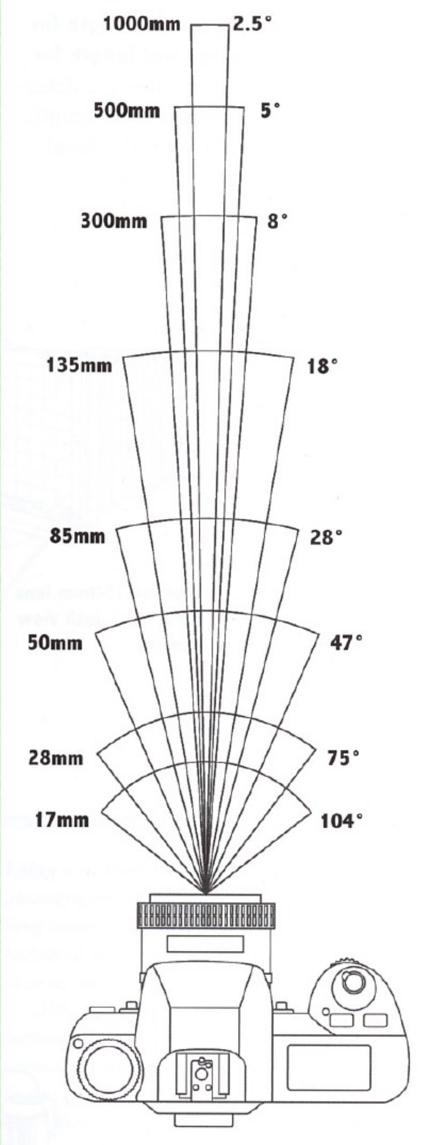


zoom out (decrease the distance)

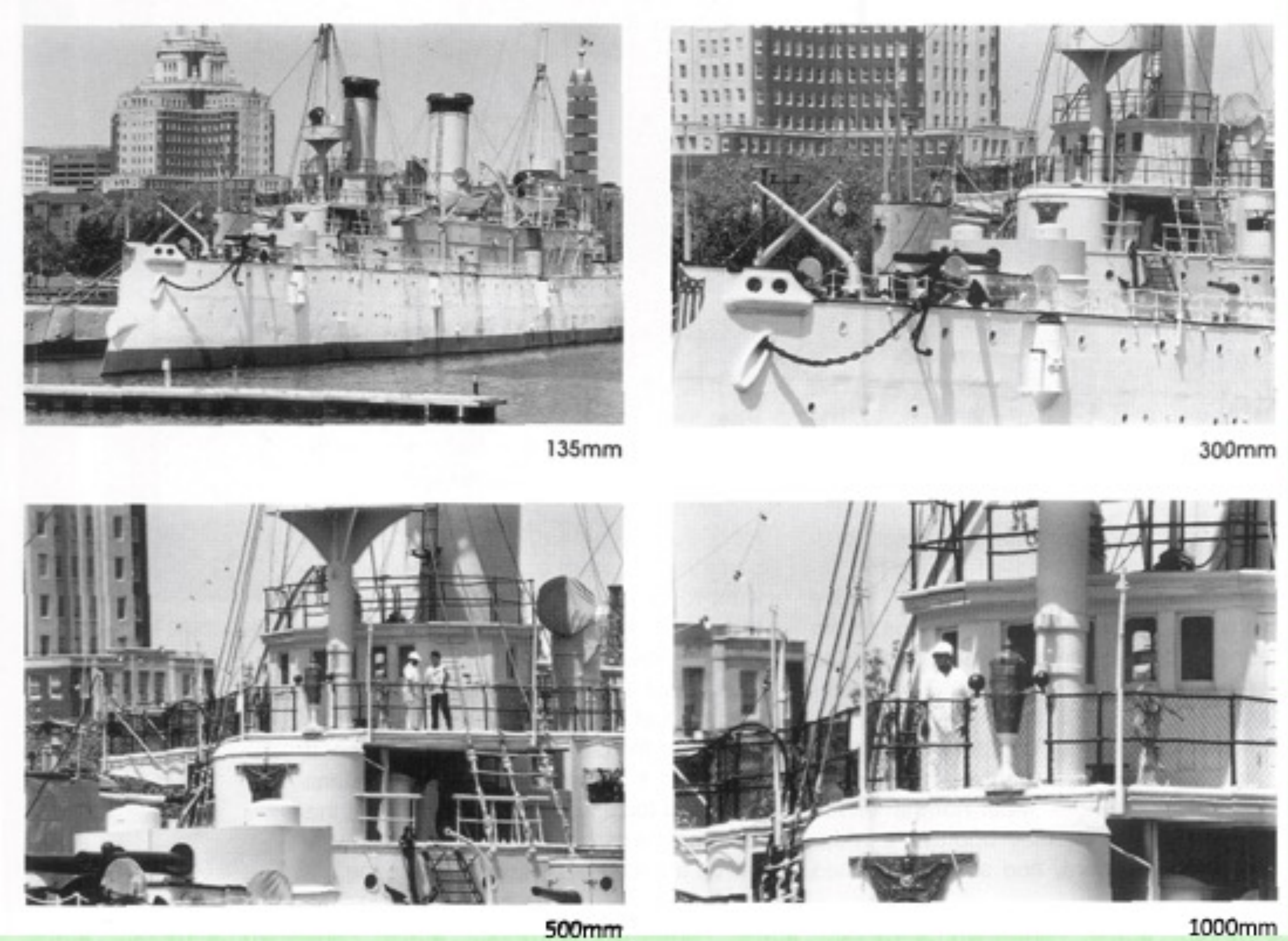
zoom in (increase the distance)

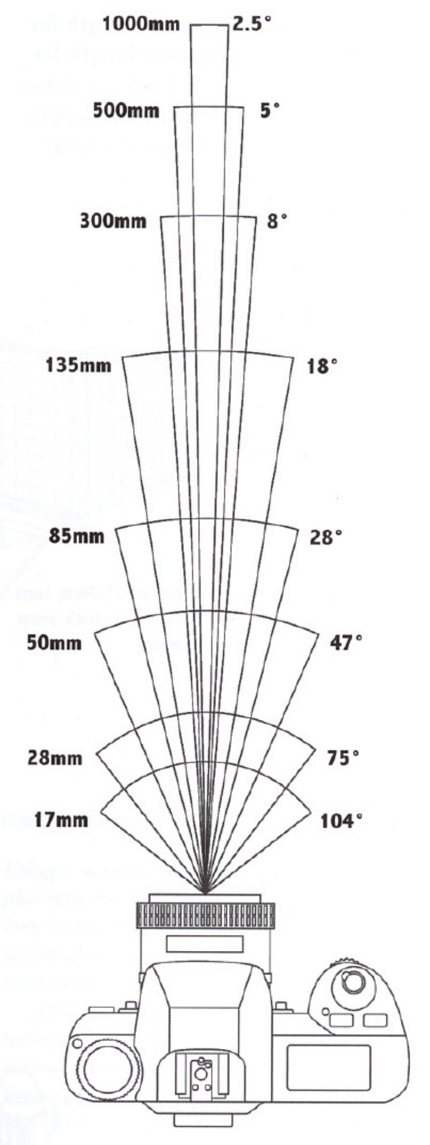
Zooming In shrinks the Field of View





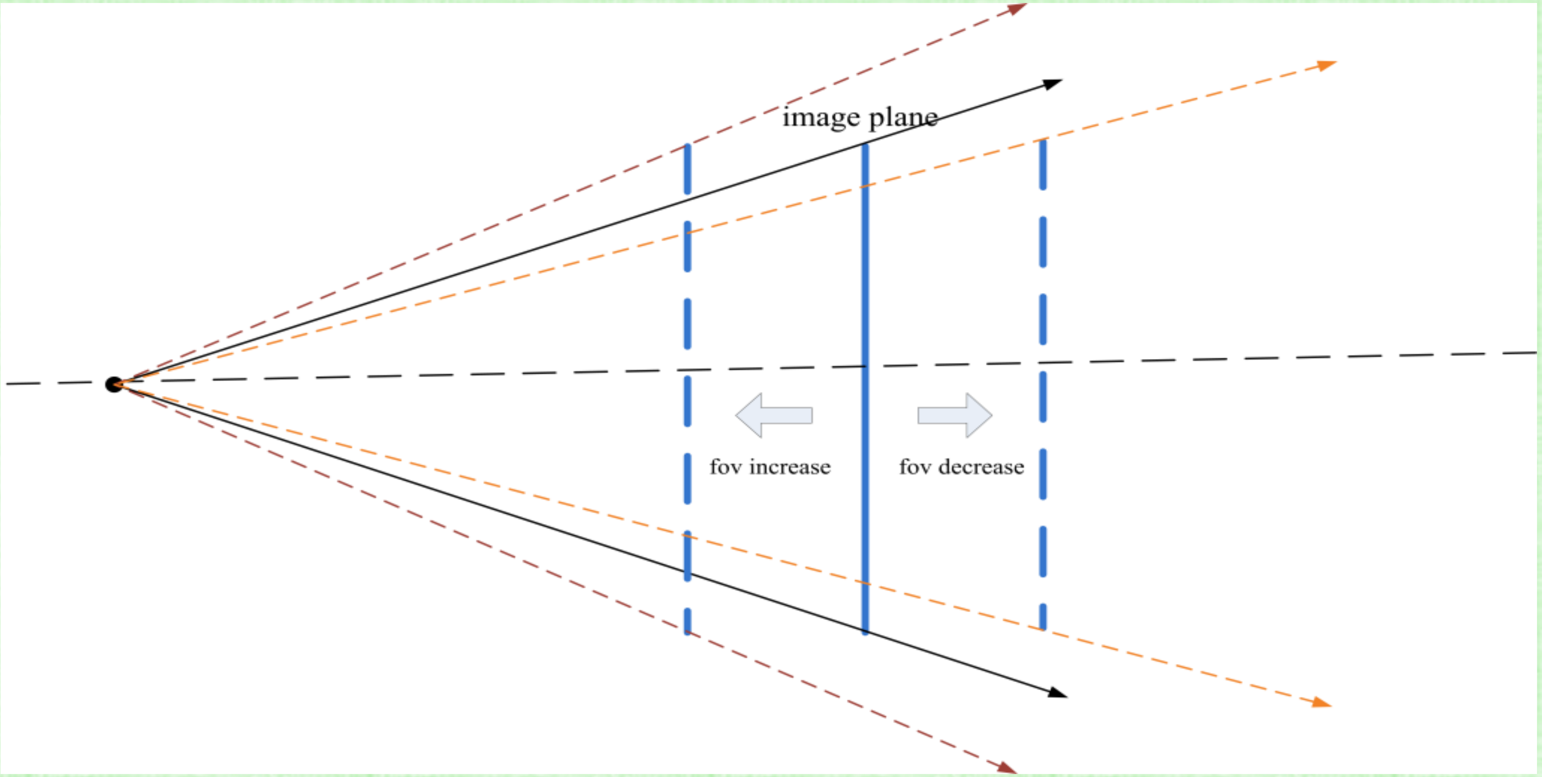
Zooming In shrinks the Field of View





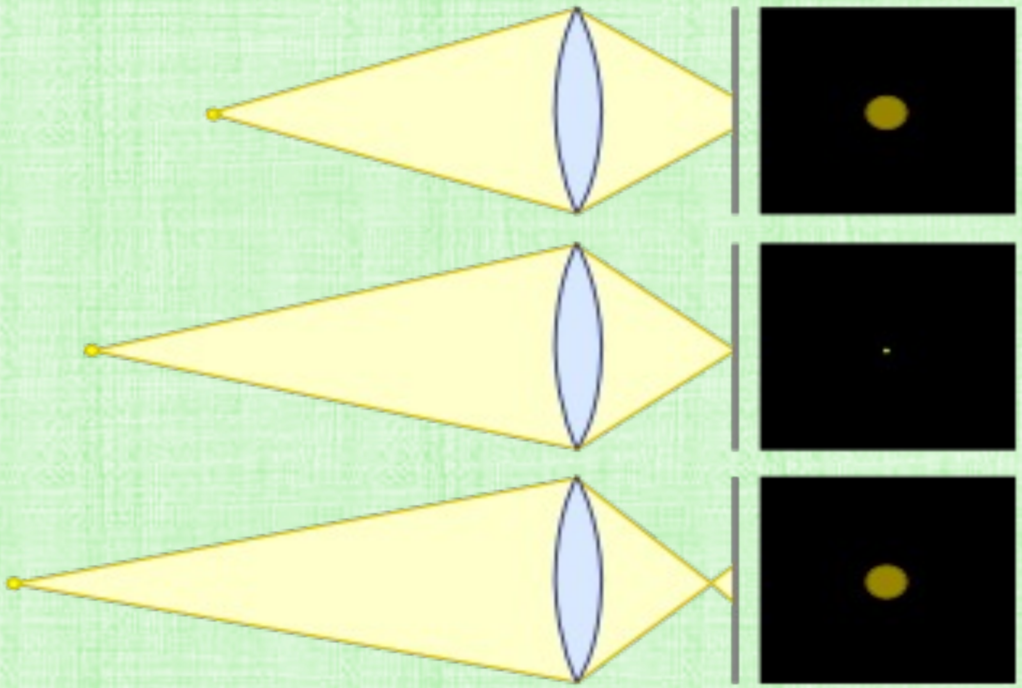
Ray Tracer Field of View (FOV)

- Ray tracer FOV is adjusted by changing the distance between the aperture and the image plane
- Alternatively, can change the sensor/film size (unlike in a real camera)
- Common mistake is to place the film plane too close to objects
  - Then, the desired FOV is (incorrectly) obtained by placing the aperture very close to the film plane, or by making a very large film plane (un-natural fish-eye lens effect)



Circle of Confusion

- An optical “spot”, caused by a cone of light rays not entirely re-focusing when imaging a point
- When the spot is about the size of a pixel, the object is “in focus”
- Objects at varying distances require varying sensor placement to keep the object “in focus”
- Depth of Field - distance between the nearest and farthest objects in a scene that appear to be “in focus” (i.e., the range of distances where the circle of confusion is not too big)

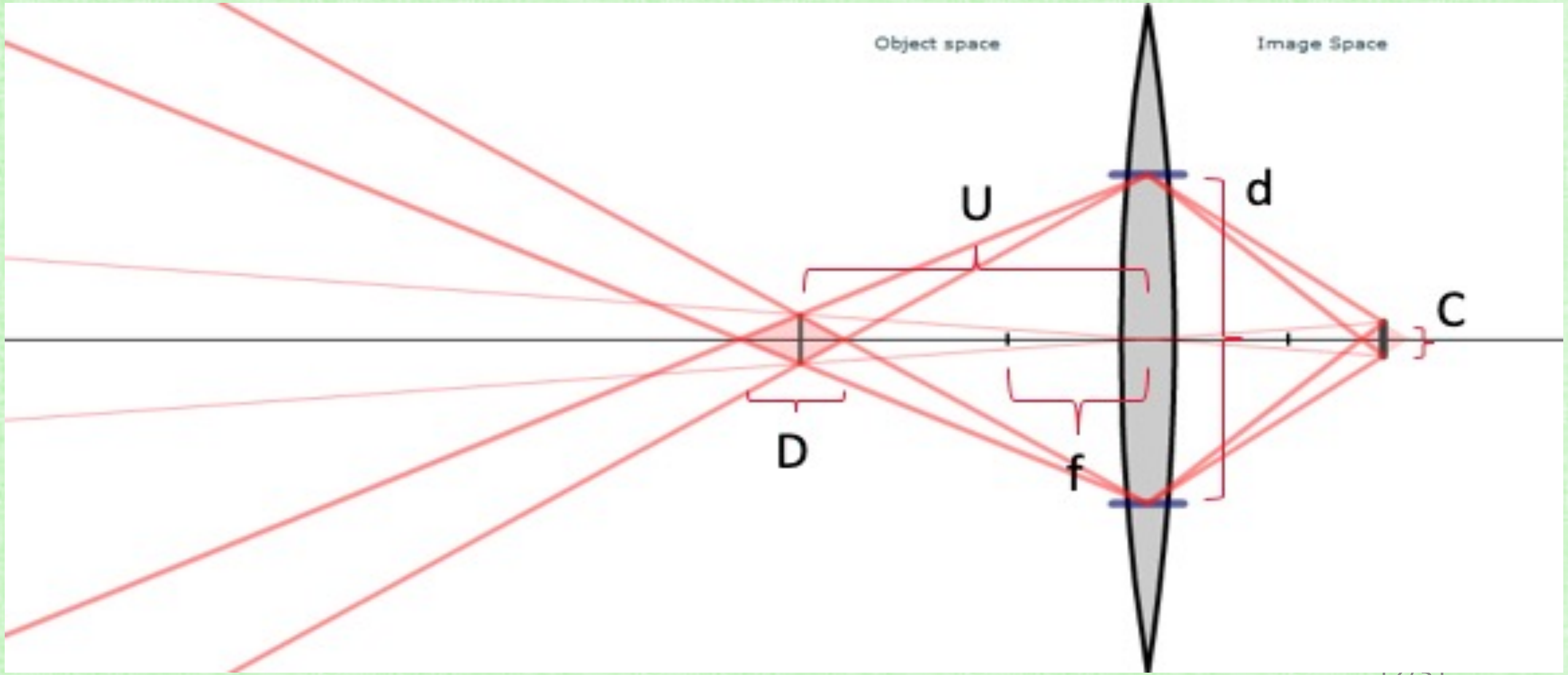


Depth of Field

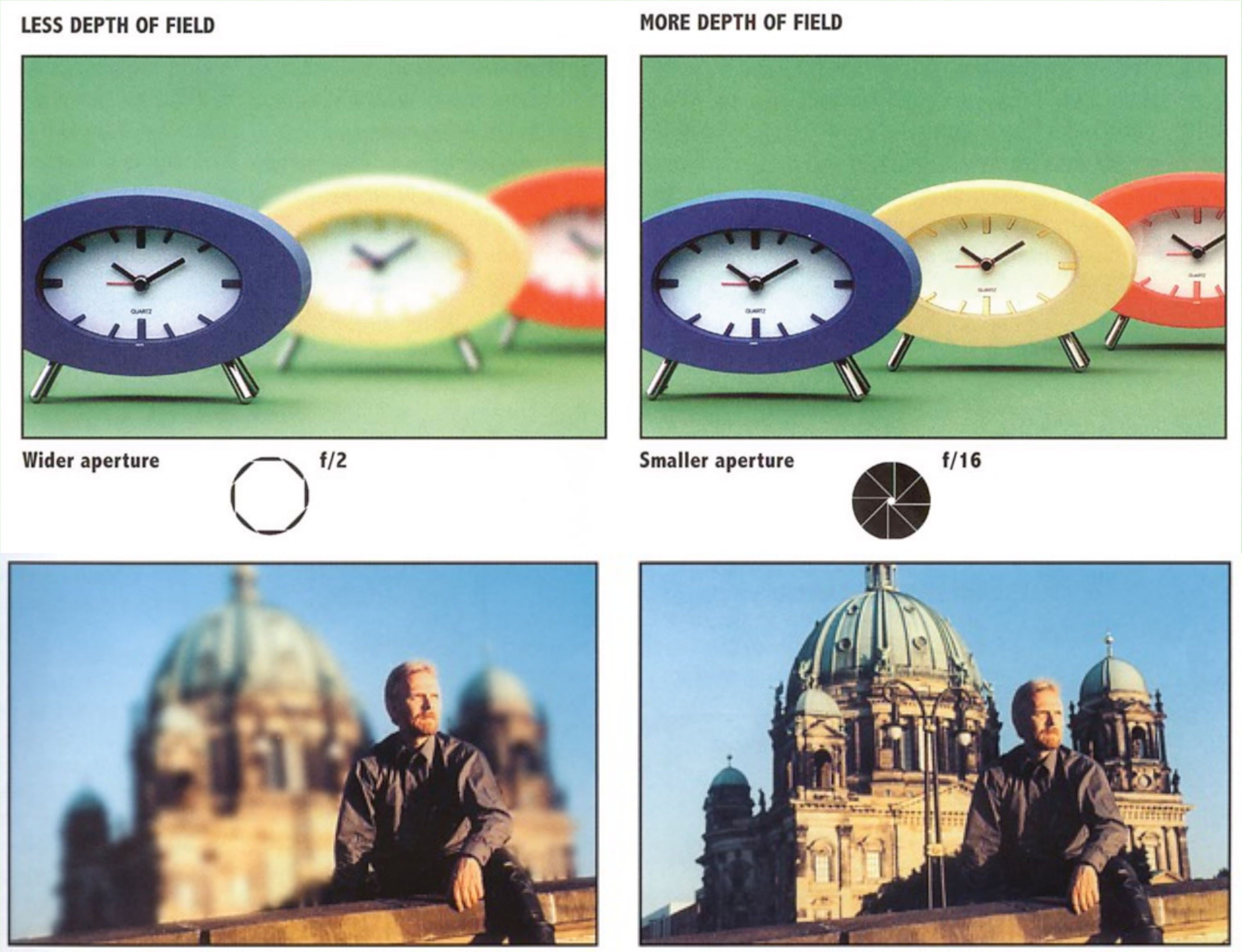
- Pinhole cameras have infinite depth of field
- Making the aperture smaller increases the depth of field
  - However, that limits amount of light entering the camera (and the image becomes too dark/noisy)
  - Decreasing shutter speed lets in more light (but creates motion blur)
  - Also, a small aperture causes undesirable light diffraction

$$D \sim \frac{U^2 C}{df}$$

- d is aperture diameter
- U is the distance to the center of focus
- f is focal length
- C is allowable circle of confusion

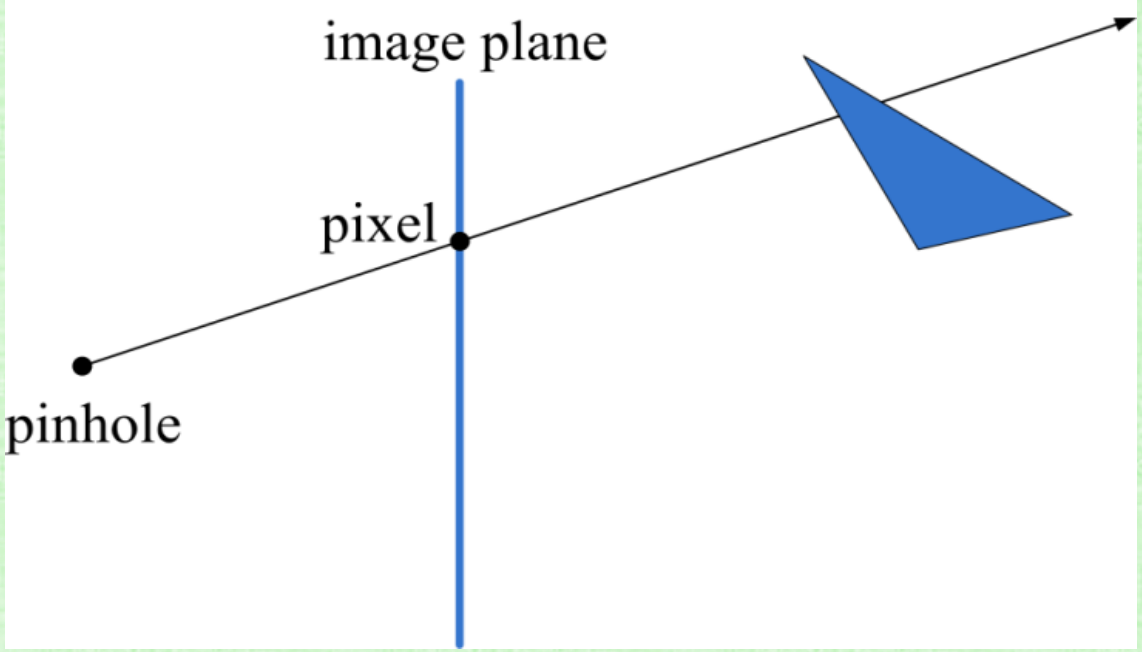


Aperture vs. Depth of Field

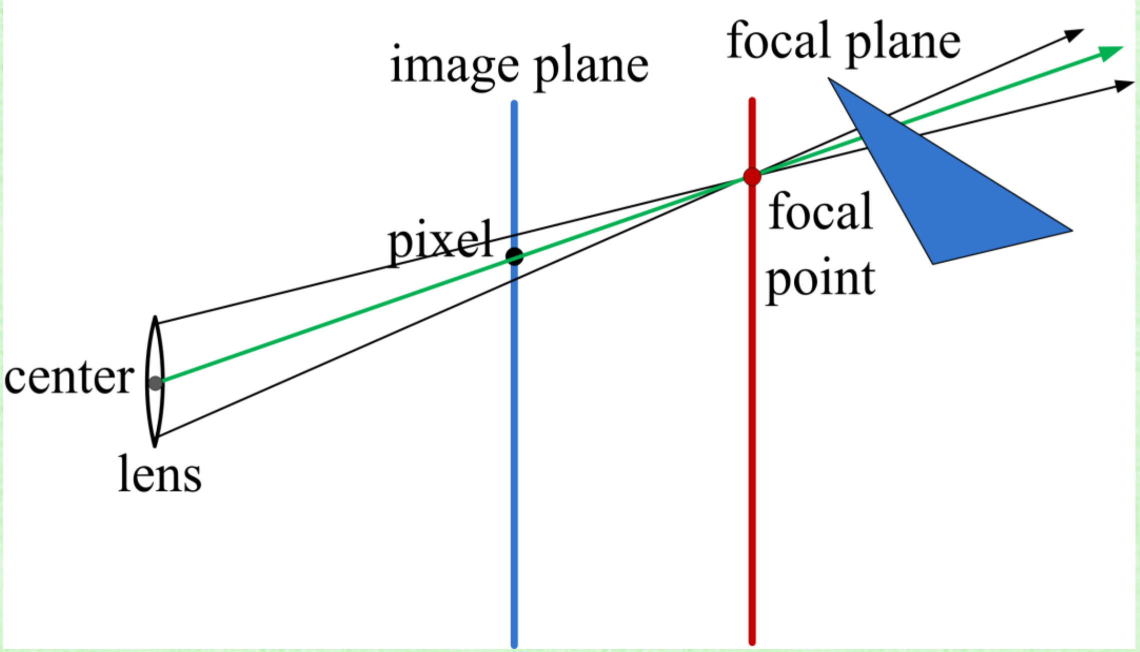


Depth of Field Ray Tracer

- Specify the focal plane (**red**) where objects are desired to be in focus
- For each pixel:
  - Calculate the “focal point” by intersecting the standard ray (**green**) with the focal plane (**red**)
  - Replace the pinhole (aperture) with a circular region
  - Cast multiple rays from the circular region through the focal point (and average the results)
- Objects further away from the focal plane are more blurred



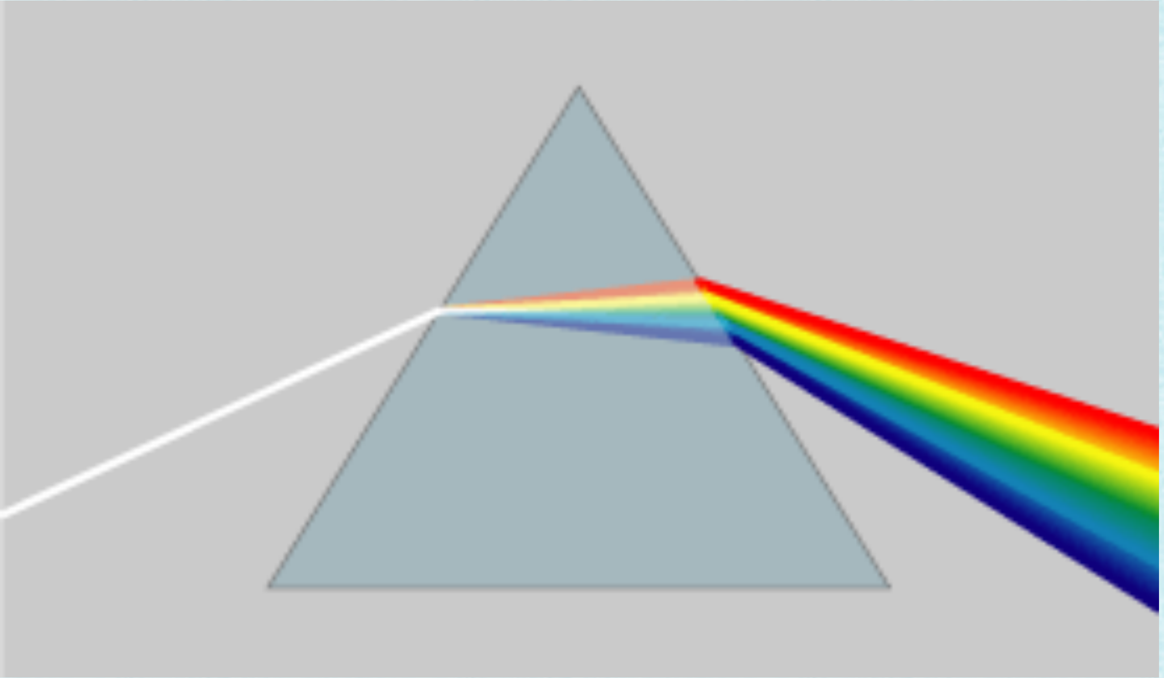
standard ray tracer



depth-of-field ray tracer

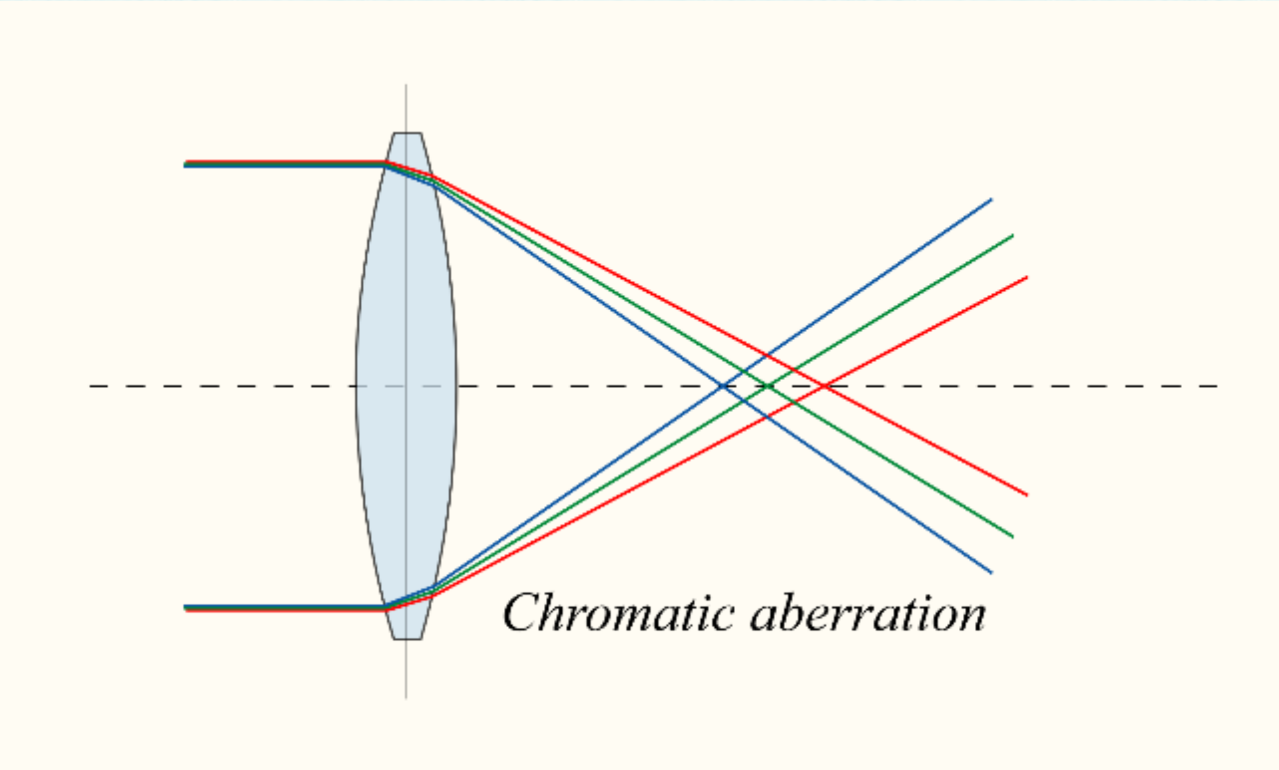
Dispersion

- The index of refraction depends on the frequency/wavelength of the light
- Index of refraction: air $n_1(\lambda) \approx 1$, glass/water $n_2(\lambda) > 1$
- Typically, n decreases (towards 1) as wavelength increases
- So, **blue light** ($\lambda \approx 400\text{nm}$) bends more than **red light** ($\lambda \approx 700\text{nm}$)
- Cauchy's approximation: $n(\lambda) = A + \frac{B}{\lambda^2} + \frac{C}{\lambda^4}$ with material parameters A, B, C



Chromatic Aberration

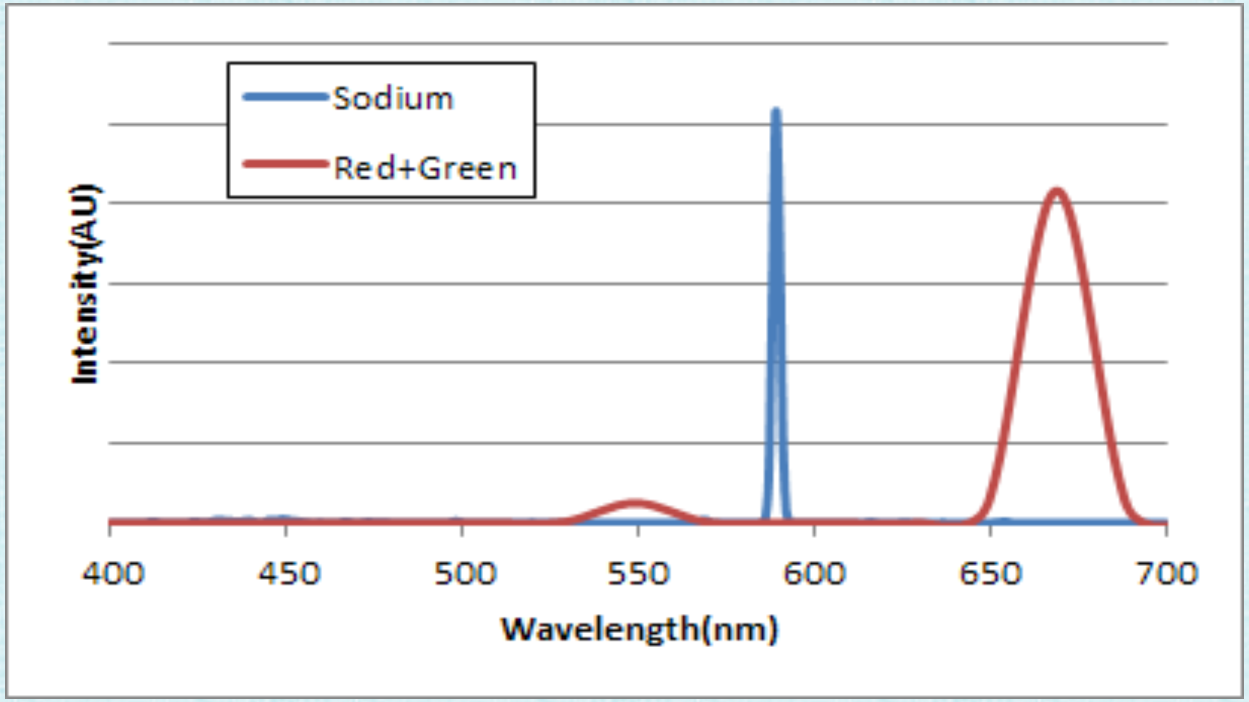
- Blue light bends more easily than red light, resulting in unequal focusing
- Focusing the blue light blurs the red light, and vice versa



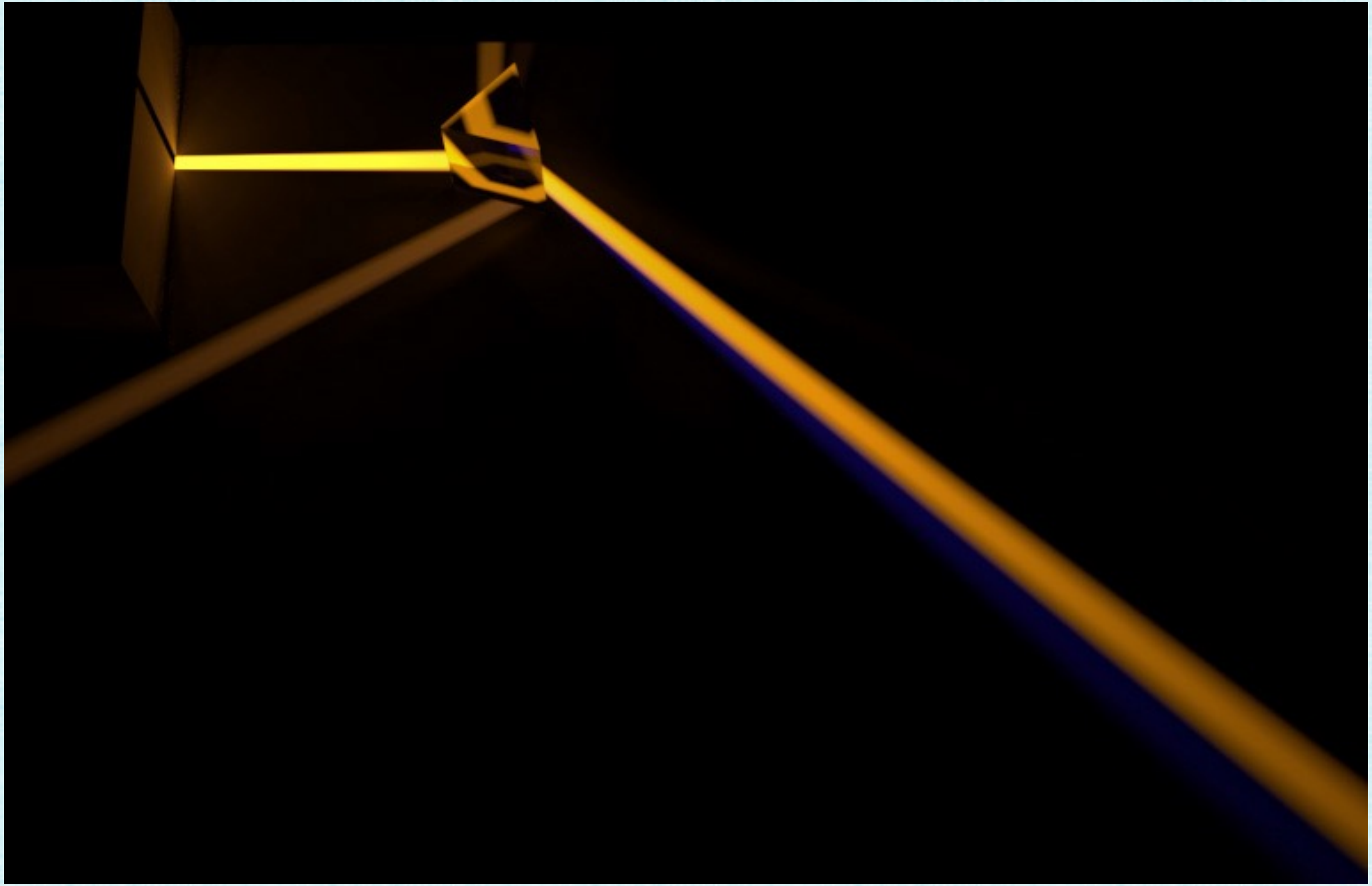
Spectral Power Distribution

- Light's interaction with materials cannot (generally) be described using only RGB values
  - The same RGB values map to many different power distributions (as we have discussed)
- Light's interaction with materials (often) requires the use of spectral power distributions

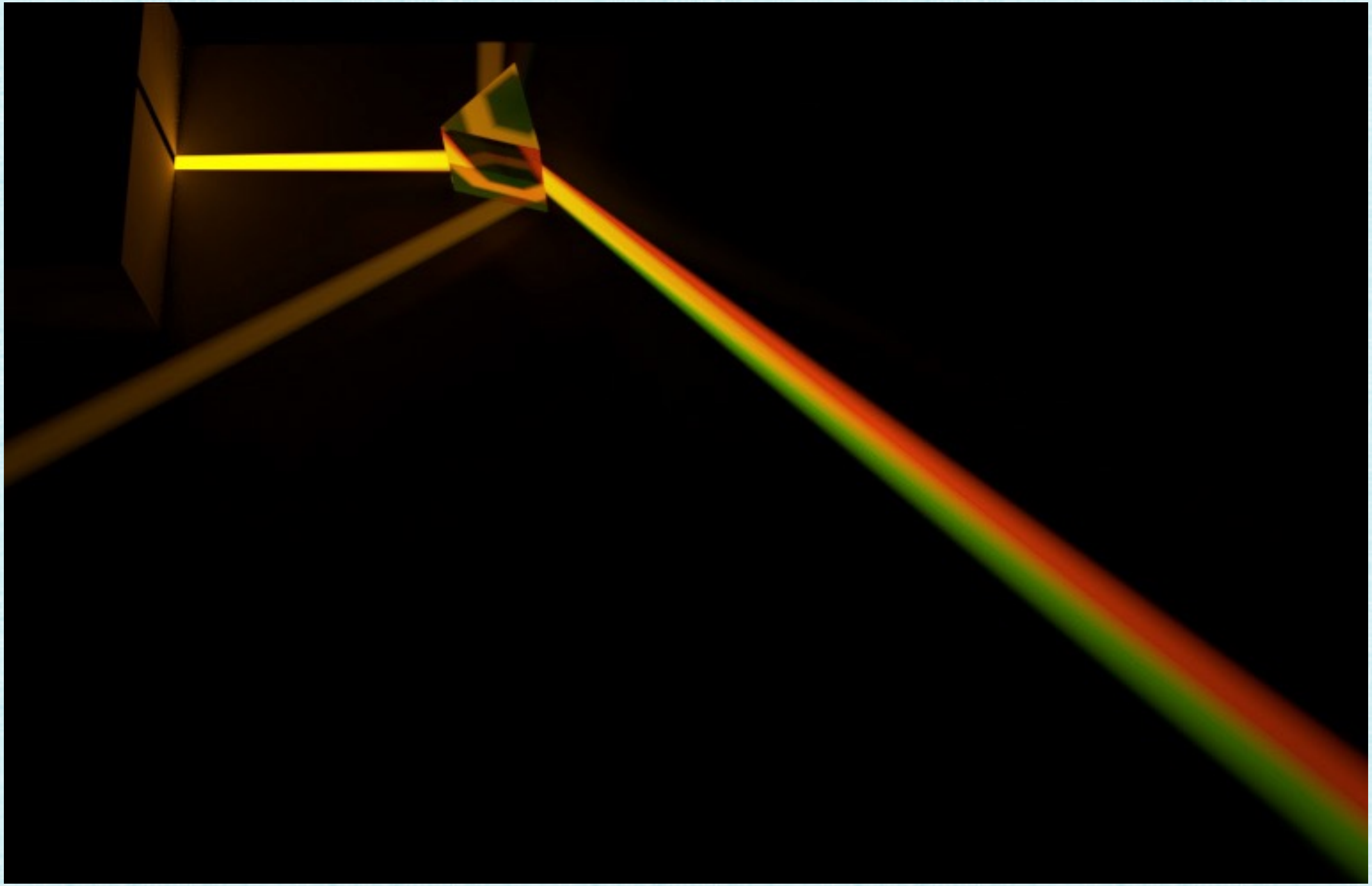
- Consider 2 different lights, with **identical RGB values** but **different spectral power distributions**:



Sodium Light



Red/Green Light



Wavelength Light Map

- When tracing photons from a light source, <u>importance sample the spectral power distribution</u> (instead of using R,G,B) to obtain a λ for each photon
- Use the photon's λ (and the reflectance/transmittance behavior for λ) to trace the photon throughout the scene
- Store incident power and wavelength of the photon in the photon map (λ-colored lights)

Gathering (from a Wavelength Light Map)

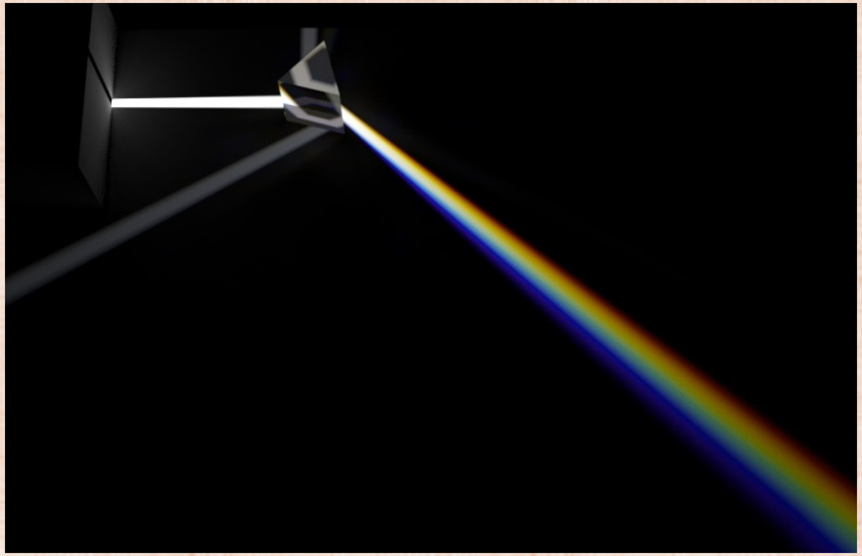
- When tracing rays from the camera, calculate the spectral power distribution at an intersection point using the nearby (λ-colored) photons and the BRDF (as usual)
- Multiply/Integrate the calculated spectral power distribution by the tristimulus response functions to obtain R, G, B values (to store in the image, as usual)
- Requires significantly more samples in the photon map

Participating Media

- Light is scattered towards the eye/camera by dust, mist, etc.

Participating Media

- That’s how we see the light from the prism experiment (above), or a rainbow



Absorption

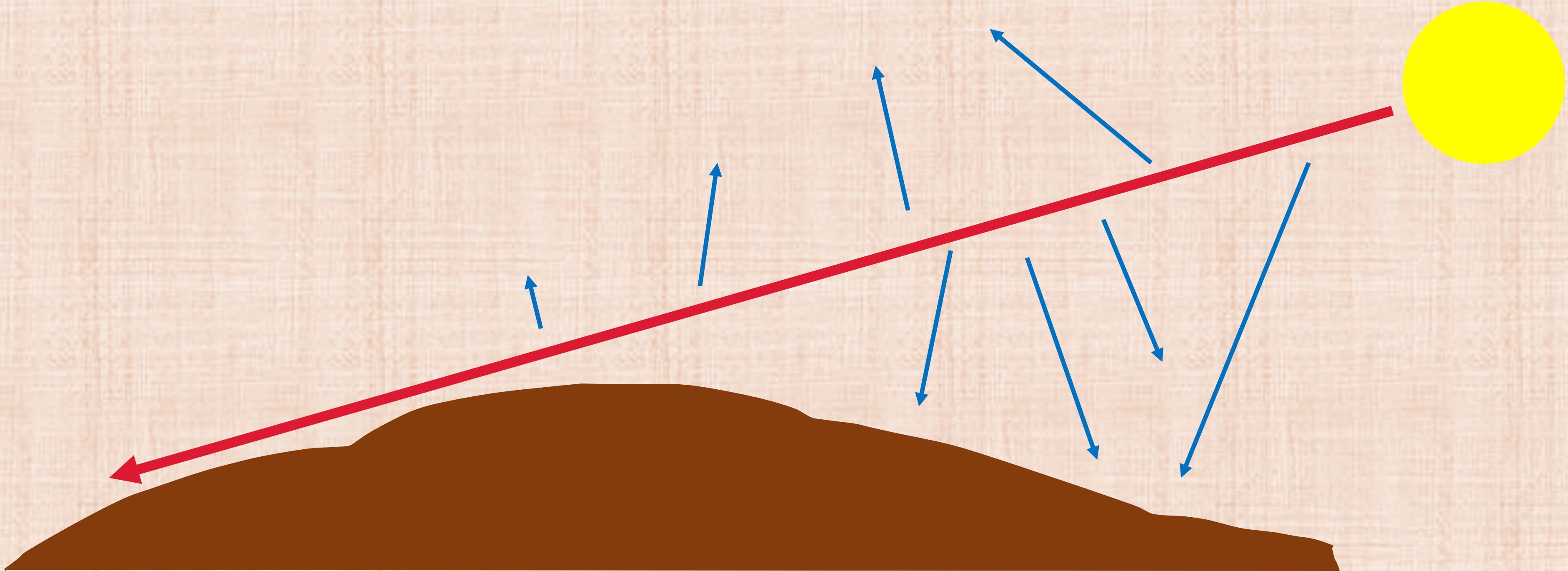
- While traveling through participating media, light can be absorbed (and converted into another form of non-visible energy, e.g. heat)
- As light moves a distance dx (along a ray), a fraction (absorption coefficient $\sigma_a(x)$) of the radiance $L(x, \omega)$ given by $\sigma_a(x)L(x, \omega)$ is absorbed: $dL(x, \omega) = -\sigma_a(x)L(x, \omega)dx$

Out-Scattering

- While traveling through participating media, light can be scattered off in various directions
- The atmosphere scatters blue light more readily than red light, which makes the sunset red (the light travels through a lot of atmosphere to reach our eyes)
- As light moves a distance dx (along a ray), a fraction (scattering coefficient $\sigma_s(x)$) of the radiance $L(x,\omega)$ given by $\sigma_s(x)L(x,\omega)$ is scattered off into another direction (and no longer travels along the ray): $\mathrm{d}L(x,\omega) = -\sigma_s(x)L(x,\omega)\mathrm{d}x$

Out-Scattering

- The atmosphere scatters blue light much more readily than red light
- This makes sunsets red
- The light travels through a lot of atmosphere to reach your eyes

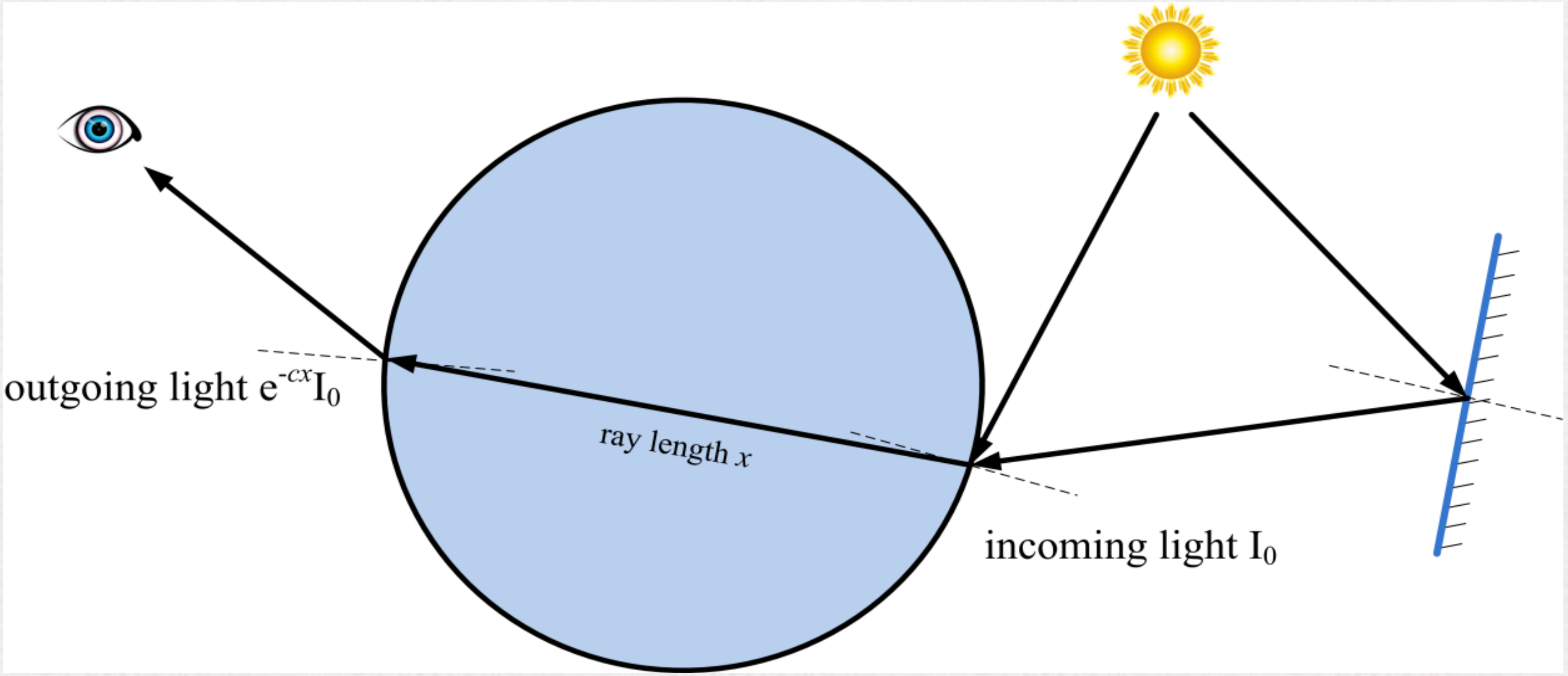


Total Attenuation

- The total fraction of light absorbed or out-scattered per unit length is: $c(x) = \sigma_a(x) + \sigma_s(x)$
- As light moves a distance dx (along a ray), a fraction of the radiance is attenuated (and no longer travels along the ray): $dL(x, \omega) = -c(x)L(x, \omega)dx$
- This affects all rays (primary rays from the camera, shadow rays, reflected/transmitted rays)

Recall: Beer's Law

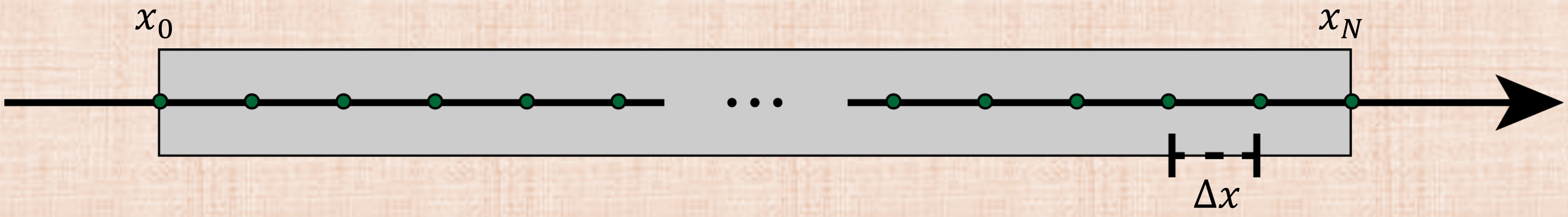
- For homogeneous media, attenuation can be approximated by Beer's Law
- Light with intensity I is attenuated over a distance x via the Ordinary Differential Equation (ODE): $\frac{dI}{dx} = -cI$ where c is the attenuation coefficient
- This ODE has an exact solution: $I(x) = I_o e^{-cx}$ where I_o is the original amount of light



Heterogeneous Beer's Law

- For non-homogeneous media, the attenuation coefficient $c(x)$ varies spatially (based on the concentration of the inhomogeneities)
- Discretize the ray into N smaller segments
- Treat c as a constant over each segment (converges as $N \to \infty$)
- Given $\Delta x = (x_N - x_0)/N$ and segment endpoints $x_i = x_0 + i\Delta x$ for $i \in [0, N]$, the total attenuation along the ray is:

$$I_o e^{-c\left(\frac{x_0+x_1}{2}\right)\Delta x} e^{-c\left(\frac{x_1+x_2}{2}\right)\Delta x} \cdots e^{-c\left(\frac{x_{N-1}+x_N}{2}\right)\Delta x}$$



Shadow Ray Attenuation

- Shadow rays cast from the ground plane to the light source have their light attenuated by the smoke volume
- This allows smoke to cast a shadow onto the ground plane
- The shadow is not completely black, since some light makes it through the smoke (to the other side)

Camera Ray Attenuation

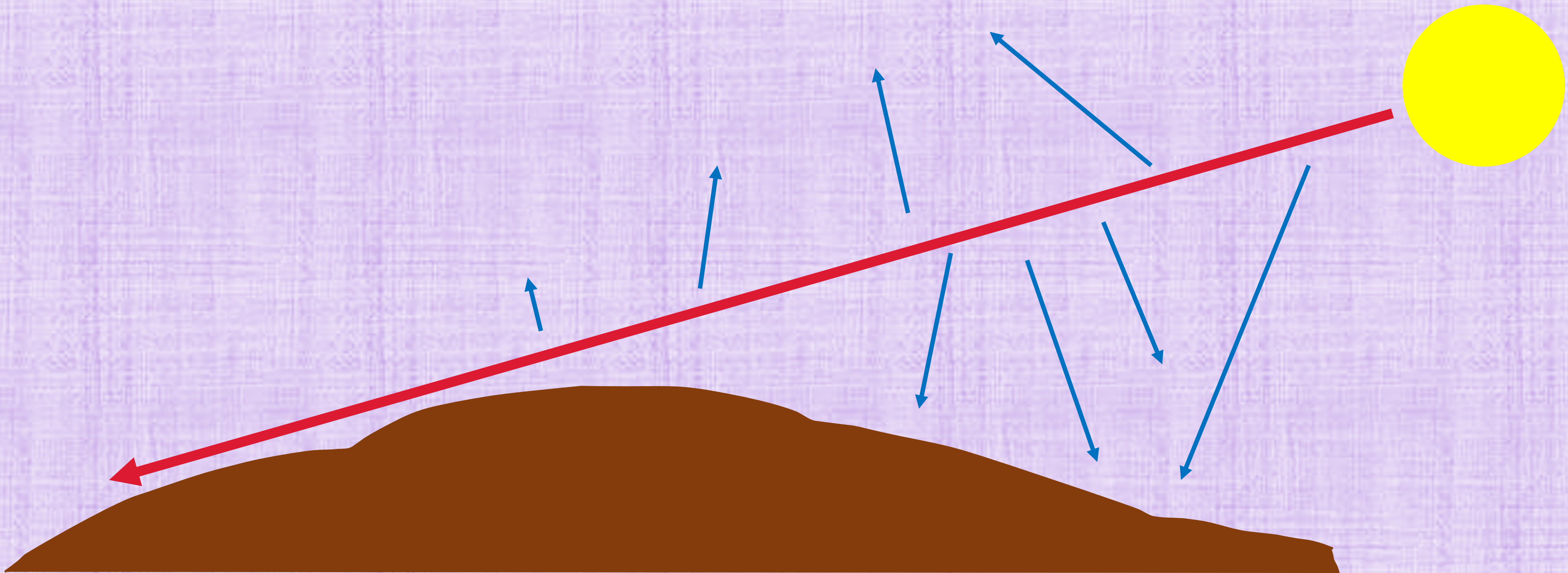
- Rays from the camera intersect objects, and a color is calculated (as usual, e.g. blue here)
- That color is attenuated by the participating media intersecting the ray
- The object color could be partially or completely attenuated
- Complete attenuation would lead to black pixels, if the smoke itself were colorless

In-Scattering

- At each point along a ray, participating media can out-scatter light traveling in other directions
- Some of that out-scattered light could be in-scattered into the ray direction
- This in-scattering <u>increases</u> the radiance along the ray
- The sky appears blue because atmospheric particles scatter blue light in every direction, and some of it is scattered towards your eyes (otherwise, the sky would appear black)

In-Scattering

- The atmosphere scatters blue light much more readily than red light
- Some of it is scattered towards your eyes, making the sky appear blue (instead of black)



In-Scattering

- Add the radiance contribution from in-scattering to the color of the rays from cameras to objects (as well as to shadow rays)
- Without in-scattering, complete attenuation of object color (by participating media) results in black pixels
- In-scattered light gives participating media its own appearance (clouds, smoke, etc.)
- The darker underside of a cloud has less light available to in-scatter, because the top of the cloud absorbs and out-scatters much of the light (from the sun)

Volumetric Light Map

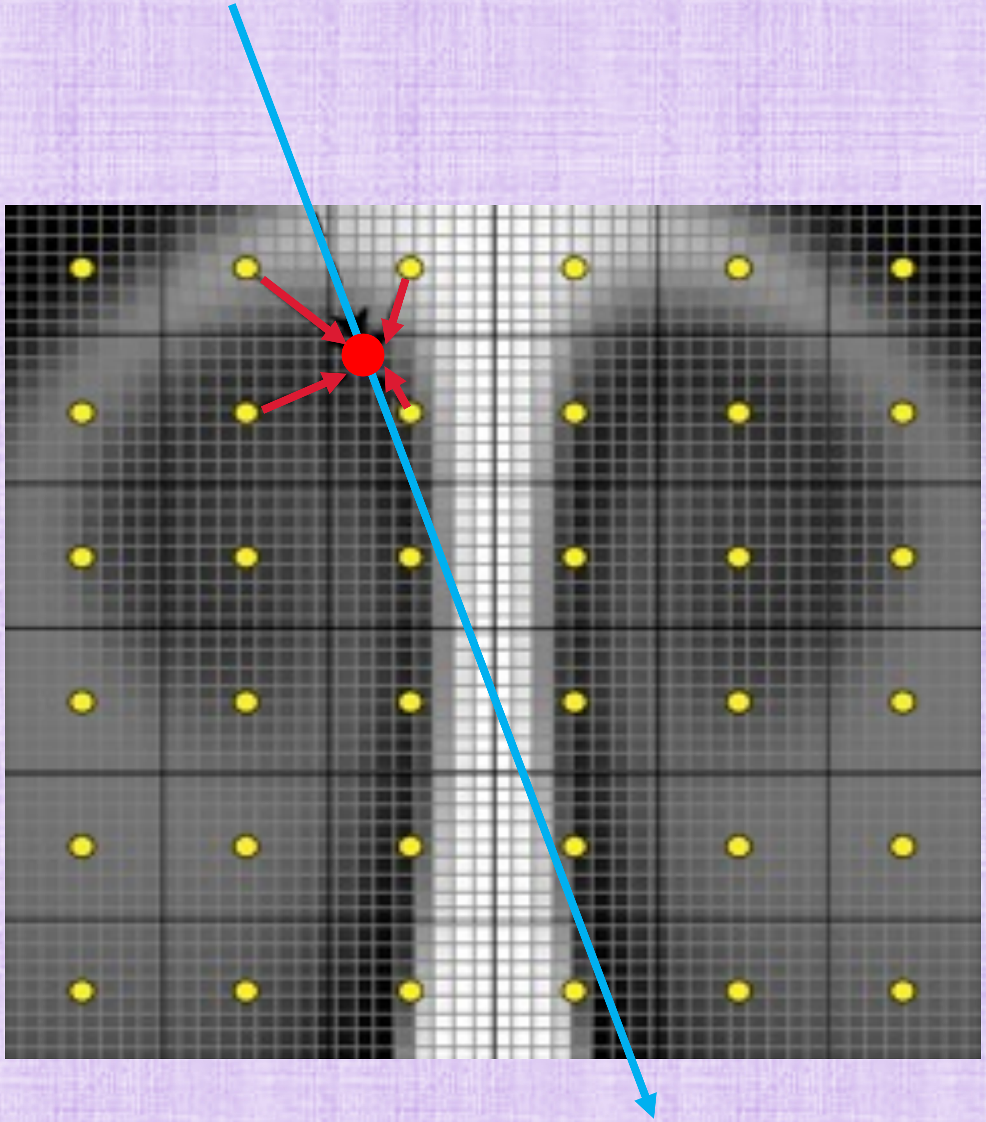
- At each of many sample points along a ray, cast a shadow ray to the light source to compute how much light is available for in-scattering
- These shadow rays are expensive to compute, since they use inhomogeneous Beer's law to attenuate light with the participating media along the ray

- For efficiency, precompute a volumetric light map:
  - Enclose the participating media with a uniform grid (or octree, or other spatial partition)
  - At each grid point, cast a shadow ray to the light source to precompute how much light is available for in-scattering

- Later, when tracing camera/shadow rays, use the volumetric light map to determine how much light is available for in-scattering (along each segment of any ray passing through it)
- Add in-scattered light to the total light at each point (noting that it too gets attenuated on subsequent segments along the ray)
  - Thus, this calculation needs to be done from object to camera

In-Scattering (with a Volumetric Light Map)

- At the midpoint of each segment of the discretized ray, interpolate available radiance $L(x, \omega)$ from the volumetric light map
- Compute the incoming direction ω from the light source to the interpolation point (a separate volumetric light map is required for each light source)
- A <u>phase function</u> $p(\omega, \omega')$ gives the probability that incoming light from direction ω is scattered into direction ω' of the camera/shadow ray
- The radiance at this point x scattered into the ray direction is $p(\omega, \omega')\sigma_s(x)L(x, \omega)$
  - σ_s is the probability of <u>any</u> scattering in <u>any</u> direction, and p selects the subset that scatters into the ray direction
- The entire in-scattered radiance from a segment of length Δx is $p(\omega, \omega')\sigma_s(x)L(x, \omega)\Delta x$



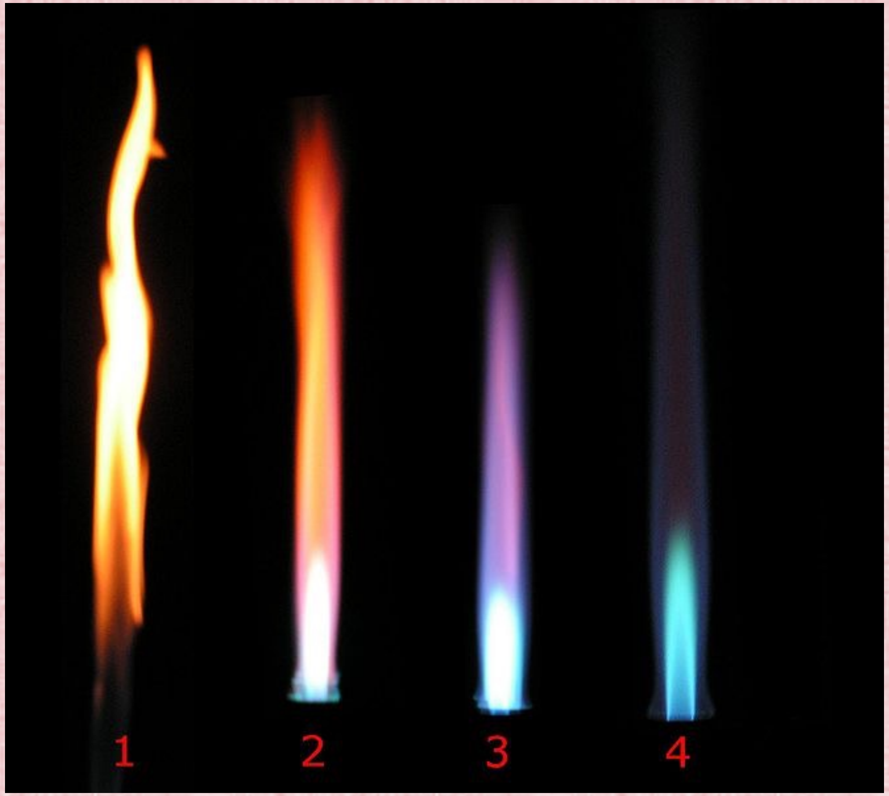
Phase Functions

- Everything goes somewhere: $\int_{\mathrm{sphere}} p(\omega, \omega')\mathrm{d}\omega' = 1$
- Phase angle: $\cos\theta = \omega \cdot \omega'$

1. Isotropic: $p(\cos\theta) = \frac{1}{4\pi}$
2. Rayleigh: $p(\cos\theta) = \frac{3}{8}(1 + \cos^2\theta)$
   - Models scattering due to particles smaller than the wavelength of light, such as in the atmosphere
3. Henyey-Greenstein: $p(\cos\theta) = \frac{\frac{1}{4\pi}(1-g^2)}{(1+g^2-2g\cos\theta)^{1.5}}$
   - g can be treated as a tunable parameter, which allows one to adjust the appearance of a medium
   - $g = 0$ results in the isotropic phase function

Volumetric Emission

- Participating media emit light
  - Hot carbon soot emits blackbody radiation (based on temperature)
  - Electrons emit light energy as they fall from higher energy excited states to lower energy states
- This light information can be added as a separate volumetric light map
- This volumetric emission is in <u>every</u> direction



Volumetric Emission

- Adding volumetric emission to the light map gives the desired orange/blue/etc. colors
- But only adding it to the light map doesn't allow it to cast shadows and light the scene
- To do this, treat this region as a volume light
  - Model a volume light with many small point lights (similar to an area light)
  - These point lights are used just like every other light in the scene: shadow rays, creating photon maps, etc.
  - They also participate in the creation of the volumetric light map (for self shadowing of participating media)