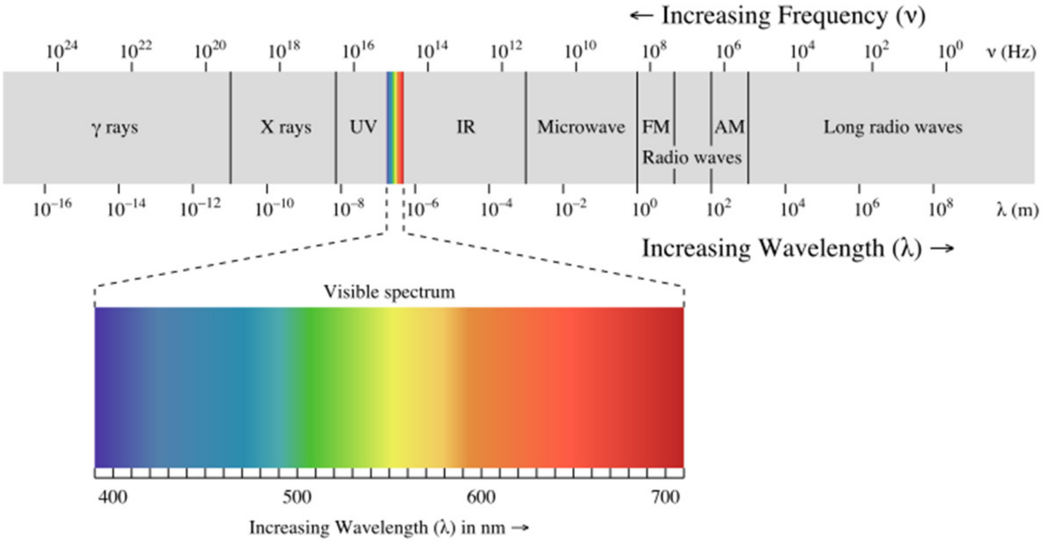
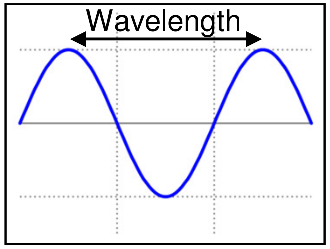
CSC 321 Computer Graphics

Illumination

What is light?

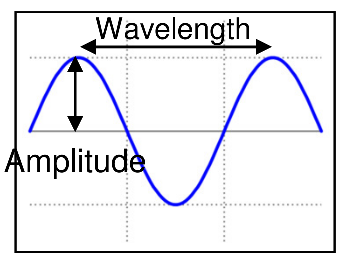
- Electromagnetic radiation that is visible to the human eye
  - Carried by "photons" but also exhibits wave behaviors
- Properties
  - Speed (constant)
  - Direction (straight)
  - Wavelength ("color")

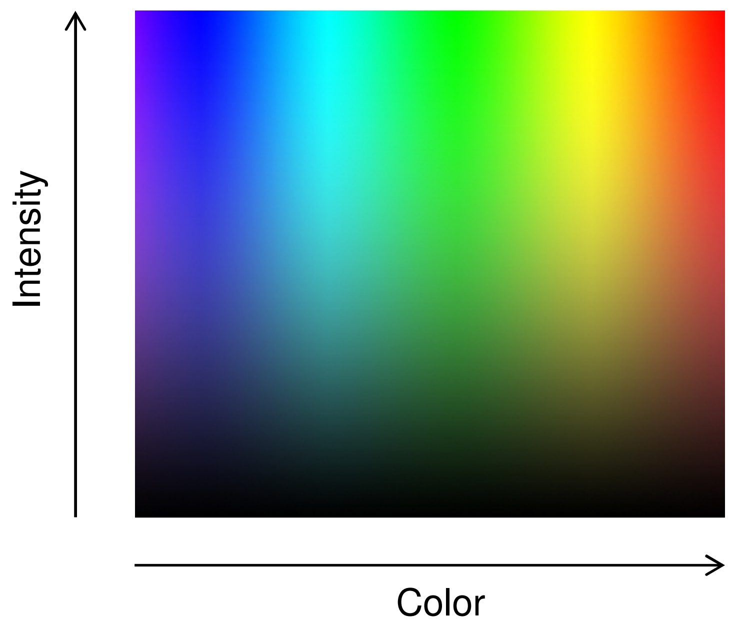




What is light?

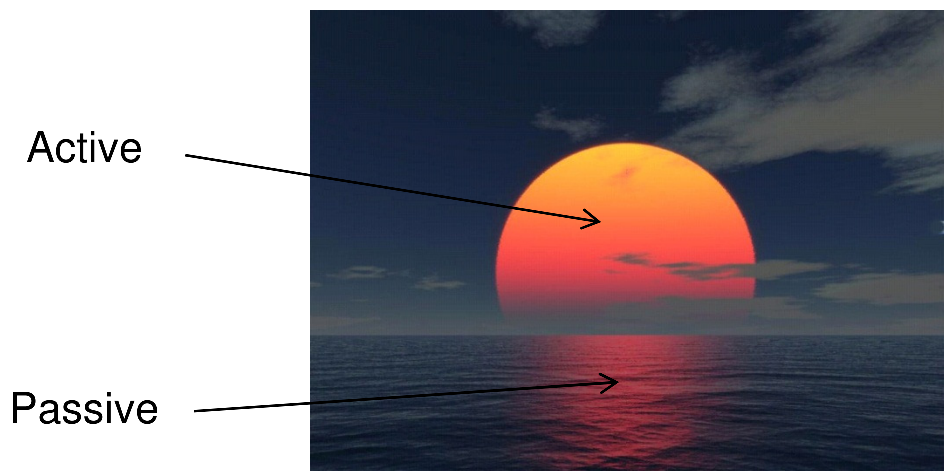
- Electromagnetic radiation that is visible to the human eye
  - Carried by “photons” but also exhibits wave behaviors
- Properties
  - Speed (constant)
  - Direction (straight)
  - Wavelength (“color”)
  - Amplitude (“intensity”)





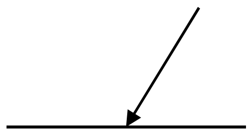
How is a surface lit?

- By the (color and intensity of) light that is transmitted from the surface in the direction towards our eye
  - Active: the surface is emitting light (i.e., a light source)
  - Passive: the light originates from somewhere else



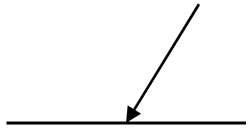
Passive Lighting

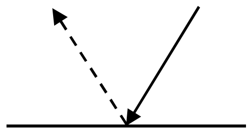
- When a photon hits a surface it may
  - Get absorbed (turned into heat or other energy)

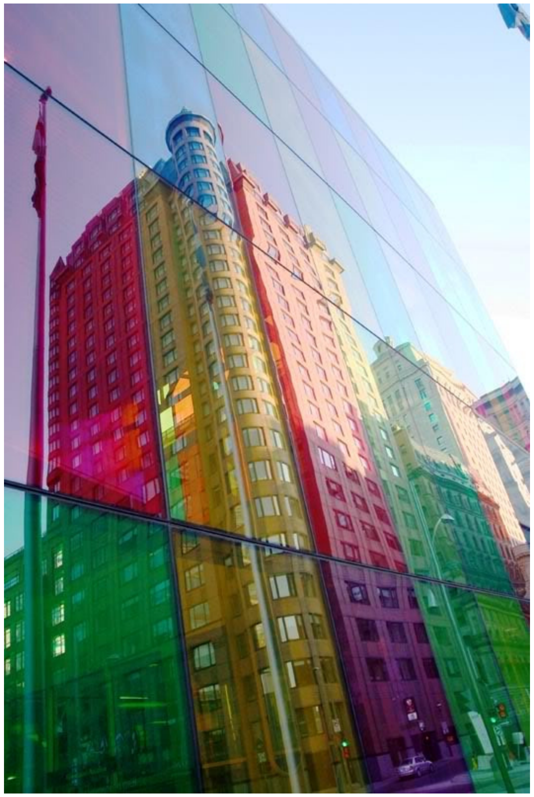


Passive Lighting

- When a photon hits a surface it may
  - Get absorbed
  - Reflect

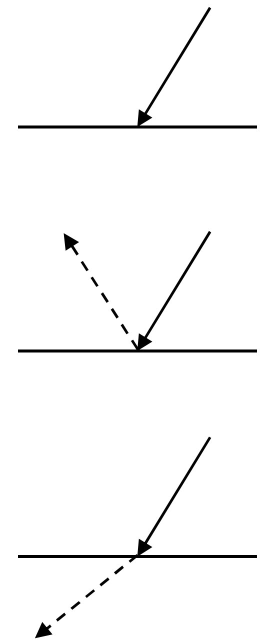






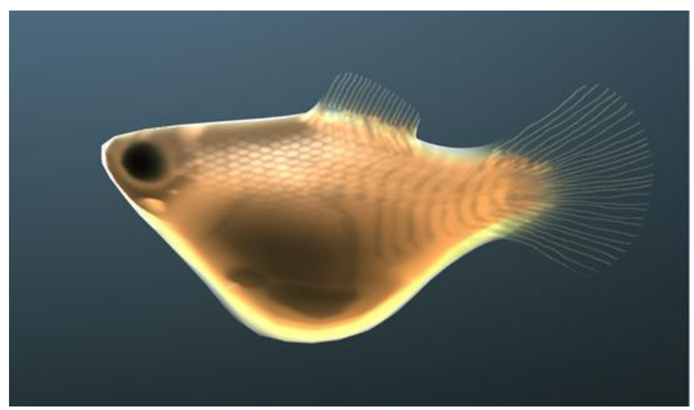
Passive Lighting

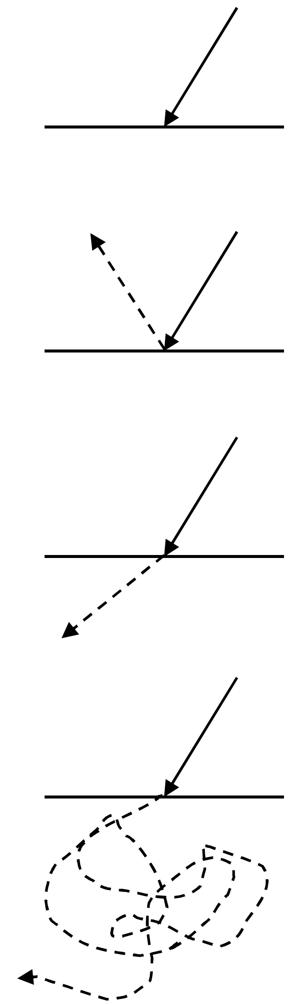
- When a photon hits a surface it may
  - Get absorbed
  - Reflect
  - Refract



Passive Lighting

- When a photon hits a surface it may
  - Get absorbed
  - Reflect
  - Refract
  - Scatter



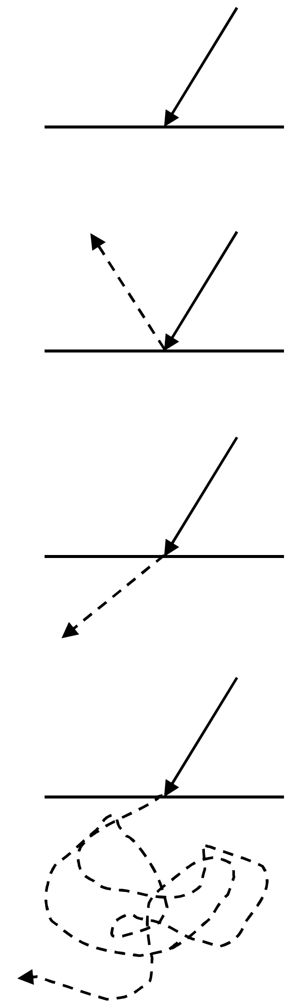


Passive Lighting

- When a photon hits a surface it may
  - Get absorbed
  - Reflect
  - Refract
  - Scatter

  (Reflect, Refract, Scatter:) May change direction, color, and intensity

- What happens depends on:
  - The color and intensity of light
  - Color and material of surface
    - Reflective? Diffusive? Transparent? Opaque? Translucent?
  - Orientation of surface with respect to the light source and eye

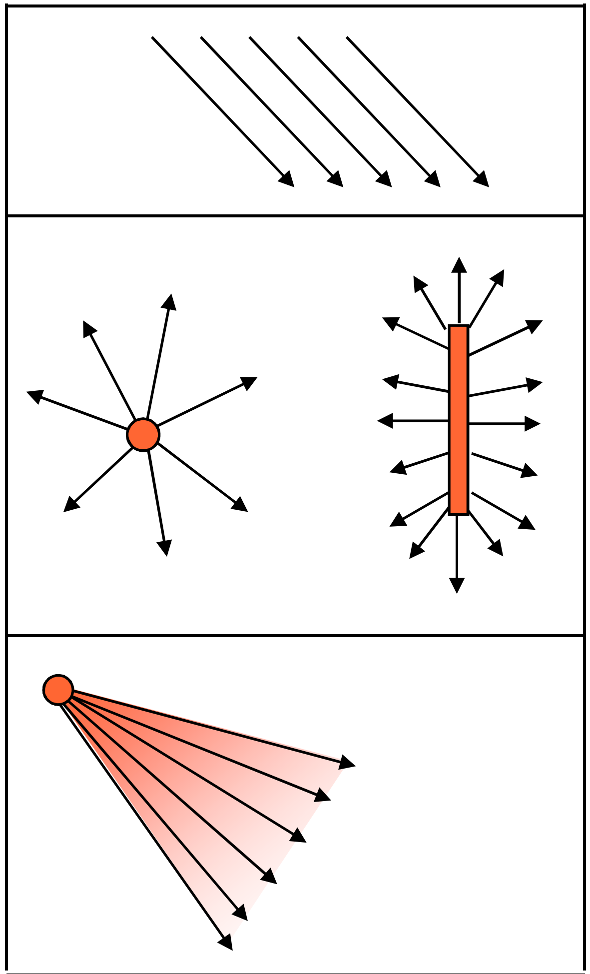


Illumination (in Computer Graphics)

- Given
  - Light sources, object surfaces and the camera
- Compute
  - Color of each pixel on the screen
    - As intensity of photons that come towards the camera in that viewing direction

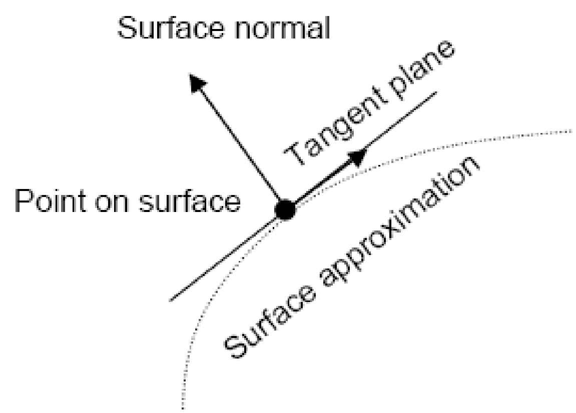
Computer Representation

- Light sources
  - Directional light (e.g., the sunlight)
    - Emitting photons in one direction
  - Point/Area light (e.g., light bulb)
    - Emitting photons in all directions from a single source
  - Spot light (e.g., a flashlight)
    - Emitting photons from a single source forming a cone
- Light attenuation
  - Intensity falls off with distance
    - Not applicable to directional light



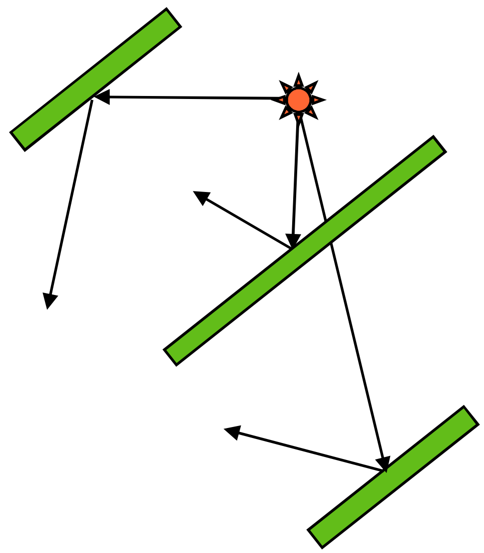
Computer Representation

- Surface
  - A geometric surface can be
    - Discrete: consisting of polygons (e.g., triangles), or
    - Continuous: parametric surface (e.g., the sphere)
  - Each surface element is locally represented by
    - The point location
    - The normal vector of the tangent plane
      - Discrete: polygon normal
      - Continuous: first derivatives



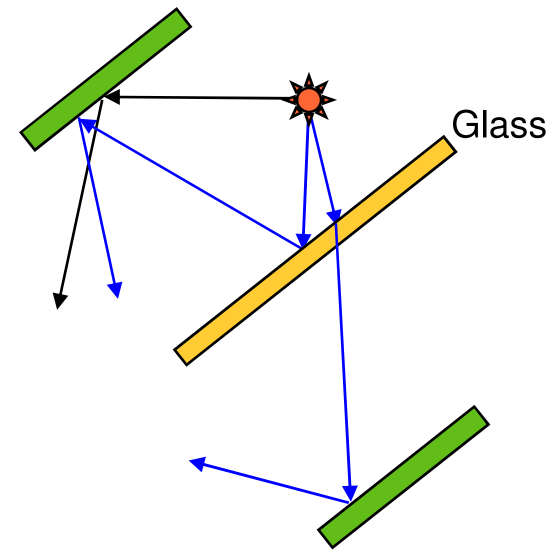
Local Illumination

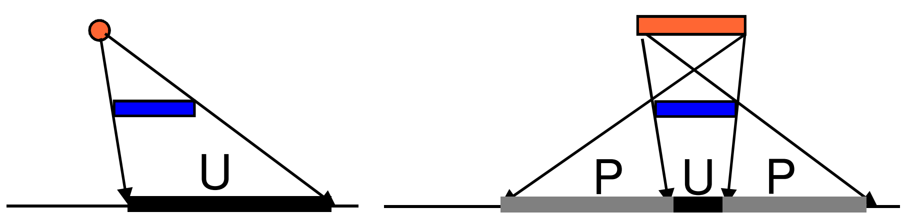
- Light interaction with one surface
  - Only considers direct reflection of the light from the source
  - Assuming the path between the light source and the surface is unblocked
- Pros: Fast
- Cons: Missing many effects
  - Shadow
  - Refraction
  - Multi-hop reflection



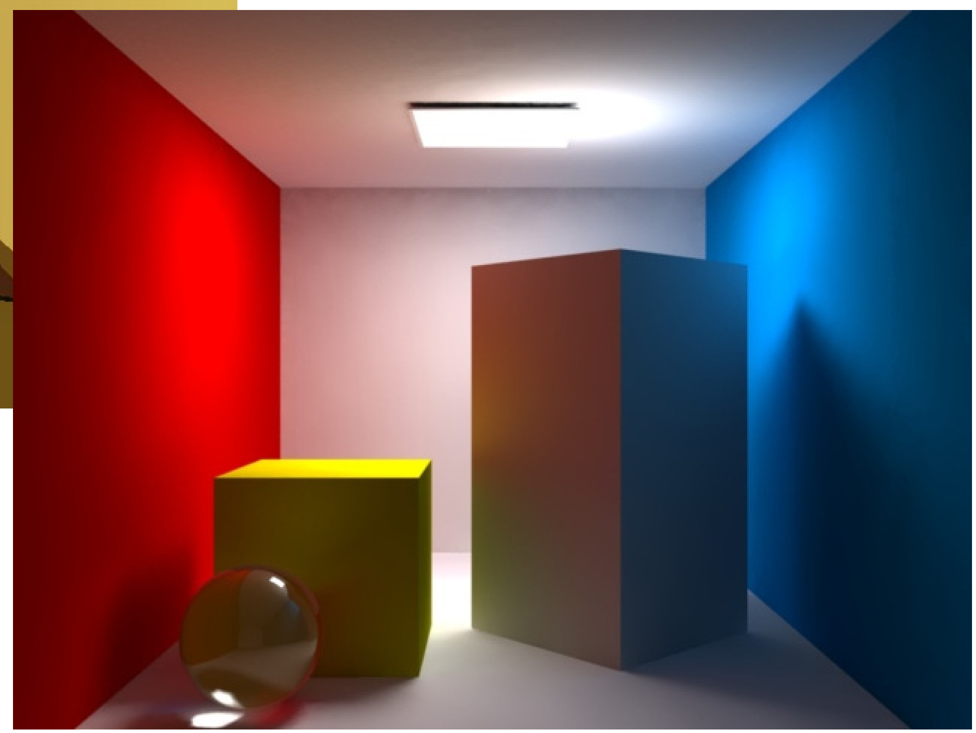
Global Illumination

- Light interaction with all surfaces
  - Reflection/refraction involving multiple surfaces
  - Considers shadows (when the path between the light source and surface is blocked)
    - Complete (*Umbra*) or incomplete (*Penumbra*)
- Pros: Realistic
- Cons: Expensive





Local vs. Global Illumination



Overview

- Local illumination
  - Local lighting model (Today)
  - Drawing polygonal models (Thursday)

- Global illumination
  - Ray tracing (Friday)
  - Radiosity

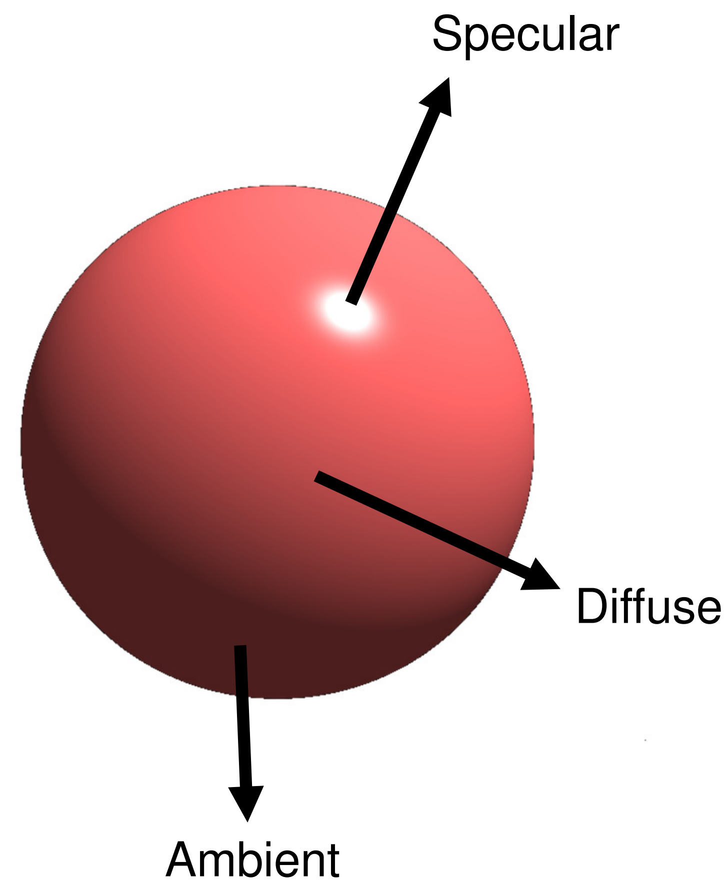
Local Lighting Model

- Factors in computing reflected light:
  - Geometric configuration (between light source, surface and camera)
  - Light properties (source type, color, attenuation)
  - Surface material (color, shininess, etc.)
  - Others (polarization, fluorescence, phosphorescence, etc.)

- Lighting model: the math that computes reflected light
  - Physical model
    - Computes actual energy transmitted, very expensive
  - Non-physical model (OpenGL)
    - “Close enough”, “looks good”, but fast

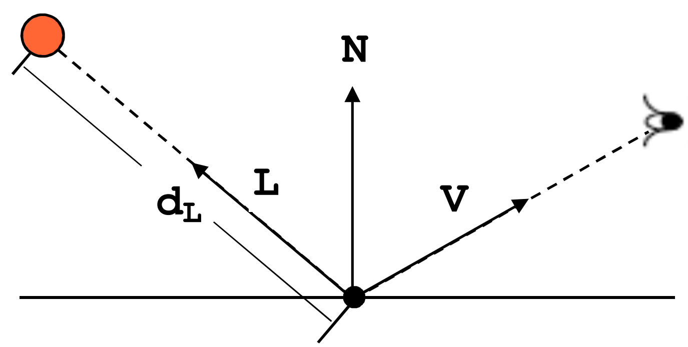
Local Lighting Model

- Point light source
- Sum of three terms
  - Diffuse light
    - Diffusive reflection
  - Specular light
    - Highlights
  - Ambient light
    - Global, environment light



Parameters

- **Geometry**
  - Surface normal (unit vec): $\mathbf{N} = \{N_x, N_y, N_z\}$
  - Direction to light source (unit vec): $\mathbf{L} = \{L_x, L_y, L_z\}$
  - Distance to light source: d_L
  - Direction to camera: $\mathbf{V} = \{V_x, V_y, V_z\}$



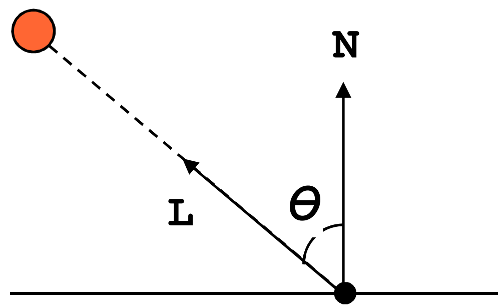
- **Light properties**
  - Color/Intensity: $I_L = \{I_L^r, I_L^g, I_L^b\}$
  - Ambient light: $I_A = \{I_A^r, I_A^g, I_A^b\}$
  - Attenuation coefficients: c_0, c_1, c_2
- **Surface material**
  - Diffuse coefficients: $k_d = \{k_d^r, k_d^g, k_d^b\}$
  - Specular coefficients: $k_s = \{k_s^r, k_s^g, k_s^b\}$ and exponent: n
  - Ambient coefficients: $k_a = \{k_a^r, k_a^g, k_a^b\}$

Diffuse Reflection

- Simulates reflection on matte surfaces
  - Independent of view direction
- Lambert's Cosine Law

$$I_{diff} = I_L k_d \cos[\theta] = I_L k_d (N \cdot L)$$

$$(\theta < \pi / 2, \text{ or } N \cdot L \geq 0)$$

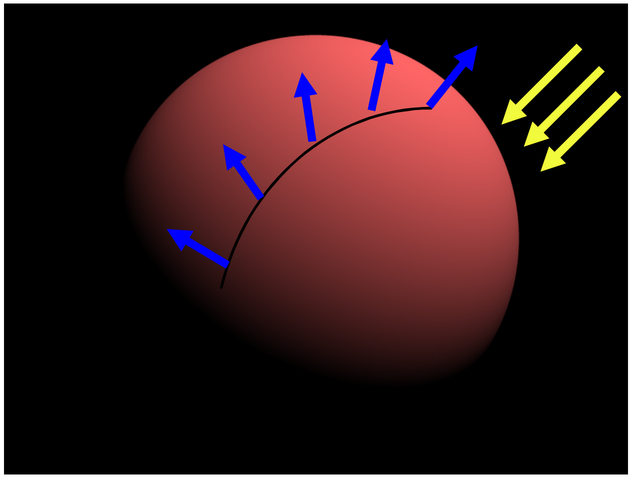


  - Compute for each color component :

$$I^r_{diff} = I^r_L k^r_d (N \cdot L)$$

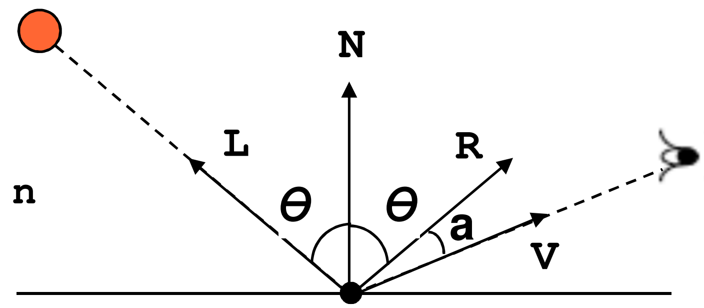
$$I^g_{diff} = I^g_L k^g_d (N \cdot L)$$

$$I^b_{diff} = I^b_L k^b_d (N \cdot L)$$



Specular Reflection

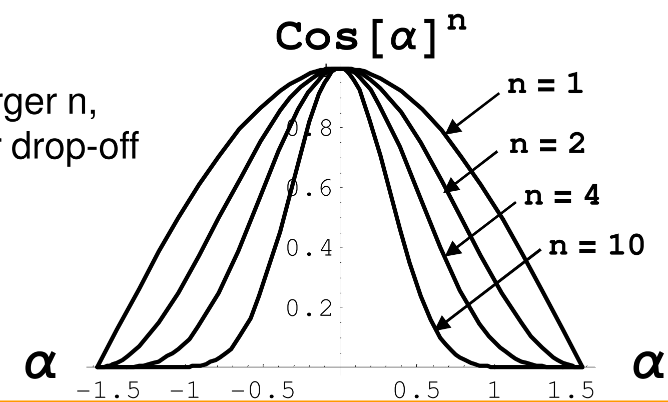
- Simulates highlight on shiny surfaces
  - Dependent on the viewing direction
- Phong's approximation

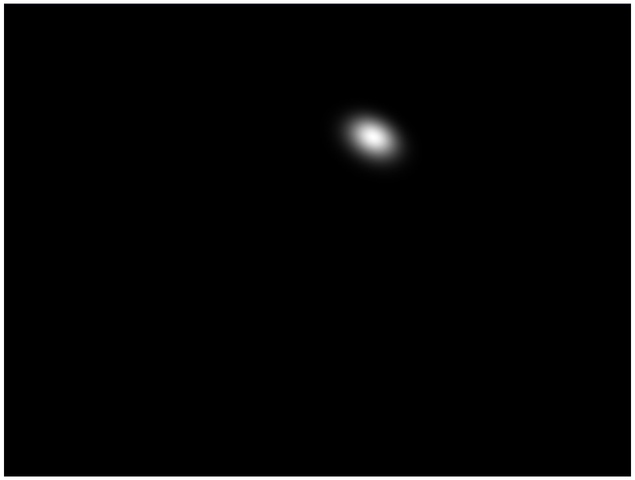


$$I_{spec} = I_L k_s \cos[\alpha]^n = I_L k_s (R \cdot V)^n$$

$$(N \cdot L \geq 0)$$

  - R: reflected light direction
  - n: specular exponent

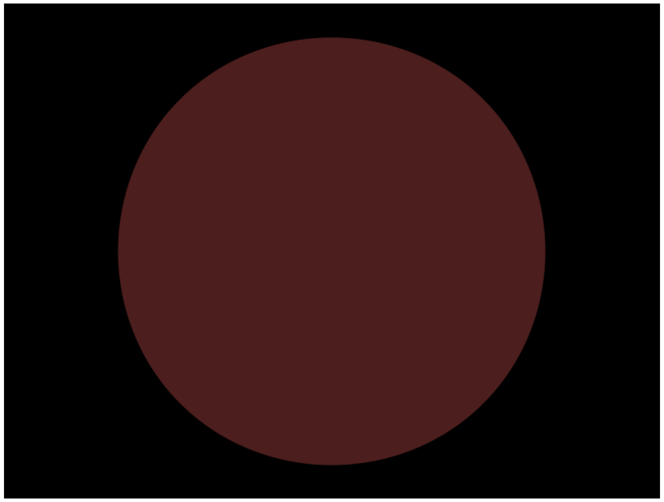




Ambient Reflection

- Simulates global illumination
  - Lights bounced off other objects
- Constant light (a simple hack)

$$I_{amb} = I_A k_a$$



Light Attenuation

- Simulates decrease of light energy over distance

$$I_L \leftarrow f_{att}\, I_L$$

  - Does not affect ambient light

- Inverse square law of energy fall-off

$$f_{att} = \frac{1}{d_L^{\,2}}$$

- In practice

$$f_{att} = \frac{1}{c_0 + c_1\, d_L + c_2\, d_L^{\,2}}$$

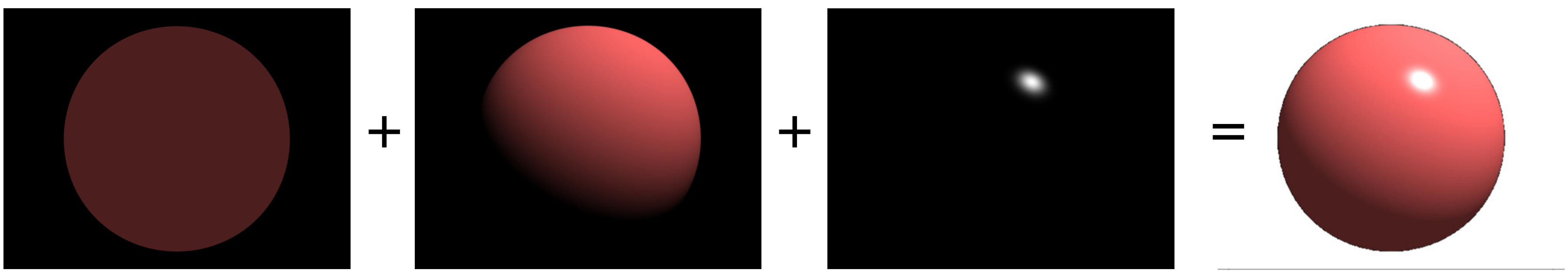
Putting Together

- Local (OpenGL) lighting model

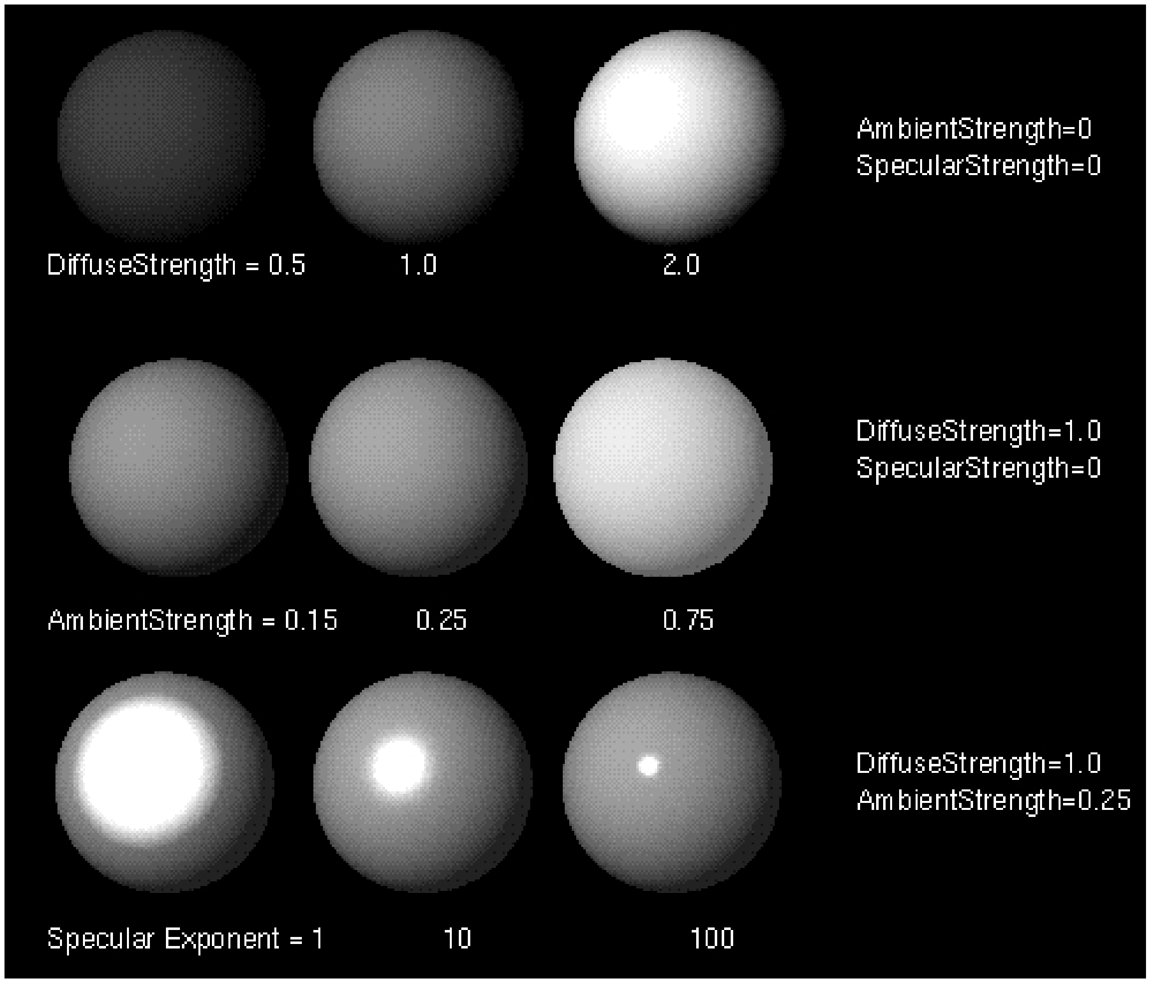
$$I = I_{amb} + I_{diff} + I_{spec}$$

$$= I_A k_a + I_L f_{att} (k_d (N \cdot L) + k_s (R \cdot V)^n)$$

  – Compute for each color component

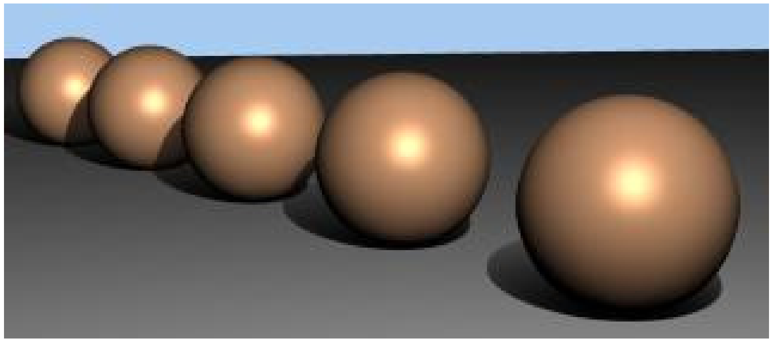


Example: Varying Parameter

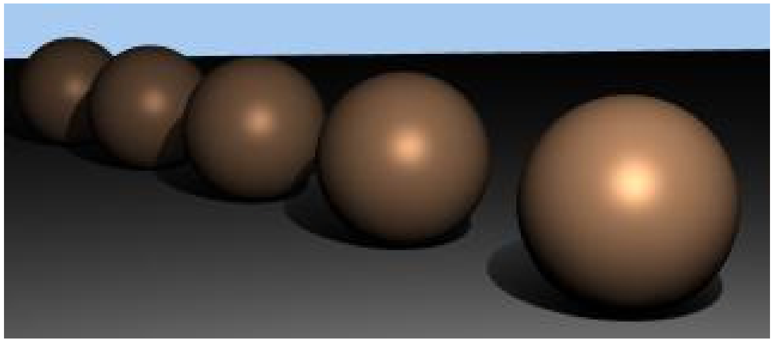


Example: Attenuation

No attenuation:



Linear attenuation:



Quadratic attenuation: