15-462 Computer Graphics I

Lecture 7

Lighting and Shading

Properties of Light
Light Sources
Phong Illumination Model
Normal Vectors

[Angel, Ch. 6.1-6.4]

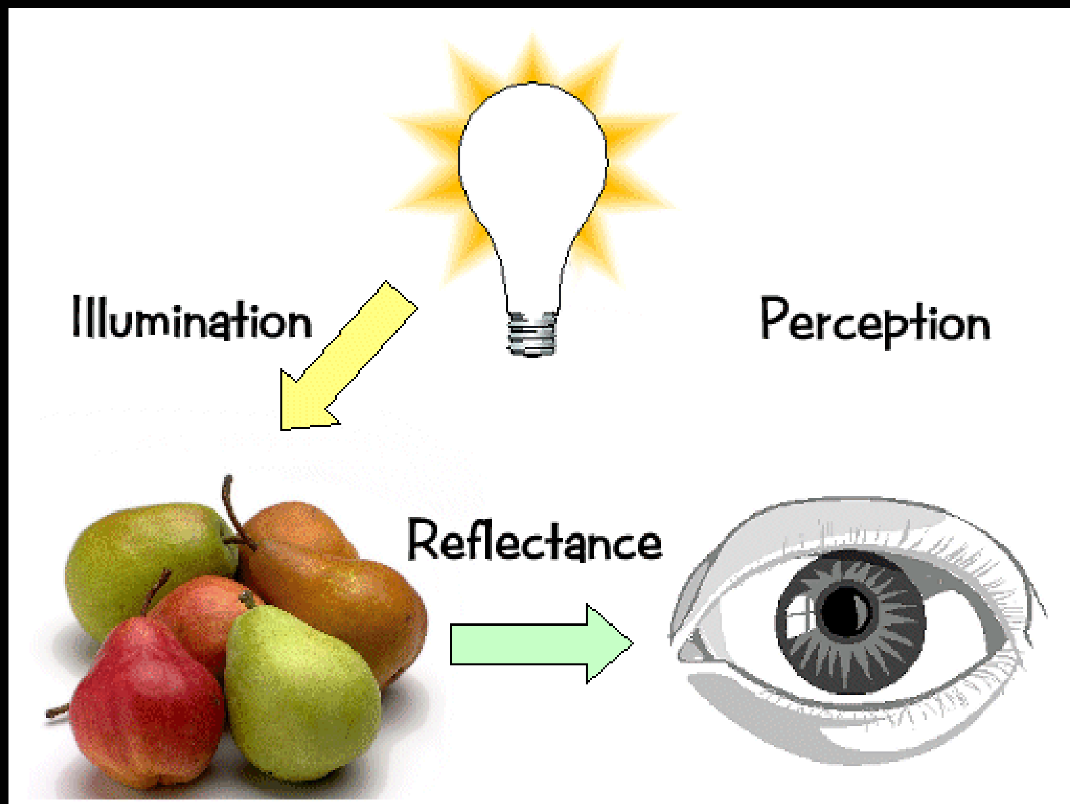
September 16, 2003

Doug James
Carnegie Mellon University

Announcements

- Programming assignment #1 due today
  - Electronic handin by midnight
- Written assignment #1 due Thursday
  - Handin at beginning of class
- Programming assignment #2 out Thursday
  - Due in 2 weeks

Light Transport



Lighting and Shading

- Approximate physical reality
- Ray tracing:
  - Follow light rays through a scene
  - Accurate, but expensive (off-line)
- Radiosity:
  - Calculate surface inter-reflection approximately
  - Accurate, especially interiors, but expensive (off-line)
- Phong Illumination model (this lecture):
  - Approximate only interaction light, surface, viewer
  - Relatively fast (on-line), supported in OpenGL

Radiosity Example

Restaurant Interior. Guillermo Leal, Evolucion Visual

Raytracing Example

Martin Moeck,
Siemens Lighting

Outline

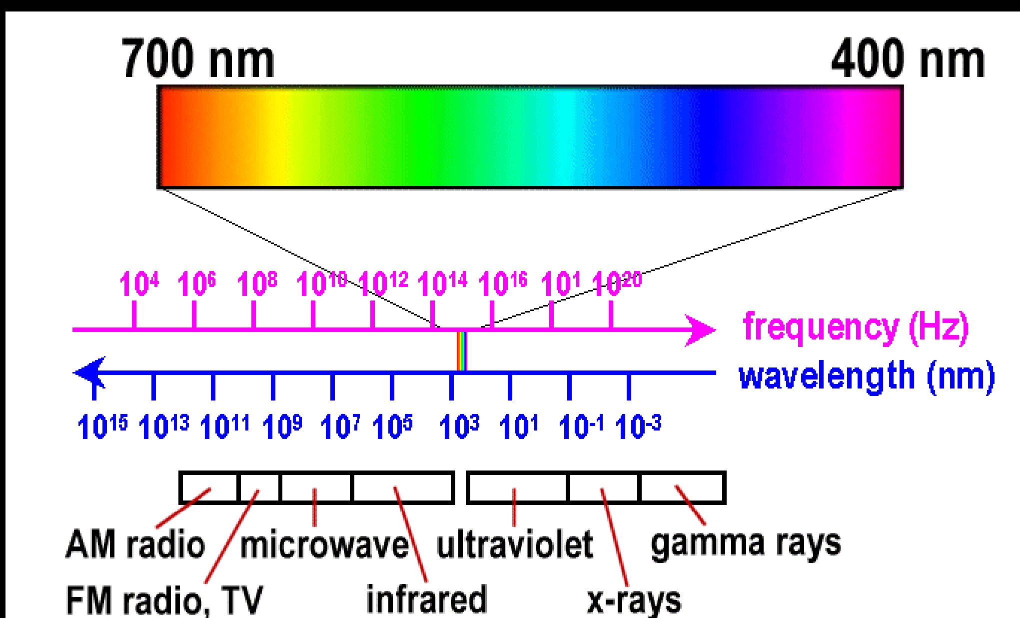
- Properties of Light
- Light Sources
- Phong Illumination Model
- Normal Vectors

Outline

- Properties of Light
  - Real light
  - How humans see light
  - How computers trick humans into thinking they're seeing light
- Light Sources
- Phong Illumination Model
- Normal Vectors

Physics of Light and Color

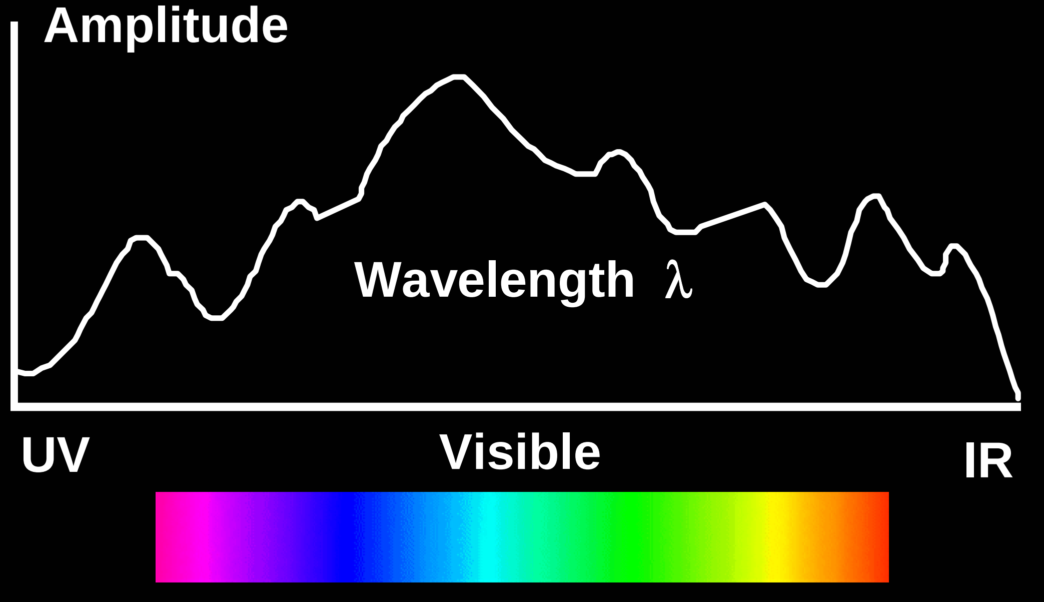
- It's all electromagnetic (EM) radiation
  - Different colors correspond to different ***wavelengths*** λ
  - Intensity of each wavelength specified by ***amplitude***
  - ***Frequency*** $\nu = 2\pi / \lambda$
    - long wavelength is low frequency
    - short wavelength is high frequency



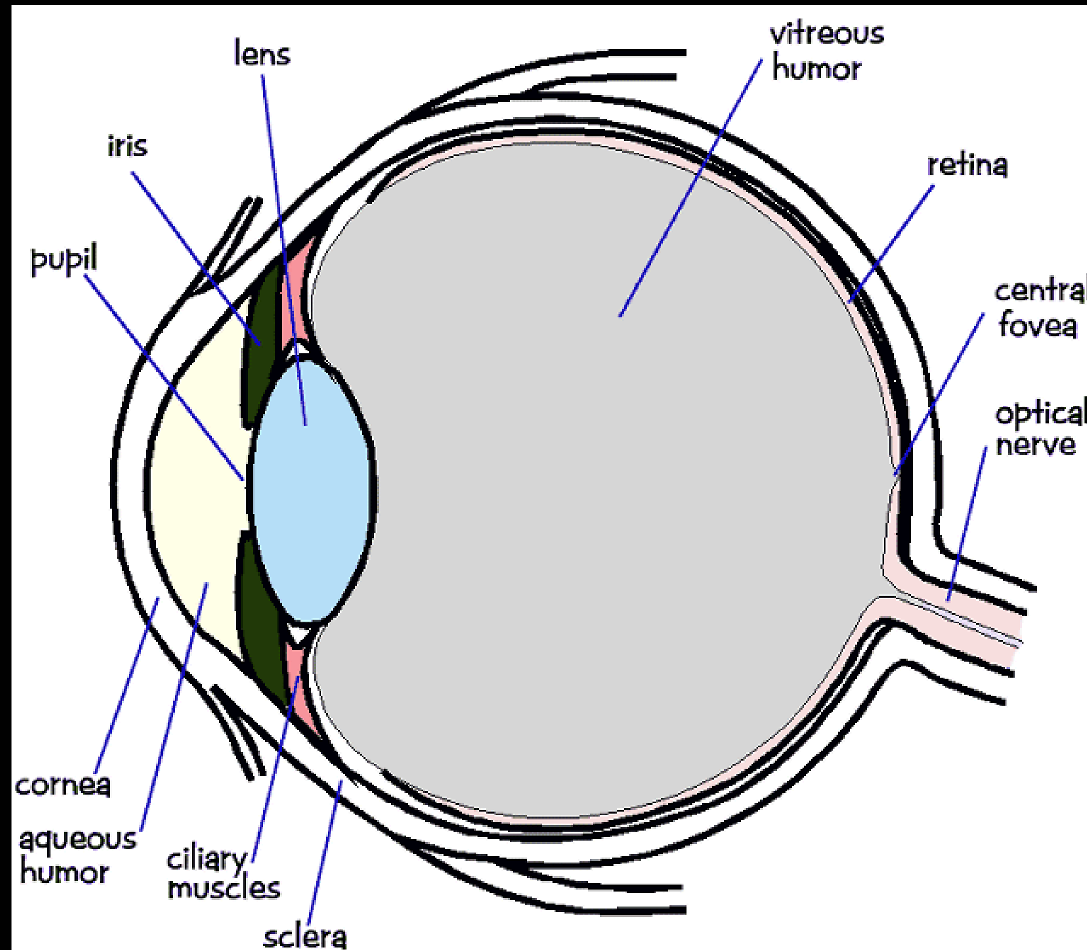
We perceive EM radiation with λ in the 400-700 nm range

Color: What's There vs. What We See

- Human eyes respond to “visible light”
  - tiny piece of spectrum between infra-red and ultraviolet
- Color defined by emission spectrum of light source
  - amplitude vs wavelength (or frequency) plot

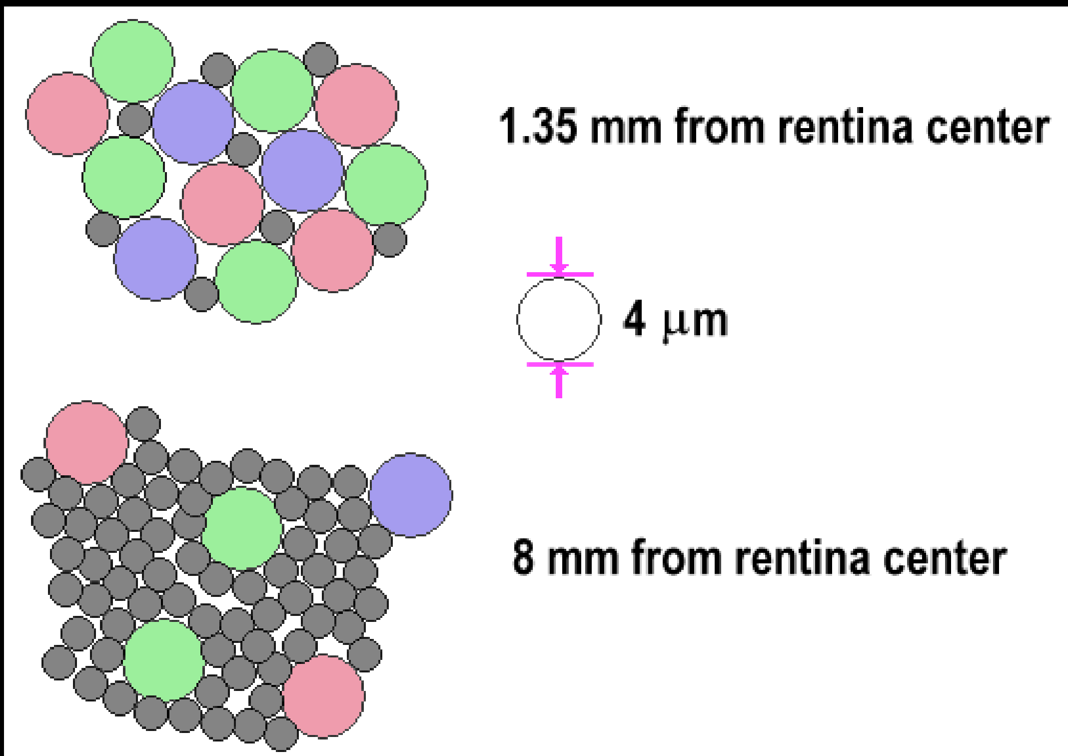


The Eye

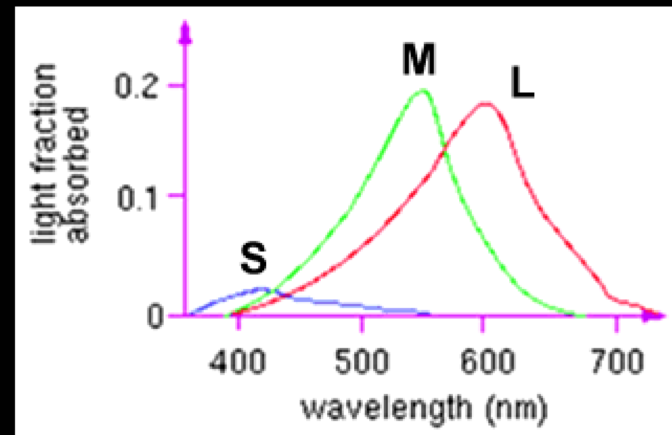


- The image is formed on the ***retina***
- Retina contains two types of cells: ***rods*** and ***cones***
- Cones measure color (red, green, blue)
- Rods responsible for monochrome night-vision

The Fovea



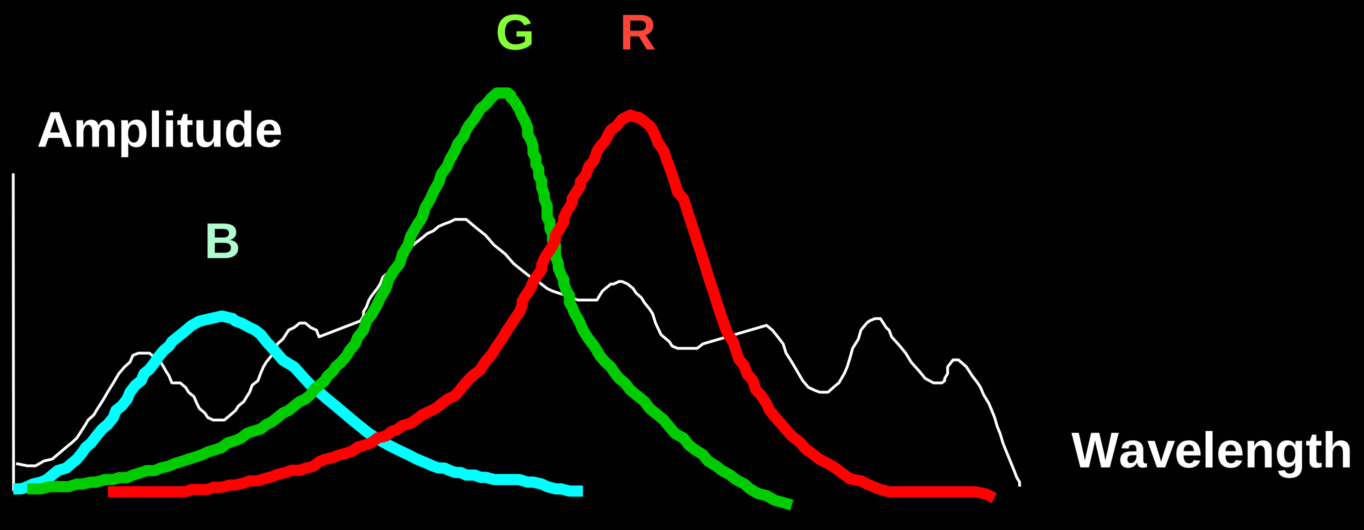
Cones are most densely packed within a region of the retina called the *fovea*



- Three types of cones: S,M,L
  - Corresponds to 3 visual pigments
- Roughly speaking:
  - S responds to blue
  - M responds to green
  - L responds to red
- Note that these are not uniform
  - more sensitive to green than red
- Colorblindness
  - deficiency of one cone/pigment type

Color Filters

- Rods and cones can be thought of as filters
  - Cones detect red, green or blue parts of spectrum
  - Rods detect average intensity across spectrum
- To get the output of a filter
  - Multiply its response curve by the spectrum, integrate over all wavelengths
- A physical spectrum is a complex function of wavelength
  - But what we see can be described by just 3 numbers—the color filter outputs
  - How can we encode a whole function with just 3 numbers?
    - A: we can't! We can't distinguish certain colors--*metamers*



Vision and Thought are One

- The retina is part of the central nervous system
- 2 million fibers from retina to *LGN*, 10 million from there to brain.
- Primary connection is *Primary Visual Cortex* or *V1,* 2 cm^2 on back of brain
  - Hypothesis: V1 gets used as a sort of image buffer for higher processing in the rest of the brain
- Steps:
  1. Saccade ends
  2. Retina accumulates image
  3. LGN opens connections, image gets written to V1
  4. Rest of brain accesses that info
  5. Meanwhile, a point of interest is being generated for next saccade
  6. Next saccade happens perhaps 250ms later; go back to step 1
- All very automatic; that's why pointing with eyes doesn't work for user interfaces.

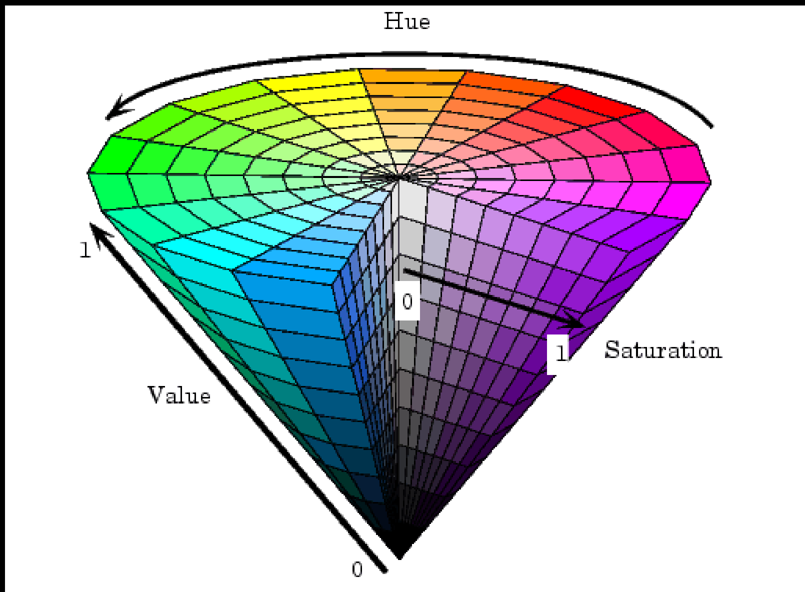
Color Models

- Okay, so our visual system is quite limited
- But maybe this is good news. . .
- We can avoid computing and reproducing the full color spectrum since people only have 3 color channels
  - TV would be much more complex if we perceived the full spectrum
    - transmission would require much higher bandwidths
    - display would require much more complex methods
  - real-time color 3D graphics is feasible
  - any scheme for describing color requires only three values
  - lots of different color spaces--related by 3x3 matrix transformations

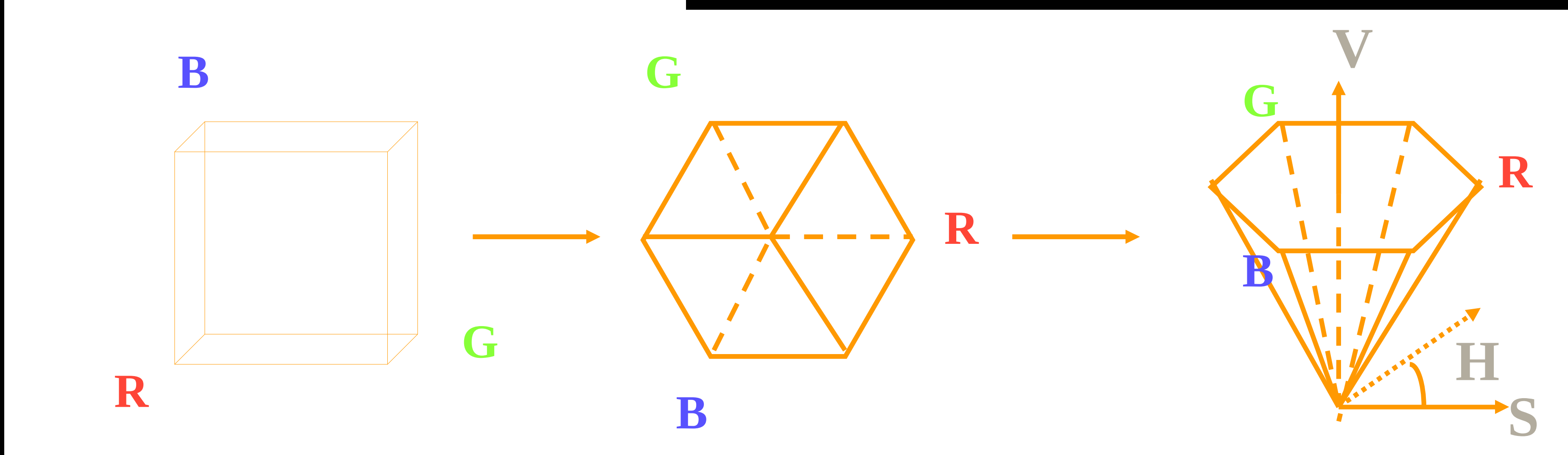
Color Spaces

- There are many ways to describe color
  - Spectrum
    - allows any radiation (visible or invisible) to be described
    - usually unnecessary and impractical
  - RGB
    - convenient for display (CRT uses red, green, and blue phosphors)
    - not very intuitive
  - HSV
    - an intuitive color space
    - H is hue - what color is it? S is saturation or purity - how non-gray is it? V is value - how bright is it?
    - H is cyclic therefore it is a non-linear transformation of RGB
  - CIE XYZ
    - a linear transform of RGB used by color scientists

HSV



From mathworks



Additive vs. Subtractive Color

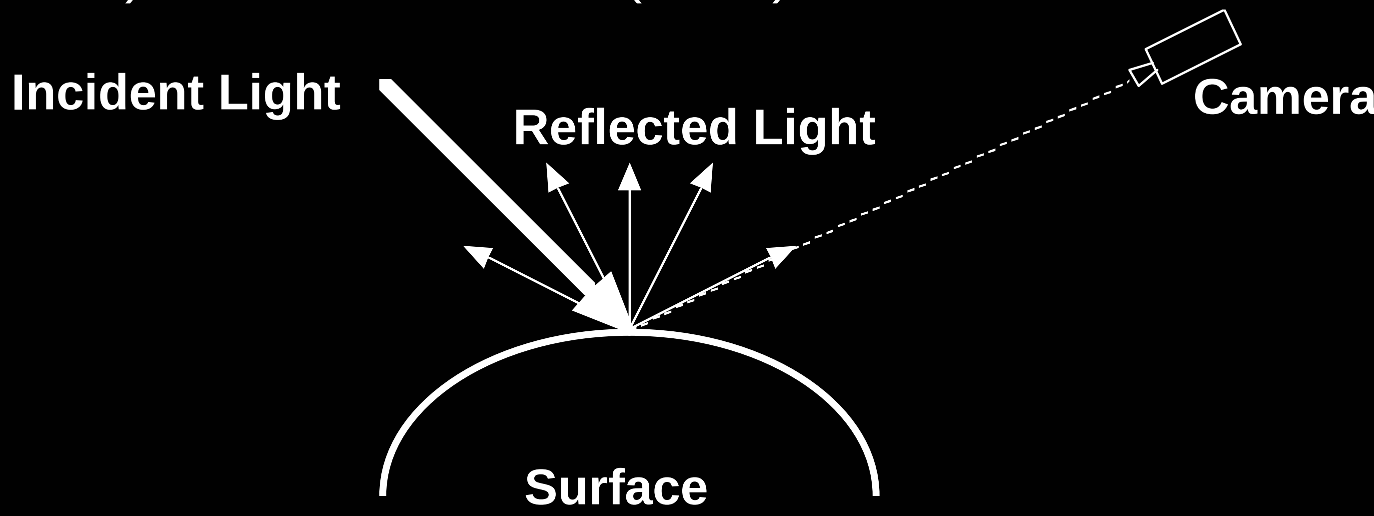
- Working with light: additive primaries
  - Red, green and blue components are added by the superposition property of electromagnetism
  - Conceptually: start with black, primaries add light
- Working with pigments: subtractive primaries
  - Typical inks (CMYK): cyan, magenta, yellow, black
  - Conceptually: start with white, pigments filter out light
  - The pigments remove parts of the spectrum

dye color	**absorbs**	**reflects**
cyan	red	blue and green
magenta	green	blue and red
yellow	blue	red and green
black	all	none

  - Inks interact in nonlinear ways--makes converting from monitor color to printer color a challenging problem
  - Black ink (K) used to ensure a high quality black can be printed

Surface Reflection

- When light hits an opaque surface some is absorbed, the rest is reflected (some can be transmitted too--but never mind for now)
- The reflected light is what we see
- Reflection is not simple and varies with material
  – the surface’s micro structure define the details of reflection
  – variations produce anything from bright specular reflection (mirrors) to dull matte finish (chalk)



Units of Light and Color

Quantity	*Dimension*	*Units*
solid angle	solid angle	[steradian]
power	energy/time	[watt]=[joule/sec]
irradiance (intensity)	power/area	[watt/m^2]
radiance (intensity)	power/(area*solid angle)	[watt/(m^2*steradian)]
reflectance	unitless	[1]

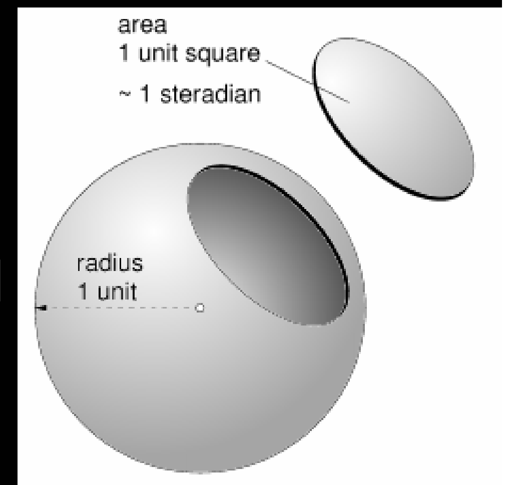
solid angle: a two-dimensional angle (proportional to area on a sphere)

power: photons per second; radiance integrated over incoming directions, over a finite area.

irradiance: Brightness of light hitting the surface (or image) at this point (incident light)

radiance: Brightness of light reflected at this point along this direction (reflected light)

reflectance: what fraction of the light is reflected by a material? typically between 0 and 1.



The Meaning of "Color"

- What's an image?
  - Irradiance: each pixel measures the incident light at a point on the film
  - Proportional to integral of scene radiance hitting that point
- What's Color?
  - Refers to radiance or irradiance measured at 3 wavelengths
  - Scene color: radiance coming off of surface (for illumination)
  - Image color: irradiance (for rendering)
  - These quantities have different units and should not be confused

Outline

- Properties of Light
- Light Sources
- Phong Illumination Model
- Normal Vectors

Light Sources and Material Properties

- Appearance depends on
  - Light sources, their locations and properties
  - Material (surface) properties
  - Viewer position
- Ray tracing: from viewer into scene
- Radiosity: between surface patches
- Phong Model: at material, from light to viewer

Common Types of Light Sources

- Ambient light: no identifiable source or direction
- Point source: given only by point
- Distant light: given only by direction
- Spotlight: from source in direction
  - Cut-off angle defines a cone of light
  - Attenuation function (brighter in center)
- Light source described by a luminance
  - Each color is described separately
  - $I = [I_r \;\; I_g \;\; I_b]^T$ (I for intensity)
  - Sometimes calculate generically (applies to r, g, b)

Ambient Light

- Global ambient light
  - Independent of light source
  - Lights entire scene
- Local ambient light
  - Contributed by additional light sources
  - Can be different for each light and primary color
- Computationally inexpensive

$$\mathbf{I}_a = \begin{bmatrix} I_{ar} \\ I_{ag} \\ I_{ab} \end{bmatrix}$$

Point Source

- Given by a point p_0
- Light emitted equally in all directions

$$\mathbf{I}(\mathbf{p}_0) = \begin{bmatrix} I_r(\mathbf{p}_0) \\ I_g(\mathbf{p}_0) \\ I_b(\mathbf{p}_0) \end{bmatrix}$$

- Intensity decreases with square of distance

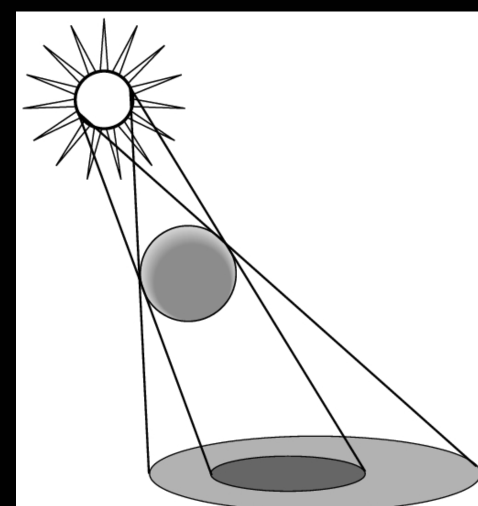
$$\mathbf{I}(\mathbf{p}, \mathbf{p}_0) = \frac{1}{|\mathbf{p} - \mathbf{p}_0|^2} \mathbf{I}(\mathbf{p}_0)$$

Limitations of Point Sources

- Shading and shadows inaccurate
- Example: penumbra (partial “soft” shadow)
- Similar problems with highlights
- Compensate with attenuation

$$\frac{1}{(a + bd + cd^2)}$$

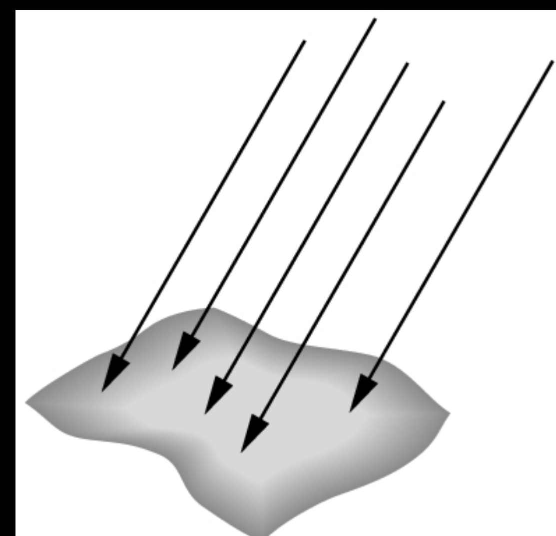
d = distance $|p - p_0|$
a, b, c constants



- Softens lighting
- Better with ray tracing
- Better with radiosity

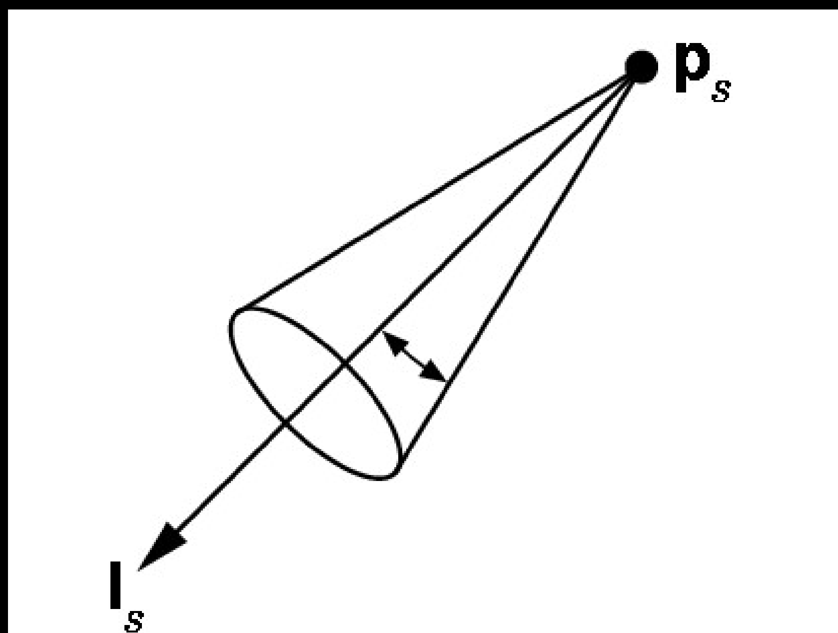
Distant Light Source

- Given by a vector v
- Simplifies some calculations
- In OpenGL:
  - Point source $[x\ \ y\ \ z\ \ 1]^T$
  - Distant source $[x\ \ y\ \ z\ \ 0]^T$



Spotlight

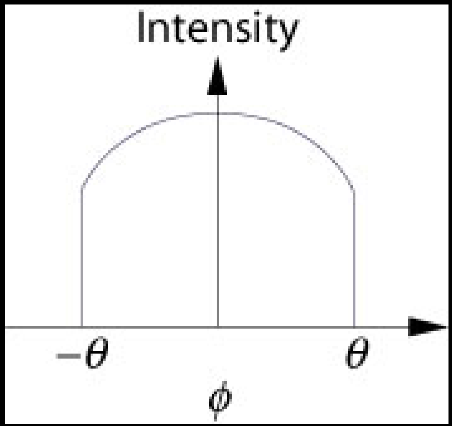
- Most complex light source in OpenGL
- Light still emanates from point
- Cut-off by cone determined by angle θ



Spotlight Attenuation

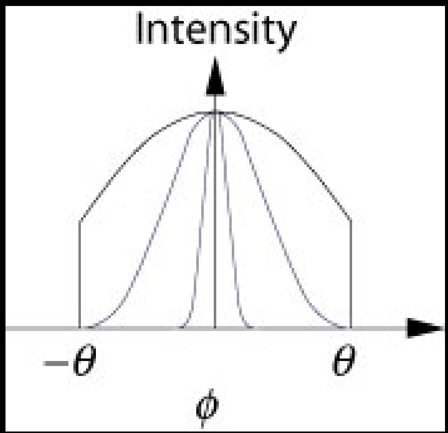
- Spotlight is brightest along l_s
- Vector v with angle ϕ from p to point on surface
- Intensity determined by cos ϕ
- Corresponds to projection of v onto l_s
- Spotlight exponent e determines rate

$$I = \cos^e(\phi) = (\mathbf{v} \cdot \mathbf{l}_s)^e$$



for e = 1

for e > 1
curve narrows



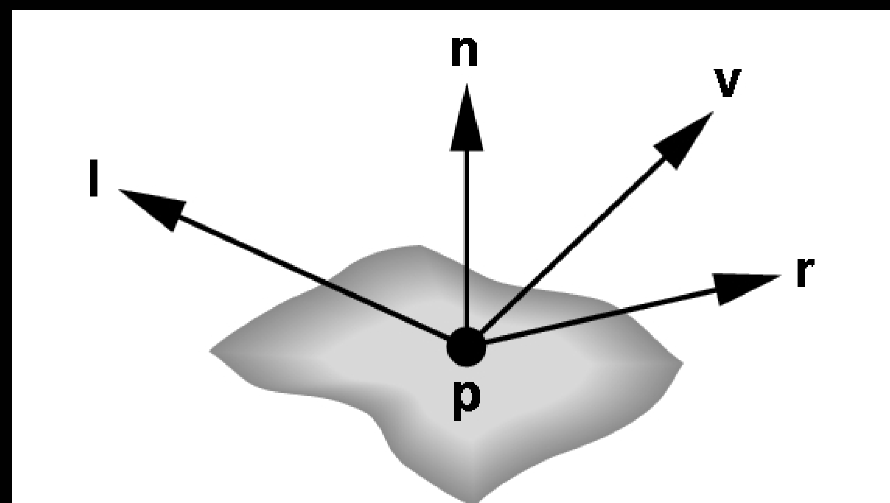
Outline

- Properties of Light
- Light Sources
- Phong Illumination Model
- Normal Vectors

Phong Illumination Model

- Calculate color for arbitrary point on surface
- Compromise between realism and efficiency
- Local computation (no visibility calculations)
- Basic inputs are material properties and l, n, v:

l = vector to light source
n = surface normal
v = vector to viewer
r = reflection of l at p
(determined by l and n)



Basic Calculation

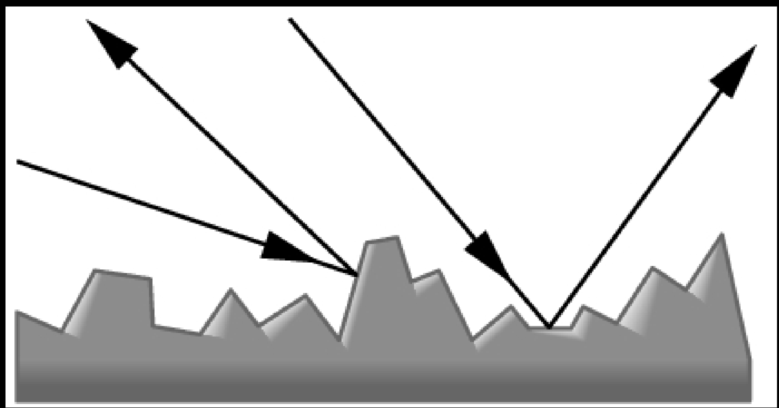
- Calculate each primary color separately
- Start with global ambient light
- Add reflections from each light source
- Clamp to [0, 1]
- Reflection decomposed into
  - Ambient reflection
  - Diffuse reflection
  - Specular reflection
- Based on ambient, diffuse, and specular lighting and material properties

Ambient Reflection

- Intensity of ambient light uniform at every point
- Ambient reflection coefficient k_a, $0 \leq k_a \leq 1$
- May be different for every surface and r,g,b
- Determines reflected fraction of ambient light
- L_a = ambient component of light source
- Ambient intensity $I_a = k_a L_a$
- Note: L_a is not a physically meaningful quantity

Diffuse Reflection

- Diffuse reflector scatters light
- Assume equally all direction
- Called Lambertian surface
- Diffuse reflection coefficient k_d, $0 \leq k_d \leq 1$
- Angle of incoming light still critical



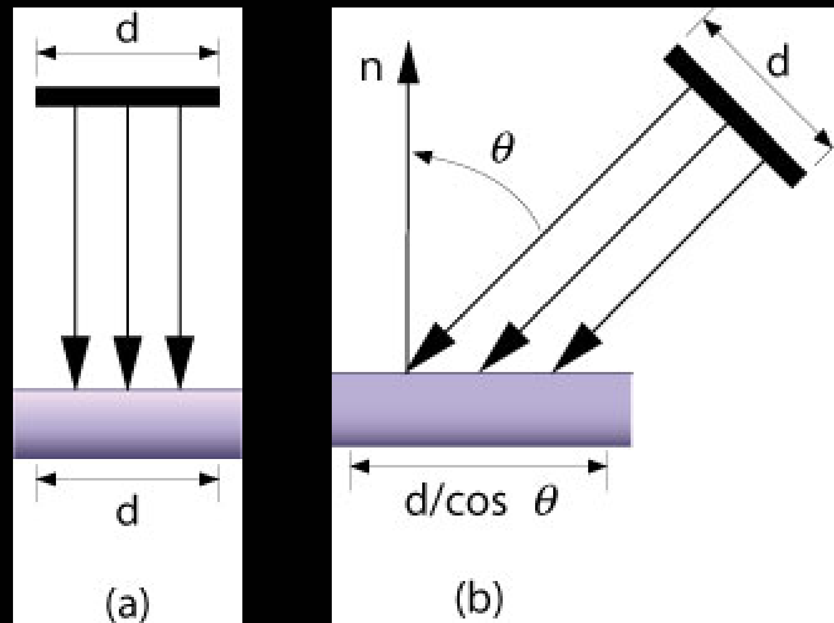
Lambert's Law

- Intensity depends on angle of incoming light
- Recall

  l = unit vector to light

  n = unit surface normal

  θ = angle to normal
- $\cos\theta = \mathbf{l} \cdot \mathbf{n}$
- $I_d = k_n (\mathbf{l} \cdot \mathbf{n}) L_d$
- With attenuation:



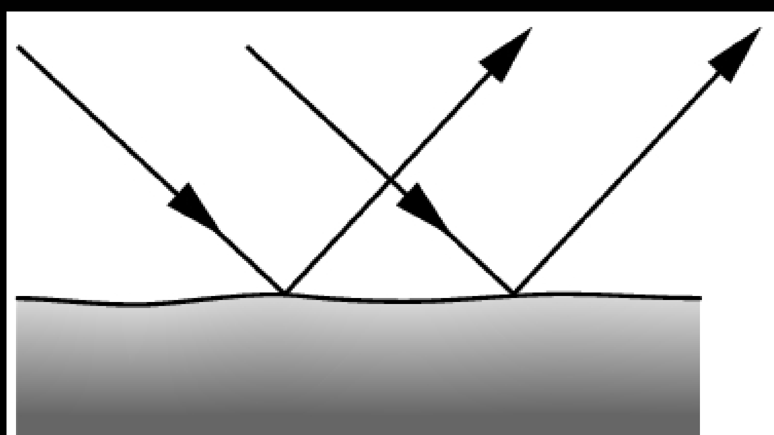
$$I_d = \frac{k_d}{a + bq + cq^2}(\mathbf{l} \cdot \mathbf{n}) L_d$$

q = distance to light source,

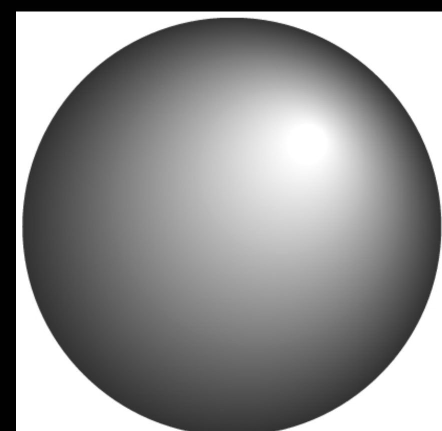
L_d = diffuse component of light

Specular Reflection

- Specular reflection coefficient k_s, $0 \leq k_s \leq 1$
- Shiny surfaces have high specular coefficient
- Used to model specular highlights
- Do not get mirror effect (need other techniques)



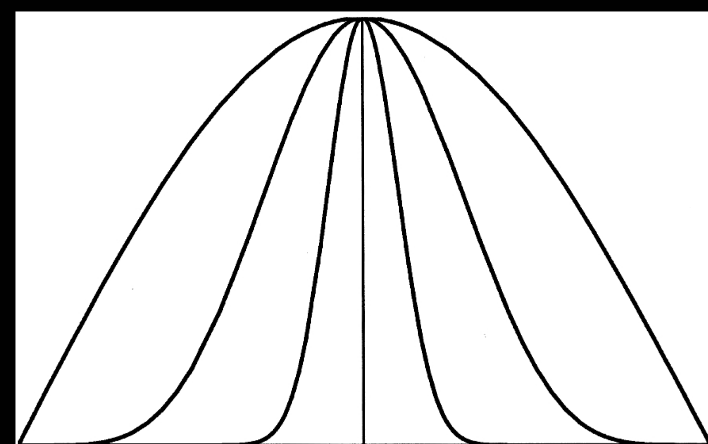
specular reflection



specular highlights

Shininess Coefficient

- L_s is specular component of light
- r is vector of perfect reflection of l about n
- v is vector to viewer
- ϕ is angle between v and r
- $I_s = k_s L_s \cos^{\alpha} \phi$
- α is shininess coefficient
- Compute $\cos \phi = r \cdot v$
- Requires $|r| = |v| = 1$
- Multiply distance term



Higher α is narrower

Summary of Phong Model

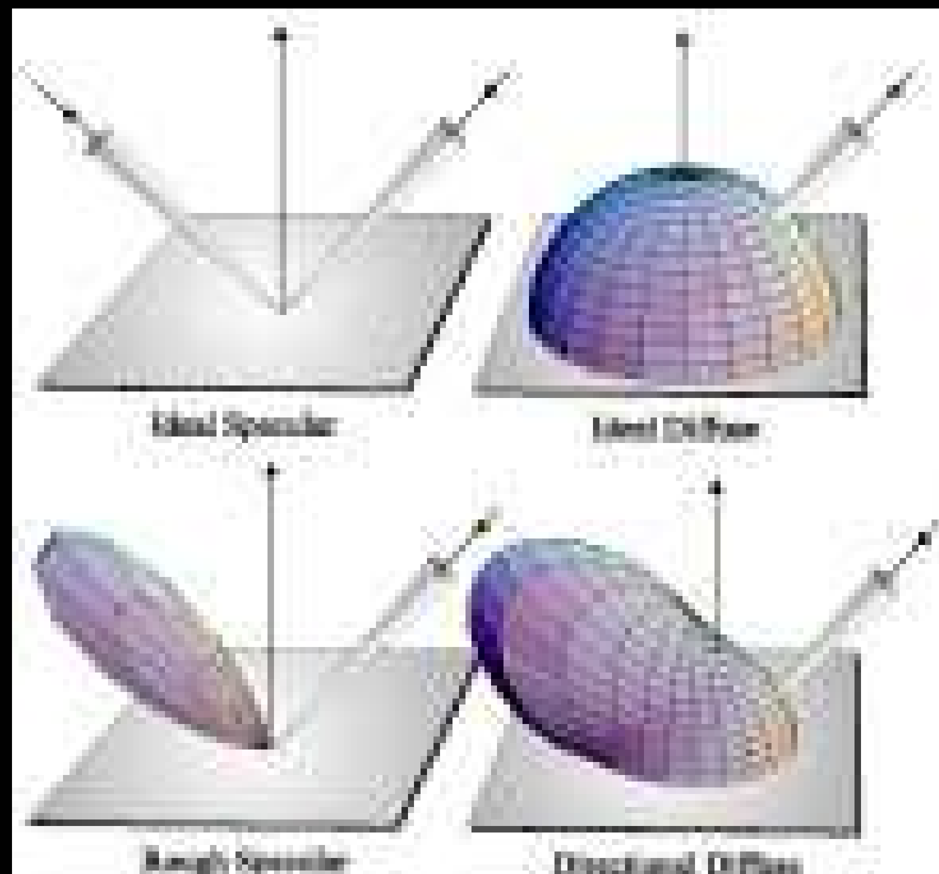
- Light components for each color:
  - Ambient (L_a), diffuse (L_d), specular (L_s)
- Material coefficients for each color:
  - Ambient (k_a), diffuse (k_d), specular (k_s)
- Distance q for surface point from light source

$$I = \frac{1}{a + bq + cq^2}(k_d L_d (\mathbf{l} \cdot \mathbf{n}) + k_s L_s (\mathbf{r} \cdot \mathbf{v})^{\alpha}) + k_a L_a$$

l = vector from light
n = surface normal
r = l reflected about n
v = vector to viewer

BRDF

- Bidirectional Reflection Distribution Function
- Measure for materials
- Isotropic vs. anisotropic
- Mathematically complex
- Programmable pixel shading?



Outline

- Properties of Light
- Light Sources
- Phong Illumination Model
- Normal Vectors

Normal Vectors

- Summarize Phong

$$I = \frac{1}{a + bq + cq^2}(k_d L_d (\mathbf{l} \cdot \mathbf{n}) + k_s L_s (\mathbf{r} \cdot \mathbf{v})^\alpha) + k_a L_a$$

- Surface normal n is critical
  - Calculate l · n
  - Calculate r and then r · v
- Must calculate and specify the normal vector
  - Even in OpenGL!
- Two examples: plane and sphere

Normals of a Plane, Method I

- Method I: given by $ax + by + cz + d = 0$
- Let p_0 be a known point on the plane
- Let p be an arbitrary point on the plane
- Recall: $u \cdot v = 0$ iff u orthogonal v
- $n \cdot (p - p_0) = n \cdot p - n \cdot p_0 = 0$
- Consequently $n_0 = [a \;\; b \;\; c \;\; 0]^T$
- Normalize to $n = n_0/|n_0|$

Normals of a Plane, Method II

- Method II: plane given by p_0, p_1, p_2
- Points must not be collinear
- Recall: $u \times v$ orthogonal to u and v
- $n_0 = (p_1 - p_0) \times (p_2 - p_0)$
- Order of cross product determines orientation
- Normalize to $n = n_0/|n_0|$

Normals of Sphere

- Implicit Equation $f(x, y, z) = x^2 + y^2 + z^2 - 1 = 0$
- Vector form: $f(p) = p \cdot p - 1 = 0$

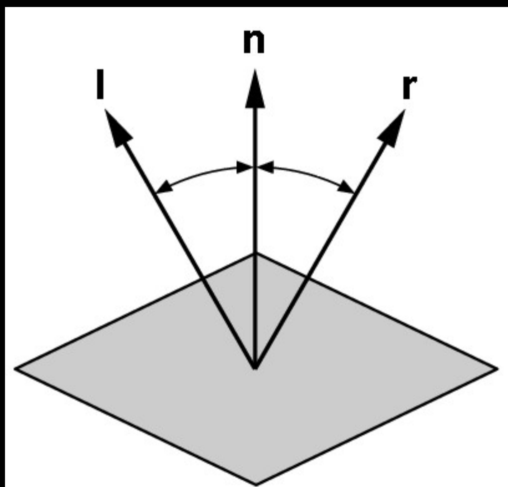
- Normal given by gradient vector

$$\mathbf{n}_0 = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{bmatrix} = \begin{bmatrix} 2x \\ 2y \\ 2z \end{bmatrix} = 2\mathbf{p}$$

- Normalize $n_0/|n_0| = 2p/2 = p$

Angle of Reflection

- Perfect reflection: angle of incident equals angle of reflection
- Also: l, n, and r lie in the same plane
- Assume |l| = |n| = 1, guarantee |r| = 1



$$\mathbf{l} \cdot \mathbf{n} = \cos\theta = \mathbf{n} \cdot \mathbf{r}$$

$$\mathbf{r} = \alpha \mathbf{l} + \beta \mathbf{n}$$

Solution: $\alpha = -1$ and $\beta = 2\,(l \cdot n)$

$$\mathbf{r} = 2(\mathbf{l} \cdot \mathbf{n})\mathbf{n} - \mathbf{l}$$

Perhaps easier geometrically

Summary: Normal Vectors

- Critical for Phong model (diffuse and specular)
- Must calculate accurately (even in OpenGL)
- Pitfalls
  - Not unit length
  - How to set at surface boundary?
- Omitted
  - Refraction of transmitted light (Snell's law)
  - Halfway vector (yet another optimization)

Summary

- Properties of Light
- Light Sources
- Phong Illumination Model
- Normal Vectors

Preview

- Polygonal shading
- Lighting and shading in OpenGL

Announcements

- Programming assignment #1 due today
  - Electronic handin by midnight
- Written assignment #1 due Thursday
  - Handin at beginning of class
- Programming assignment #2 out Thursday
  - Due in 2 weeks