CS 4204 Computer Graphics

Lighting and Shading

Yong Cao

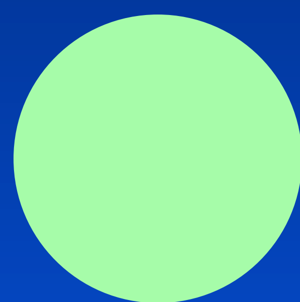
Virginia Tech

Objectives

- *Learn to shade objects so their images appear three-dimensional*
- *Introduce the types of light-material interactions*
- *Build a simple reflection model---the Phong model--- that can be used with real time graphics hardware*

Why we need shading

Suppose we build a model of a sphere using many polygons and color it with `glColor`. We get something like



But we want

Shading

Why does the image of a real sphere look like

Light-material interactions cause each point to have a different color or shade

Need to consider

- Light sources
- Material properties
- Location of viewer
- Surface orientation

Scattering

Light strikes A

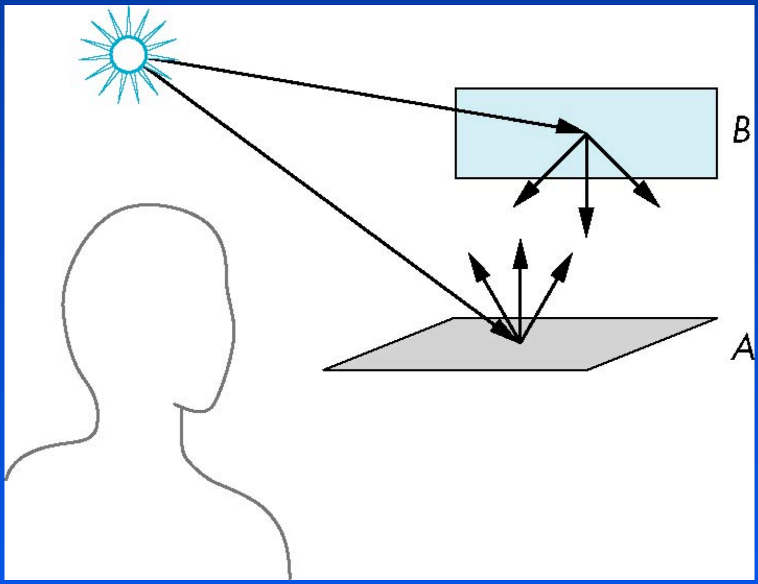
- Some scattered
- Some absorbed

Some of scattered light strikes B

- Some scattered
- Some absorbed

Some of this scattered light strikes A

and so on …



Rendering Equation

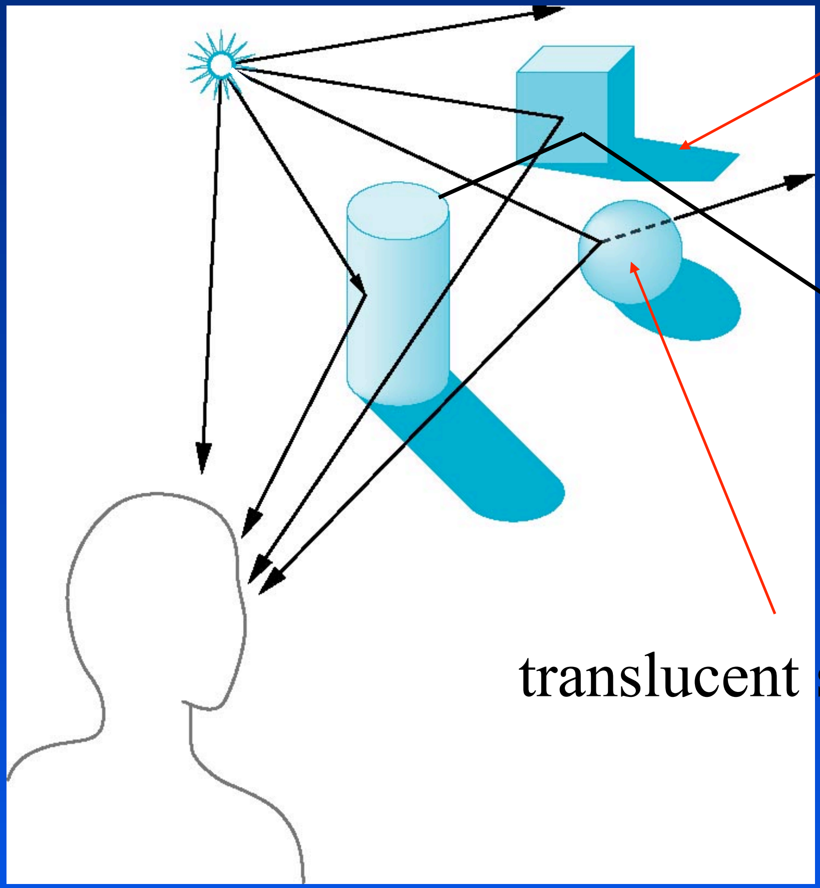
The infinite scattering and absorption of light can be described by the rendering equation

- Cannot be solved in general
- Ray tracing is a special case for perfectly reflecting surfaces

Rendering equation is global and includes

- Shadows
- Multiple scattering from object to object

Global Effects



Local vs Global Rendering

Correct shading requires a global calculation involving all objects and light sources

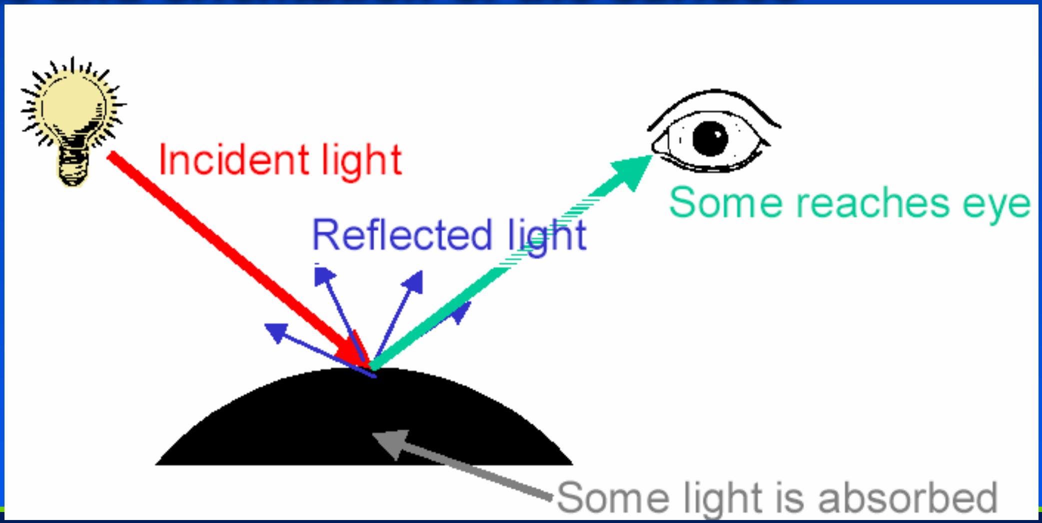
- Incompatible with pipeline model which shades each polygon independently (local rendering)

However, in computer graphics, especially real time graphics, we are happy if things "look right"

- Exist many techniques for approximating global effects

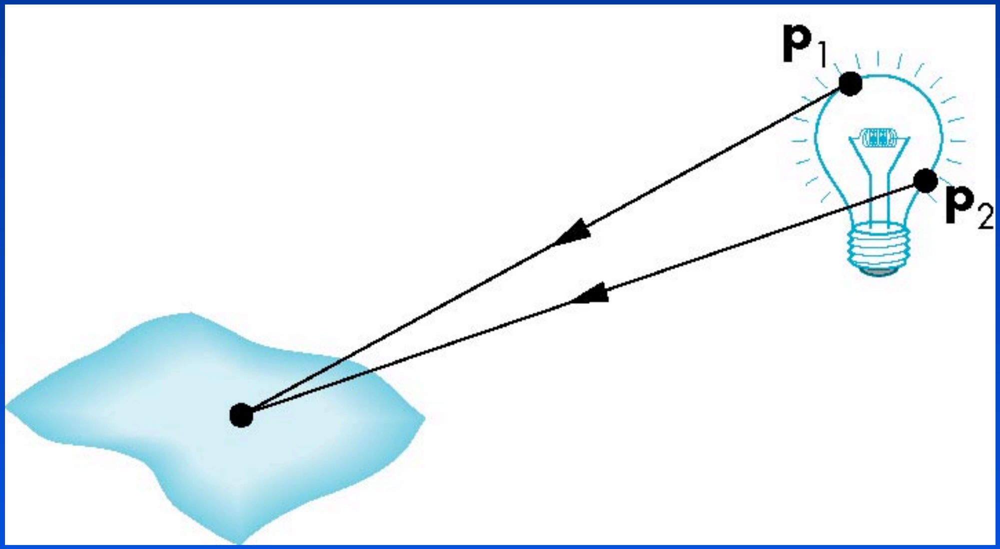
Light-Material Interaction

- ***Light that strikes an object is partially absorbed and partially scattered (reflected)***
- ***The amount reflected determines the color and brightness of the object***
  - A surface appears red under white light because the red component of the light is reflected and the rest is absorbed
- ***The reflected light is scattered in a manner that depends on the smoothness and orientation of the surface***



Light Sources

General light sources are difficult to work with because we must integrate light coming from all points on the source



Simple Light Sources

Point source

- Model with position and color
- Distant source = infinite distance away (parallel)

Spotlight

- Restrict light from ideal point source

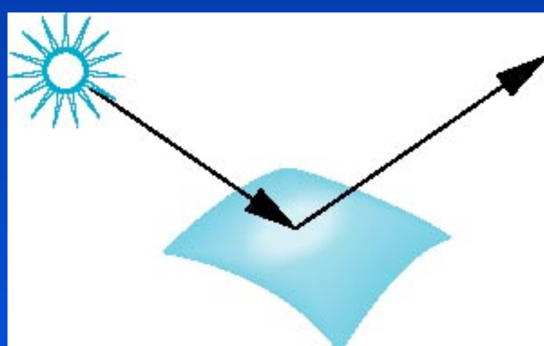
Ambient light

- Same amount of light everywhere in scene
- Can model contribution of many sources and reflecting surfaces

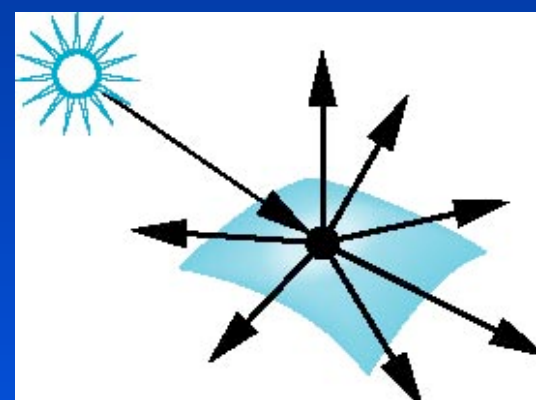
Surface Types

The smoother a surface, the more reflected light is concentrated in the direction a perfect mirror would reflected the light

A very rough surface scatters light in all directions



smooth surface

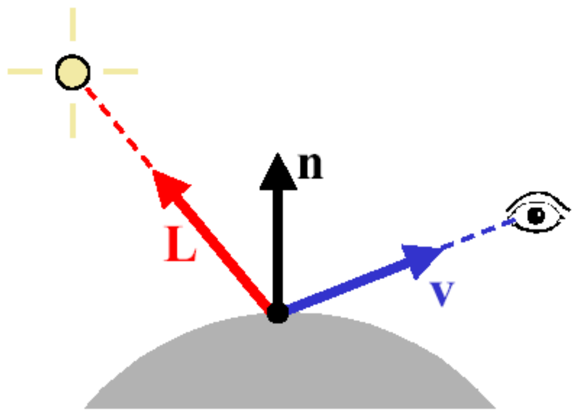


rough surface

Basic Local Illumination Model

We're only interested in light that finally arrives at view point

- a function of the light & viewing positions
- and local surface reflectance



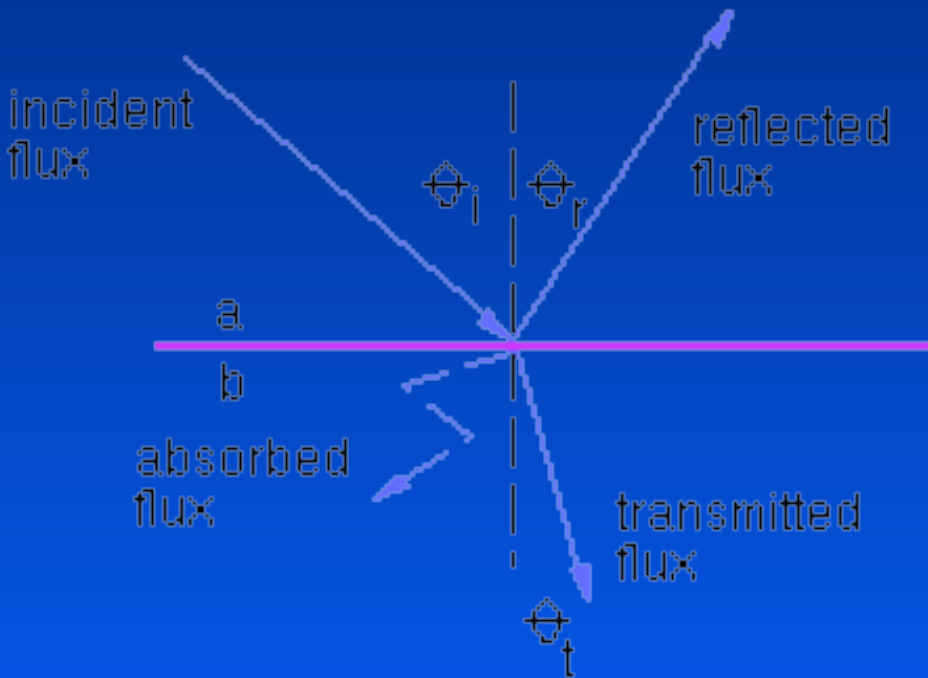
Characterize light using RGB triples

- can operate on each channel separately

Given a point, compute intensity of reflected light

Local Illumination physics

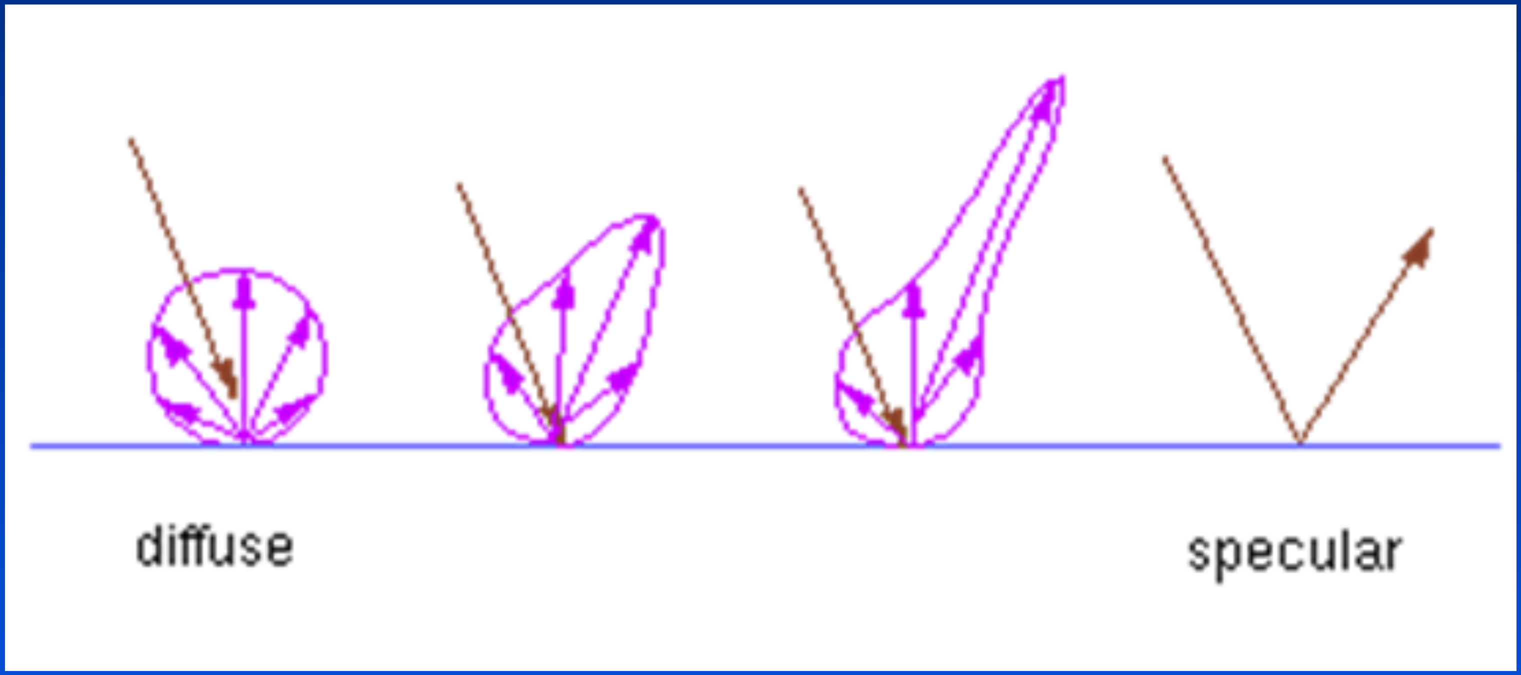
Law of reflection and Snell's law of refraction



$$\theta_i = \theta_r$$

$$\frac{n_b}{n_a} = \frac{\sin(\theta_i)}{\sin(\theta_t)}$$

What are we trying to model ?



Diffuse Reflection

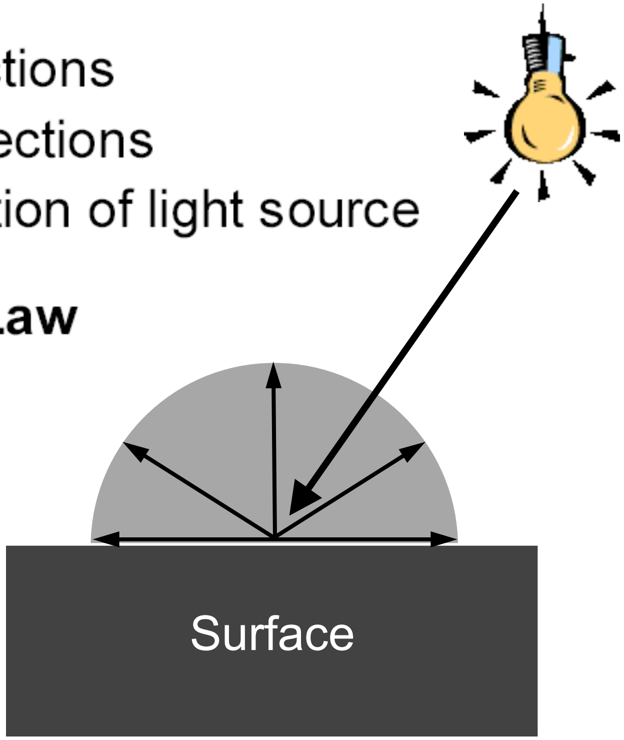
This is the simplest kind of reflection

- also called Lambertian reflection
- models dull, matte surfaces — materials like chalk

Ideal diffuse reflection

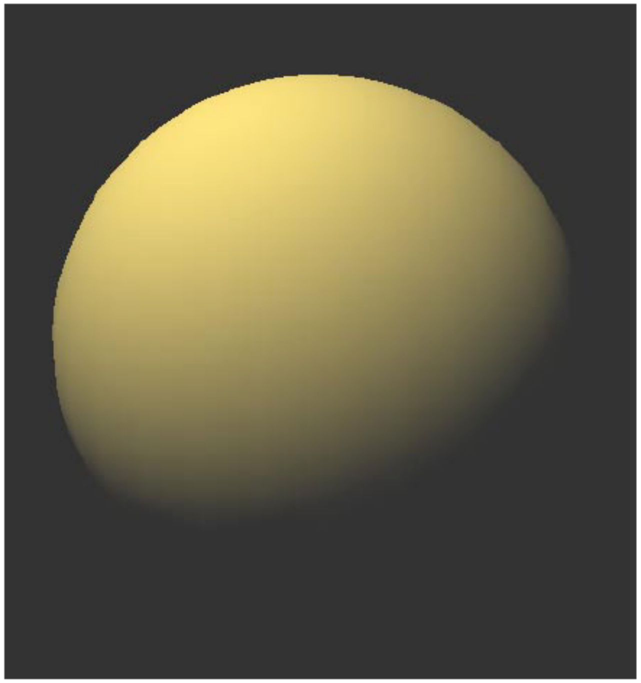
- scatters incoming light equally in all directions
- identical appearance from all viewing directions
- reflected intensity depends only on direction of light source

Light is reflected according to Lambert's Law

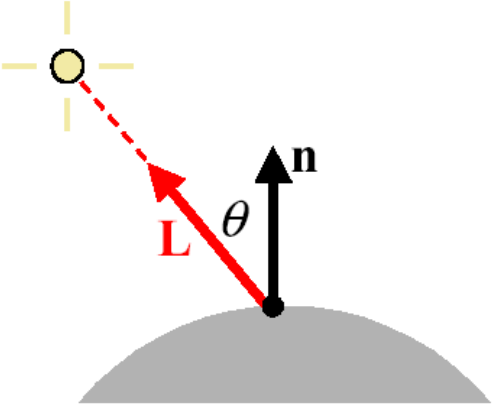


Lambert's Law for Diffuse Reflection

Purely diffuse object



$$I = I_L k_d \cos\theta$$
$$= I_L k_d (\mathbf{n} \cdot \mathbf{L})$$



I : resulting intensity

I_L : light source intensity

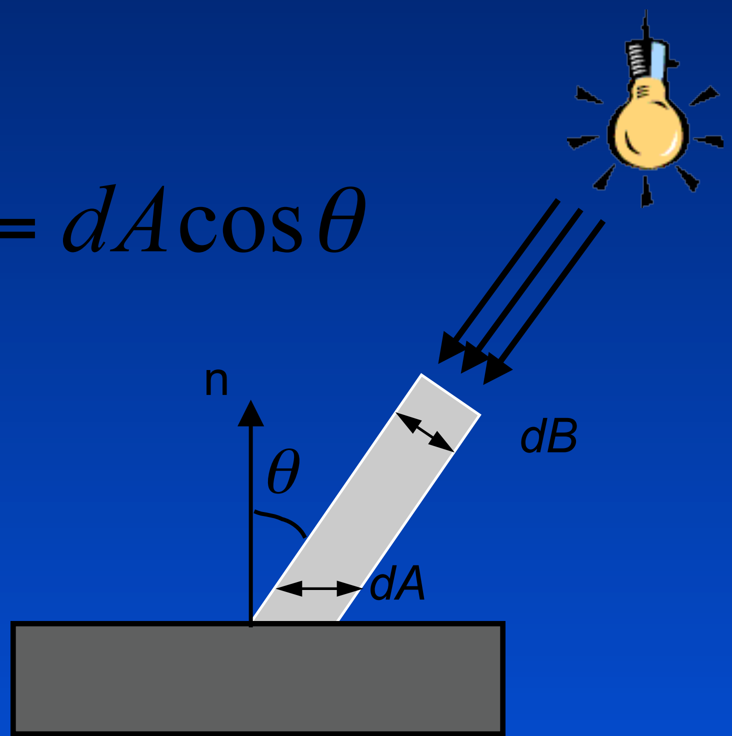
k_d : (diffuse) surface reflectance coefficient

$$k_d \in [0,1]$$

θ : angle between normal & light direction

Proof of Lambert's cosine law

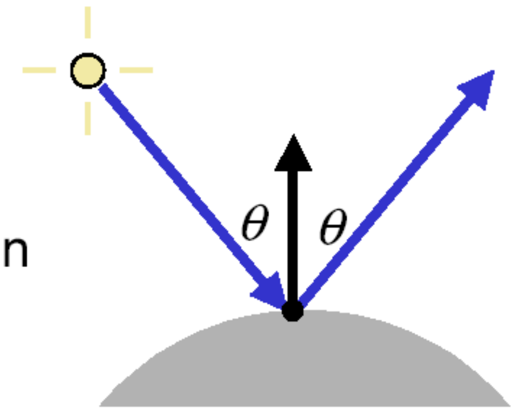
$$dB = dA\cos\theta$$



Specular Reflection

Diffuse reflection is nice, but many surfaces are shiny

- their appearance changes as the viewpoint moves
- they have glossy specular highlights (or specularities)
- because they reflect light coherently, in a preferred direction

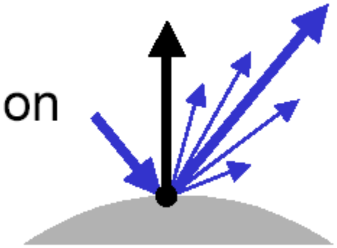


A mirror is a perfect specular reflector

- incoming ray reflected about normal direction
- nothing reflected in any other direction

Most surfaces are imperfect specular reflectors

- reflect rays in cone about perfect reflection direction



Modeling Specular Reflections

Phong proposed using a term that dropped off as the angle between the viewer and the ideal reflection increased

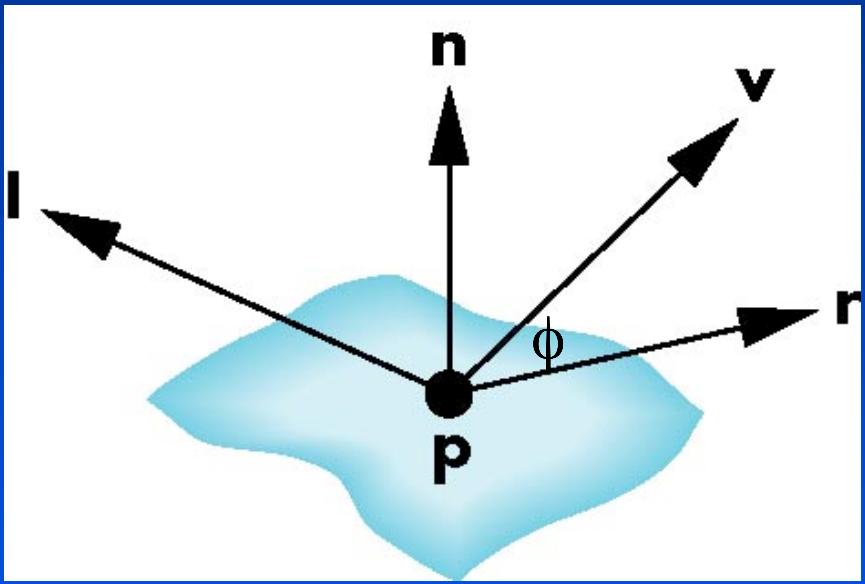
$$I_r = I_L k_s \cos^n \phi$$

reflected intensity

incoming intensity

absorption coef

shininess coef

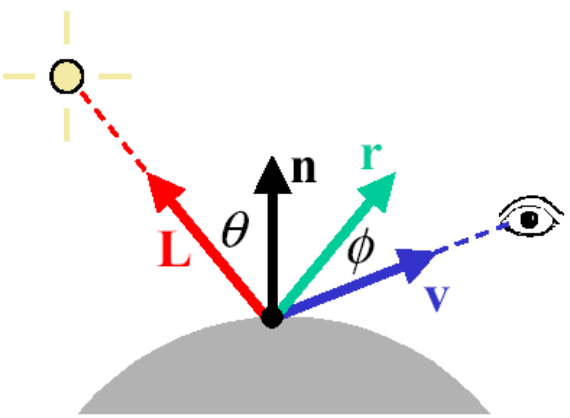


Phong Illumination Model

$$I = I_L k_d \cos\theta + I_L k_s \cos^n \phi$$
$$= I_L k_d (\mathbf{n}\cdot\mathbf{L}) + I_L k_s (\mathbf{r}\cdot\mathbf{v})^n$$

One particular specular reflection model

- quite common in practice
- it is purely empirical
- there's *no physical basis* for it



I : resulting intensity

I_L : light source intensity

k_s : (specular) surface reflectance coefficient

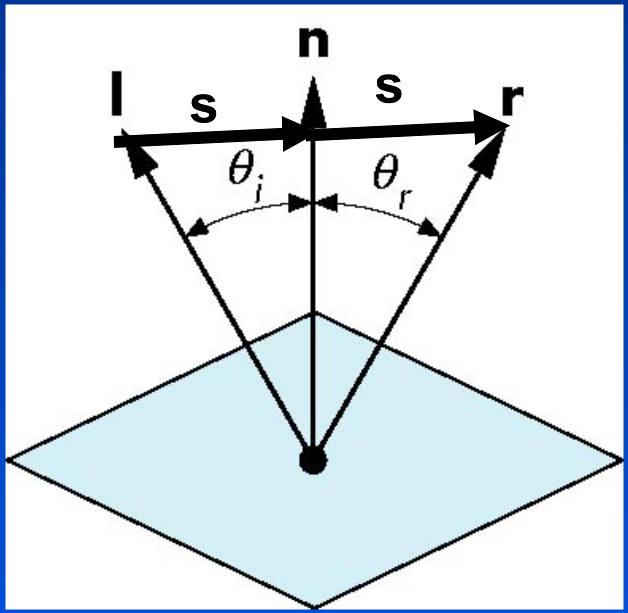
$$k_s \in [0,1]$$

ϕ : angle between viewing & reflection direction

n : "shininess" factor

Computing R

All vectors unit length!!

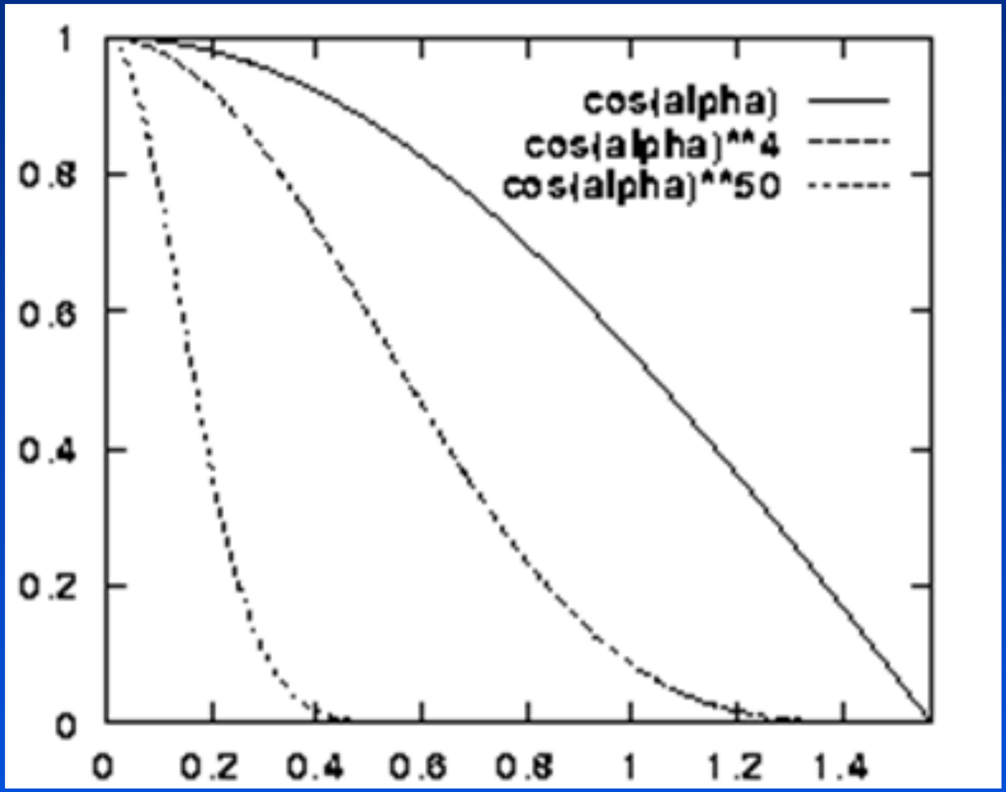


$$r = (n \cdot l)n + s$$

$$s = (n \cdot l)n - l$$

$$\Rightarrow r = 2n(n \cdot l) - l$$

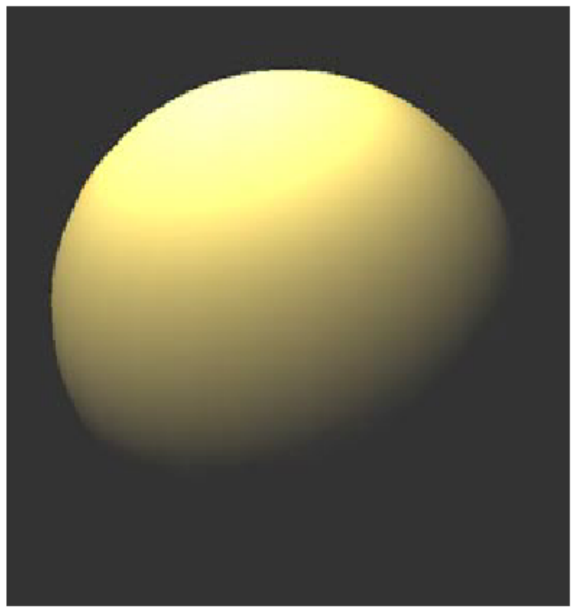
The effect of the exponent *n*



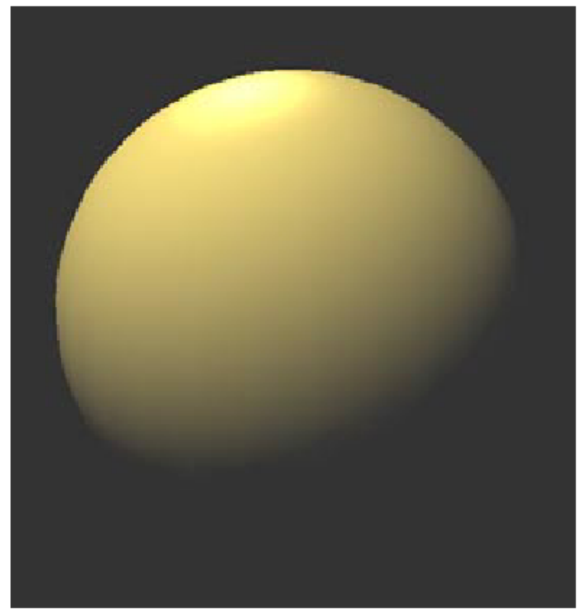
Examples of Phong Specular Model

Diffuse only

Diffuse + Specular (shininess 5)



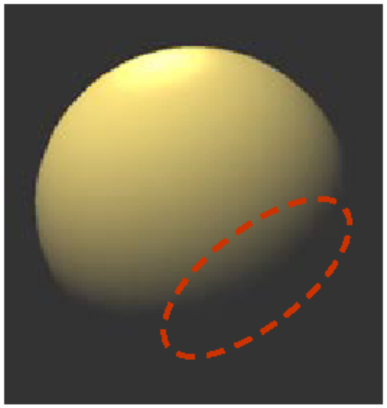
Diffuse + Specular (shininess 50)



The Ambient Glow

So far, areas not directly illuminated by any light appear black

- this tends to look rather unnatural
- in the real world, there's lots of ambient light



To compensate, we invent new light source

- assume there is a constant ambient "glow"
- this ambient glow is *purely fictitious*

Just add in another term to our illumination equation

$$I = I_L k_d \cos\theta + I_L k_s \cos^n \phi + I_a k_a$$

I_a : ambient light intensity

k_a : (ambient) surface reflectance coefficient

Our Three Basic Components of Illumination

Diffuse Specular Ambient

Combined for the Final Result

Phong Model

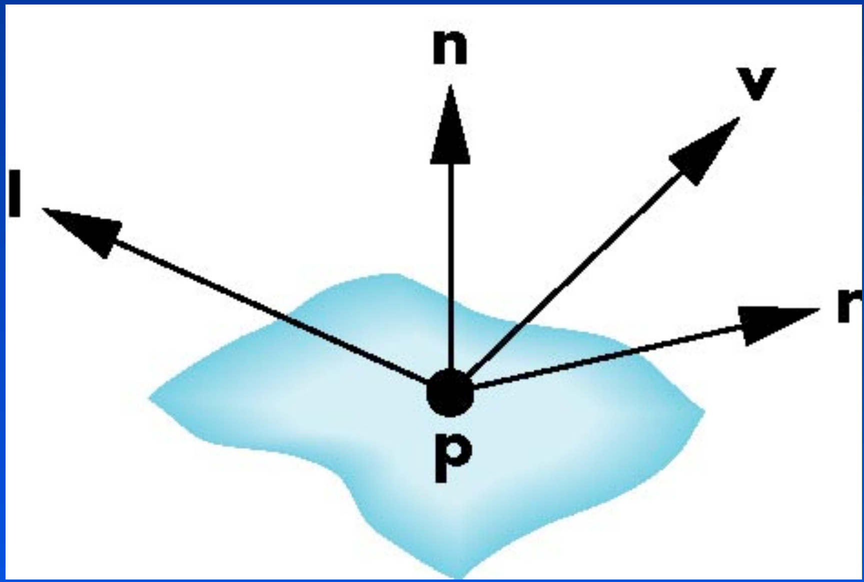
A simple model that can be computed rapidly

Has three components

- Diffuse
- Specular
- Ambient

Uses four vectors

- To source
- To viewer
- Normal
- Perfect reflector

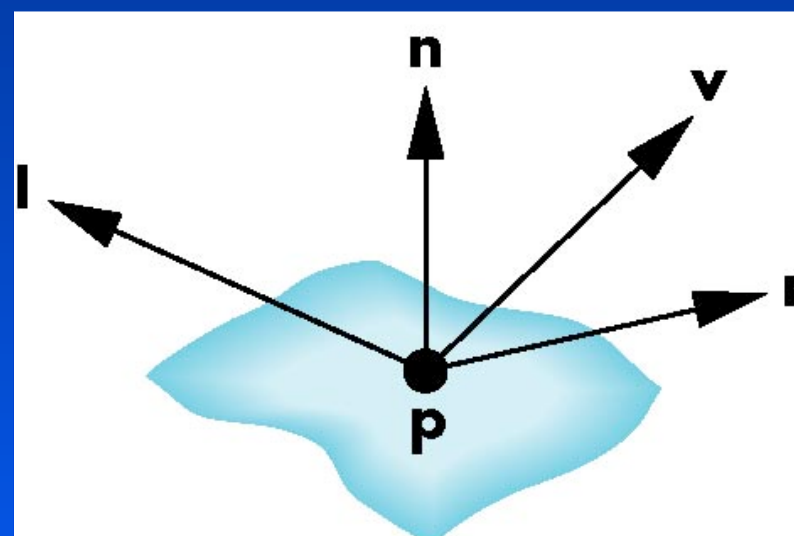


Adding up the Components

For each light source and each color component, the Phong model can be written (without the distance terms) as

$$I = k_d I_d \; l \cdot n \; + k_s I_s (v \cdot r)^n + k_a I_a$$

For each color component we add contributions from all sources



Lights and materials

$ObjectColor_r = I_r = I_{a\_r}K_{a\_r} + I_{i\_r}K_{diff\_r}(N \cdot L) + I_{i\_r}K_{spec\_r}(R \cdot V)^n$

$ObjectColor_g = I_g = I_{a\_g}K_{a\_g} + I_{i\_g}K_{diff\_g}(N \cdot L) + I_{i\_g}K_{spec\_g}(R \cdot V)^n$

$ObjectColor_b = I_b = I_{a\_b}K_{a\_b} + I_{i\_b}K_{diff\_b}(N \cdot L) + I_{i\_b}K_{spec\_b}(R \cdot V)^n$

Material properties:

$K_a, K_{diff}, K_{spec}, n$

Light properties

I_a, I_{diff}, I_{spec}

Special cases

$$I_r = I_{a\_r}K_{a\_r} + I_{i\_r}K_{diff\_r}(N \cdot L) + I_{i\_r}K_{spec\_r}(R \cdot V)^n$$

$$I_g = I_{a\_g}K_{a\_g} + I_{i\_g}K_{diff\_g}(N \cdot L) + I_{i\_g}K_{spec\_g}(R \cdot V)^n$$

$$I_b = I_{a\_b}K_{a\_b} + I_{i\_b}K_{diff\_b}(N \cdot L) + I_{i\_b}K_{spec\_b}(R \cdot V)^n$$

- What should be done if I >1?
  Clamp the value of I to one.
- What should be done if N*L < 0?
  Clamp the value of I to zero or flip the normal.
- How can we handle multiple light sources?
  Sum the intensity of the individual contributions.