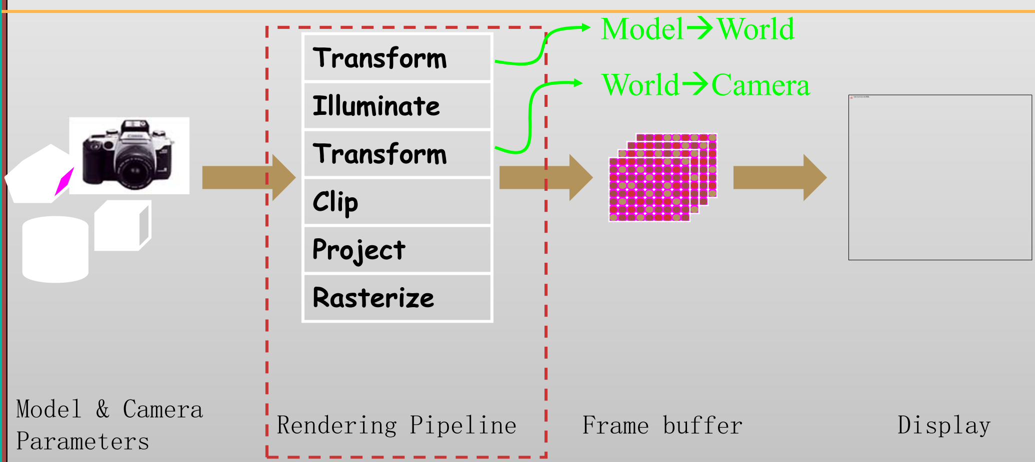
Computer Graphics

Ch9
Lighting

Instructor: Dr. MAO Aihua

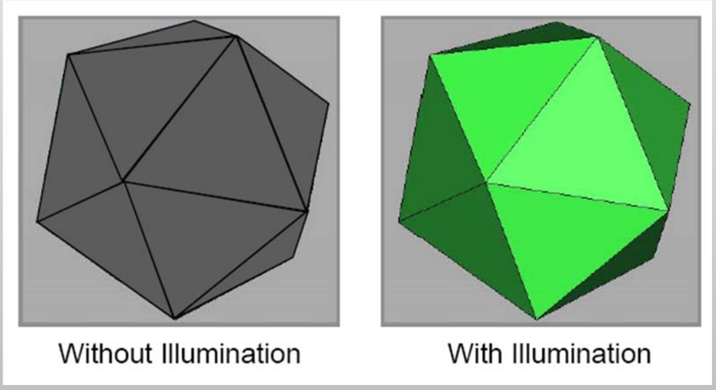
ahmao@scut.edu.cn

Rendering 3D Scenes



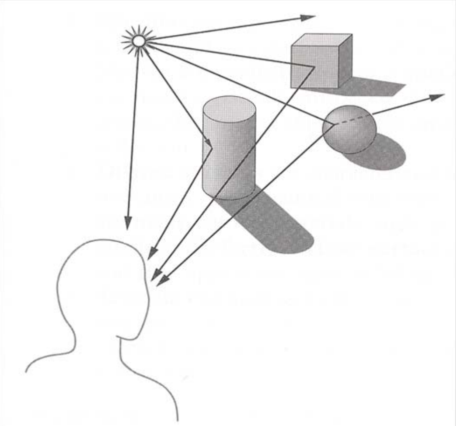
Illumination

Provides visual cues



Illumination

How do we compute radiance for a sample ray?



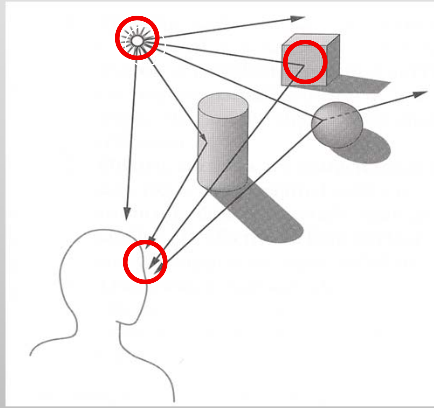
Goal

Must derive computer models for ...

- Emission at light sources
- Scattering at surfaces
- Reception at the camera

Desirable features …

- Concise
- Efficient to compute
- "Accurate"



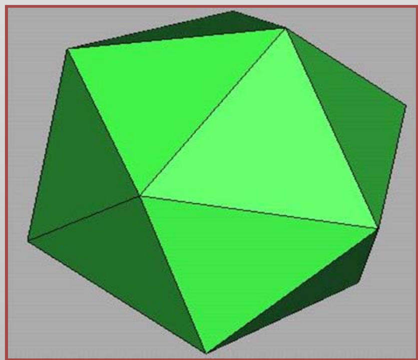
Overview

Direct (Local) Illumination

- Emission at light sources
- Scattering at surfaces

Global illumination

- Shadows
- Refractions
- Inter-object reflections



Direct Illumination

Direct Illumination

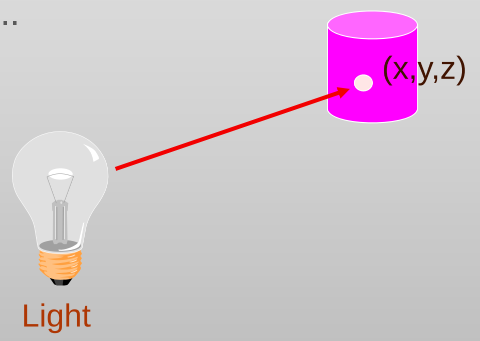
Provides a way to measure radiance at a point in a specific direction

- What points?
- What directions?
  - Towards the camera
  - No secondary effects

Modeling Light Sources

$I_L(x,y,z,\theta,\phi,\lambda)$ ...

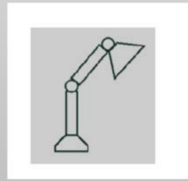
- describes the intensity of energy, leaving a light source, …
- arriving at location(x,y,z), ...
- from direction (θ,ϕ), ...
- with wavelength λ



OpenGL Light Source Models

Simple mathematical models:

- Point light
- Directional light
- Spot light



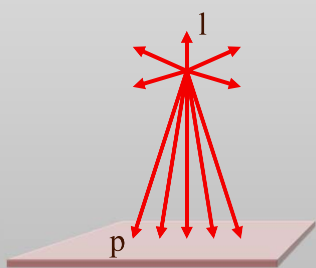
Point Light Sources

A point light source emits light equally in all directions from a single point

The direction to the light from a point on a surface thus differs for different points:

- So we need to calculate a normalized vector to the light source for every point we light:

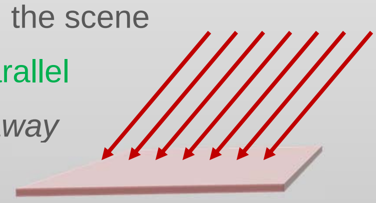
$$\bar{d} = \frac{\bar{p} - \bar{l}}{\left\| \bar{p} - \bar{l} \right\|}$$



Directional Light Sources

For a directional light source (e.g., sun) we make simplifying assumptions

- Direction is constant for all surfaces in the scene
- All rays of light from the source are parallel
  - *As if the source were infinitely far away from the surfaces in the scene*
  - *A good approximation to sunlight*

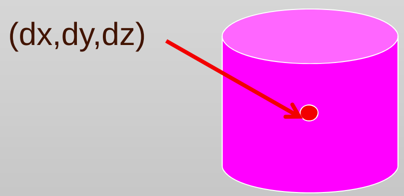


The direction from a surface to the light source is important in lighting the surface

Directional Light Sources

Models point light source at infinity (e.g., sun)

- intensity (I_0)
- direction (dx,dy,dz)



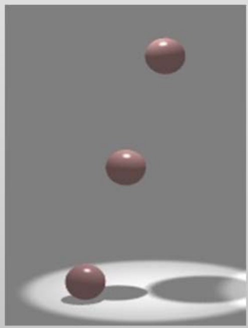
$$I_L = I_0$$

No attenuation with distance

Spot Light Sources

Spot lights are point sources whose intensity falls off directionally.

- Requires color, point direction, falloff parameters
- Supported by OpenGL

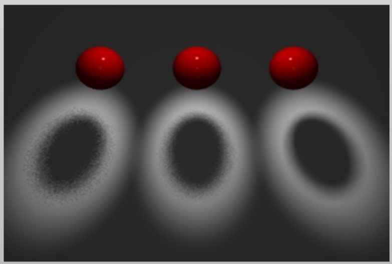


Other Light Sources

Area light sources define a 2-D emissive surface (usually a disc or polygon)

–*Good example: fluorescent light panels*

–*Capable of generating soft shadows*



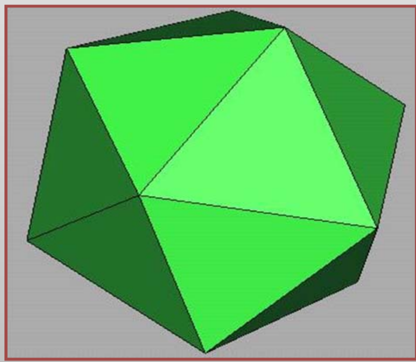
Overview

Direct (Local) Illumination

- Emission at light sources
- Scattering at surfaces

Global illumination

- Shadows
- Refractions
- Inter-object reflections



Direct Illumination

Ambient Term

Represents reflection of all indirect illumination

This is a total hack (avoids complexity of global illumination)!

Ambient Light

Objects not directly lit are typically still visible

- e.g., the ceiling in this room, undersides of desks

This is the result of indirect illumination from emitters, bouncing off intermediate surfaces

Too expensive to calculate (in real time), so we use a hack called an ambient light source

- No spatial or directional characteristics; illuminates all surfaces equally
- Amount reflected depends on surface properties

Ambient Light Sources

For each sampled wavelength (R, G, B), the ambient light reflected from a surface depends on

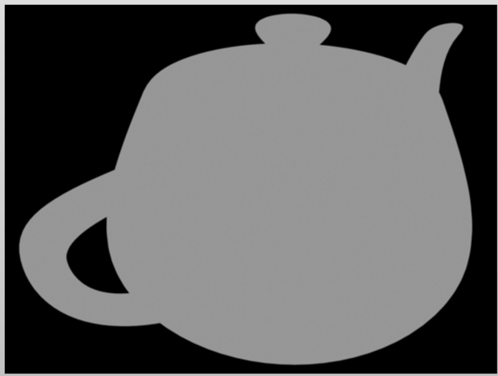
- The surface properties reflectance, $k_{ambient}$
- The intensity, $I_{ambient}$ of the ambient light source (constant for all points on all surfaces)

$$\mathrm{I}_{\text{reflected}} = \mathrm{k}_{\text{ambient}} \mathrm{I}_{\text{ambient}}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} K_{aR} * R_a \\ K_{aG} * G_a \\ K_{aB} * B_a \end{bmatrix}$$

Ambient Light Sources

A scene lit only with an ambient light source:



Light Position Not Important

Viewer Position Not Important

Surface Angle Not Important

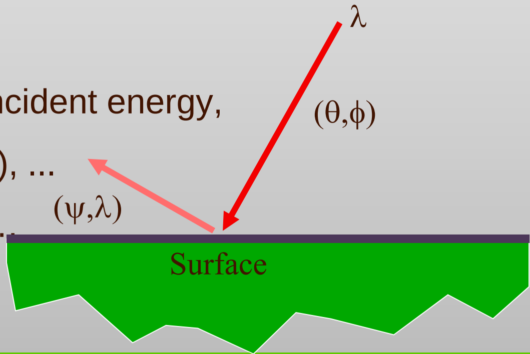
Surface Reflectance

Surface reflectance:

- Diffuse reflectance
- Specular reflection

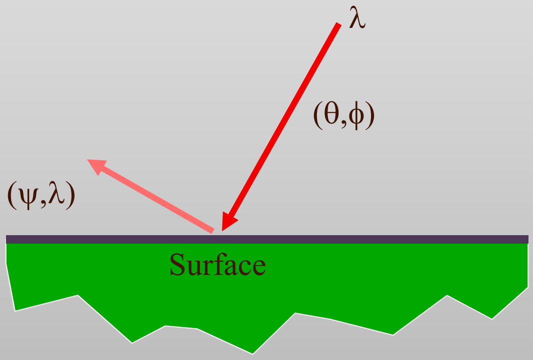
$R_S(\theta,\phi,\gamma,\psi,\lambda)$ ...

- describes the amount of incident energy,
- arriving from direction (θ,ϕ), ...
- leaving in direction (γ,ψ), ...
- with wavelength λ



Empirical Models

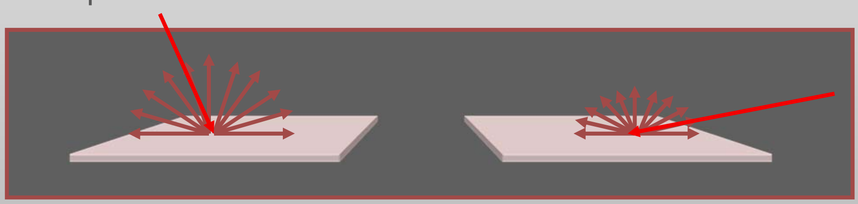
Ideally measure radiant energy for "all" combinations of incident angles is difficult in practice



The Physics of Diffuse Reflection

Ideal diffuse reflection

- An *ideal diffuse reflector*, at the microscopic level, is a rough surface (real-world example: chalk)
- Because of these microscopic variations, an incoming ray of light is equally likely to be reflected in any direction over the hemisphere:

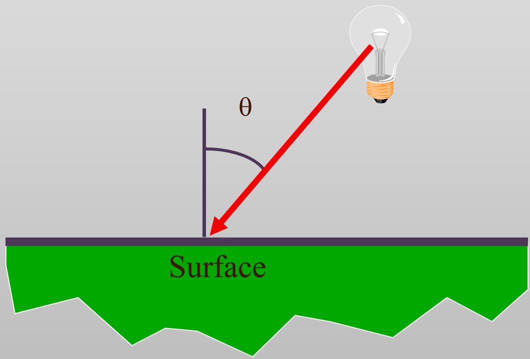


- *What does the reflected intensity depend on?*

Diffuse Reflection

How much light is reflected?

- Depends on angle of incident light

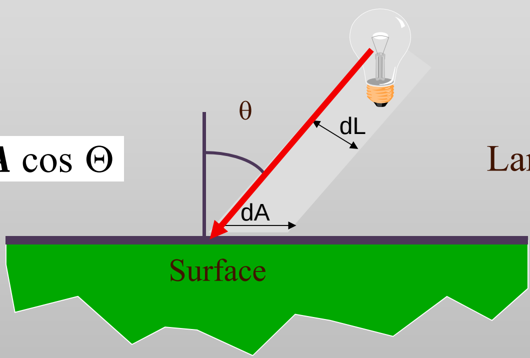


Diffuse Reflection

How much light is reflected?

- Depends on angle of incident light

$$dL = dA \cos \Theta$$



Lambert's Law

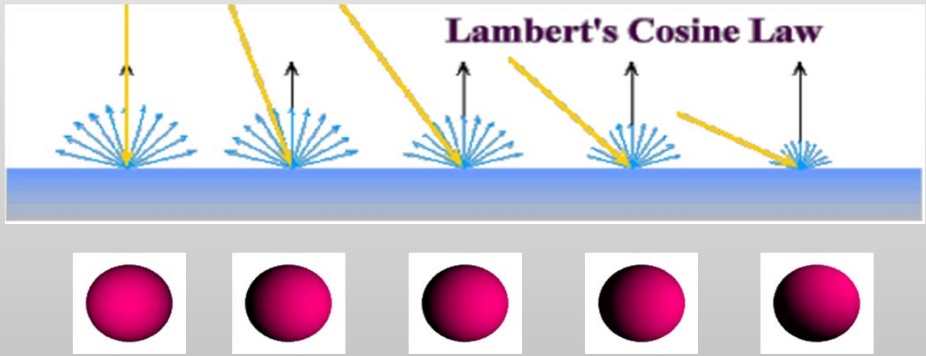
Lambert's Law

Ideal diffuse surfaces reflect according to Lambert's cosine law:

The energy reflected by a small portion of a surface from a light source in a given direction is proportional to the cosine of the angle between that direction and the surface normal

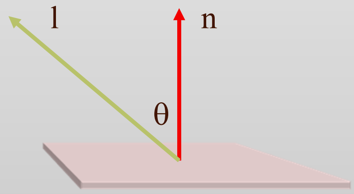
Note that the reflected intensity is independent of the viewing direction, but does depend on the surface orientation with regard to the light source

Lambert's Law



Lambert’s Law

The angle between the surface normal and the incoming light is the angle of incidence:



$$I_{diffuse} = k_d \, I_{light} \cos \theta$$

In practice we use vector arithmetic:

$$I_{diffuse} = k_d \, I_{light} \; (n \bullet l)$$

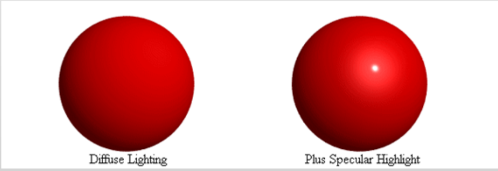
Lambert’s Law

（Utah Teapot）

Specular Reflection

Shiny surfaces exhibit specular reflection

- Polished metal
- Glossy car finish



A light shining on a specular surface causes a bright spot known as a specular highlight

Where these highlights appear is a function of the viewer's position, so specular reflectance is view dependent

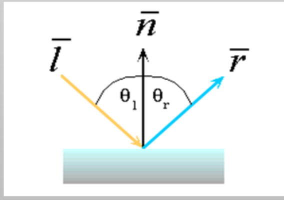
The Physics of Specular Reflection

- At the microscopic level a specular reflecting surface is very smooth
- Thus rays of light are likely to bounce off the microgeometry in a mirror-like fashion
- The smoother the surface, the closer it becomes to a perfect mirror

The Optics of Specular Reflection

Reflection follows Snell's Laws:

- The incoming ray and reflected ray lie in a plane with the surface normal
- The angle that the reflected ray forms with the surface normal equals the angle formed by the incoming ray and the surface normal:

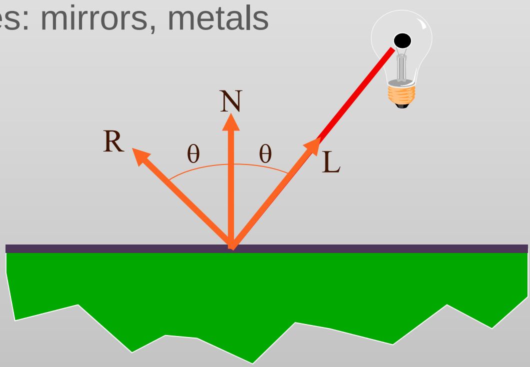


$$\theta_{(l)ight} = \theta_{(r)eflection}$$

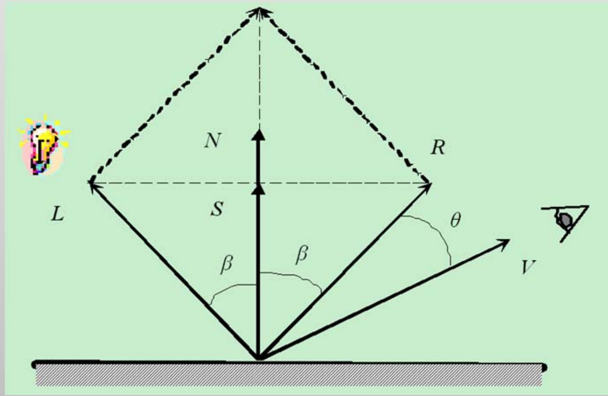
Specular Reflection

Reflection is strongest near mirror angle

- Examples: mirrors, metals



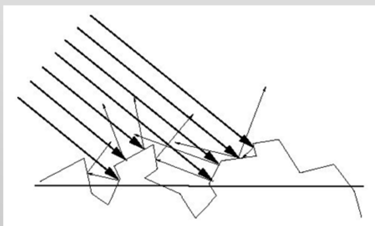
Geometry of Reflection



Non-Ideal Specular Reflectance

Snell's law applies to perfect mirror-like surfaces, but aside from mirrors, few surfaces exhibit perfect specularity

How can we capture the "softer" reflections of surface that are glossy rather than mirror-like?



Non-Ideal Specular Reflectance: An Empirical Approximation

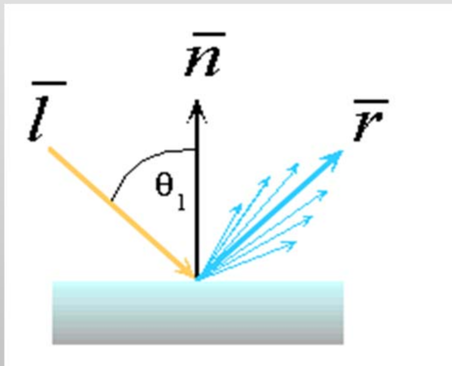
Hypothesis: most light reflects according to Snell's Law

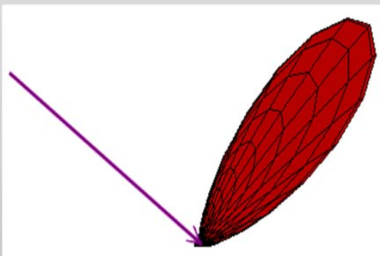
- But because of microscopic surface variations, some light may be reflected in a direction slightly off the ideal reflected ray

as we move from the ideal reflected ray, some light is still reflected

Non-Ideal Specular Reflectance: An Empirical Approximation

An illustration of this angular falloff:



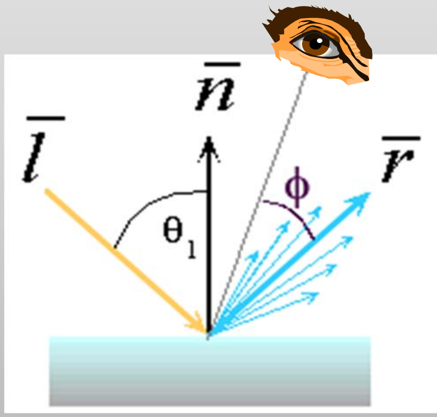


How might we model this falloff?

Phong lighting model

The most common lighting model in computer graphics was suggested by Phong:

$$I_{specular} = k_s I_{light} (\cos \phi)^{n_{shiny}}$$

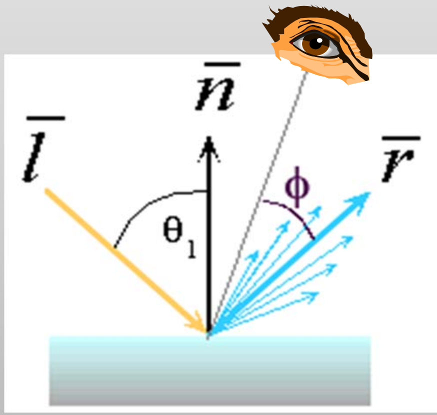


k_s: Material surface reflectance
I_{light}: Intensity of the light source
Φ: The angle between the ideal reflectance direction and viewer

Phong lighting model

The most common lighting model in computer graphics was suggested by Phong:

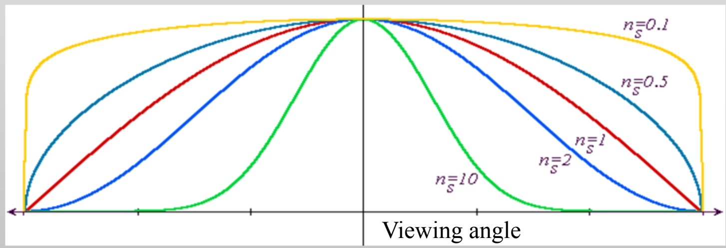
$$I_{specular} = k_s I_{light} (\cos \phi)^{n_{shiny}}$$



The n_{shiny} term is a purely empirical constant that varies the rate of falloff. Though this model has no physical basis, it works (sort of) in practice

Phong Lighting: The n_{shiny} Term

This diagram shows how the Phong reflectance term drops off with divergence of the viewing angle from the ideal reflected ray:

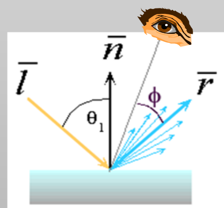


Calculating Phong Lighting

The cos term of Phong lighting can be computed using vector arithmetic:

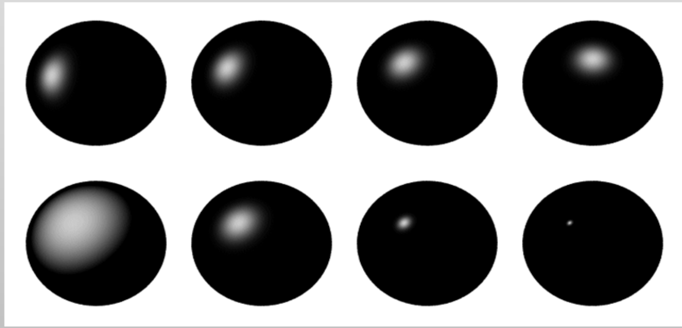
$$I_{specular} = k_s I_{light} \left(\bar{v} \cdot \bar{r} \right)^{n_{shiny}}$$

- v is the unit vector towards the viewer
- r is the ideal reflectance direction



Phong Examples

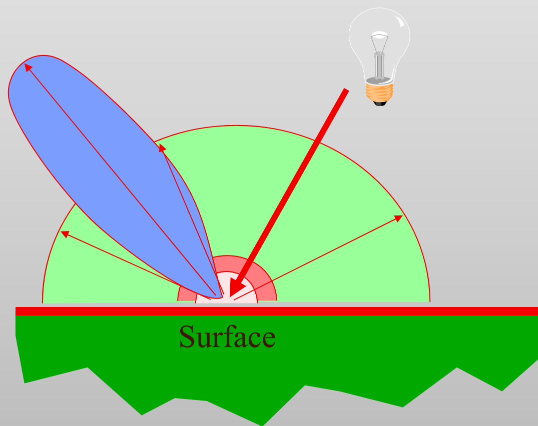
These spheres illustrate the Phong model as I and n_{shiny} are varied:



Combining Everything

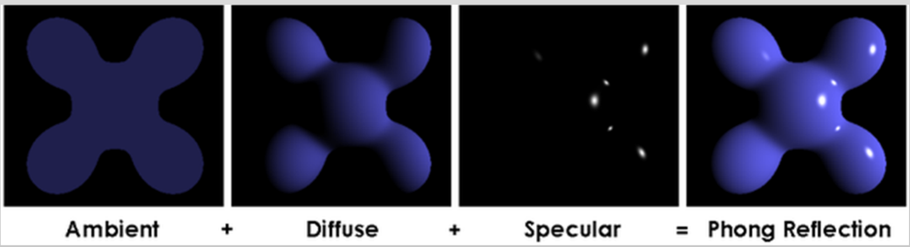
Simple analytic model:

- diffuse reflection +
- specular reflection +
- emission +
- “ambient”



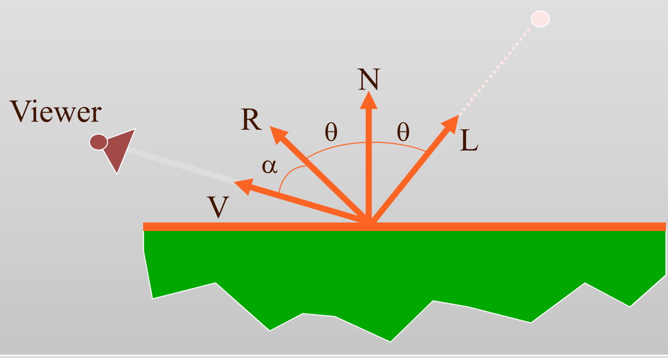
Combining Everything

Sum diffuse, specular, emission, and ambient



The Final Combined Equation

Single light source:

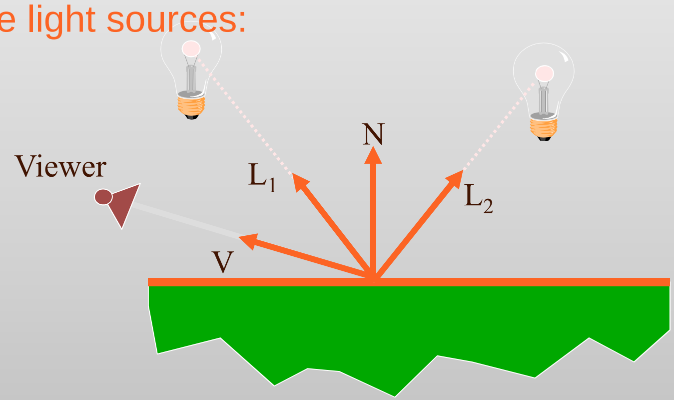


$$I = I_E + K_A I_{AL} + K_D (N \bullet L) I_L + K_S (V \bullet R)^n I_L$$

emission ambient diffuse specular

Final Combined Equation

Multiple light sources:



$$I = I_E + K_A I_{AL} + \sum_i (K_D (N \bullet L_i) I_i + K_S (V \bullet R_i)^n I_i)$$

emission ambient diffuse specular

Phong Light Model

（Utah Teapot）

OpenGL Function

void **glMaterialfv**(GLenum *face*, GLenum *pname*, const GLfloat * *params*);

face: Specifies which face or faces are being updated. Must be one of GL_FRONT, GL_BACK, or GL_FRONT_AND_BACK

pname: Specifies the material parameter of the face or faces that is being updated. Must be one of GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR, GL_EMISSION, GL_SHININESS, GL_AMBIENT_AND_DIFFUSE, or GL_COLOR_INDEXES.

Params: parameter values.

void **glNormalfv**(normal[]);

Light properties

void glLightfv(GLenum *light*, GLenum *pname*, GLfloat **params*);

light: Number ID of light

pname: GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR, GL_POSITION

GL_SPOT_DIRECTION, GL_SPOT_EXPONENT

GL_SPOT_CUTOFF

GL_CONSTANT_ATTENUATION
GL_LINEAR_ATTENUATION
GL_QUADRATIC_ATTENUATION