Illumination & Shading
Goals

Introduce the types of light-material interactions

Build a simple reflection model---the Phong model---that can be used with real time graphics hardware
Suppose we build a model of a sphere using many polygons and color it with an assigned color…we get:

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

\[
L_o(x, \omega_o, \lambda, t) = L_e(x, \omega_o, \lambda, t) + \int_\Omega f_r(x, \omega_i, \omega_o, \lambda, t) L_i(x, \omega_i, \lambda, t) (\omega_i \cdot n) \, d\omega_i
\]

Cannot be solved in general

Ray tracing is a special case for perfectly reflecting surface
Global Lighting/Rendering Effects

- Shadow
- Multiple reflection
- Translucent surface
Local vs. Global Lighting

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

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

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

Many techniques for approximating global effects
Step 1: Light Sources

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

Point source
Model with position and color

Directional (Distant) source = infinite distance away (parallel)

Ambient light
Same amount of light everywhere in scene
Models contribution of many sources and reflecting surfaces
Step 2: Light-Material Interactions

Light that strikes an object is partially absorbed and partially scattered (reflected)

The **wavelength** and **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
Surface Properties - Smoothness

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.
Phong Illumination Model

A simple model that can be computed rapidly has three components

- **Diffuse** – evenly distr. (rough surface)
- **Specular** – mirror (smooth surface)
- **Ambient** – global light

Requires four vectors

- To source
- To viewer
- Normal
- Perfect reflector
Phong Illum – Modeling Light Sources

Each light source
   diffuse
   specular
   ambient

Separate red, green and blue components

Hence, 9 coefficients for each point source

\[ L_{dr}, L_{dg}, L_{db}, L_{sr}, L_{sg}, L_{sb}, L_{ar}, L_{ag}, L_{ab} \]
Phong Illumination – Material Properties

Material properties determine how much light is absorbed and how much reflected and in which directions.

Material property for each wavelength, red, green and blue

\[ R_{dr}, R_{dg}, R_{db}, R_{sr}, R_{sg}, R_{sb}, R_{ar}, R_{ag}, R_{ab} \]
Phong Illumination Equation

Total red intensity

\[ I_r = \sum_{i} R_{ra} L_{ira} + R_{rd} L_{ird} + R_{rs} L_{irs} \]

Intensity of red ambient light (or how much red is coming in)

Reflectivity of red ambient material (or how much red is going back out)
Ambient Reflection

All the other light
Intensity is the same at all points on surface
Not all light is reflected – some is absorbed

\[ R_a = k_a \]

\[ I_a = k_a L_a \quad \text{(for each } r, g, b) \quad 0 < k < 1 \]
Diffuse Reflection

In a perfectly diffuse reflector (Lambertian Surface), light is scattered equally in all directions

appears same to all viewers!!

Amount of light reflected is proportional to the vertical (normal) component of incoming light
Diffuse Reflection

Amount of light energy that falls on a surface area element (and is reflected) is proportional to the cosine of the angle of incidence of the light

\[ R \sim \cos \theta_i \]

\[ \cos \theta_i = \mathbf{l} \cdot \mathbf{n} \] if vectors normalized

There are also three coefficients, \( k_r, k_b, k_g \) that show how much of each color component is reflected vs. absorbed

\[ I_d = k_d (\mathbf{l} \cdot \mathbf{n}) L_d \]
Most surfaces are neither ideal diffusers nor perfectly specular (ideal reflectors)

Smooth surfaces show specular highlights due to incoming light being reflected in directions concentrated close to the direction of a perfect reflection
Phong proposed using a term that dropped off as the angle between the viewer and the ideal reflection increased.

\[
I_s = k_s L_s \cos^\alpha \phi (v \cdot r)^\alpha
\]

- Reflectivity coef
- Shininess coef
- Reflected intensity
- Incoming intensity
- Reflectivity coef
Shininess Coefficient

Values of $\alpha$ between 100 and 200 correspond to metals

Values between 5 and 10 give surface that look like plastic
Computing Reflection Vector, $r$

Normal is determined by local orientation

Angle of incidence = angle of reflection

The three vectors must be coplanar

$$r = 2 \left( l \cdot n \right) n - l$$
Modified Phong Model

The specular term in the Phong model is problematic because it requires the calculation of a new reflection vector and view vector for each vertex.

Blinn suggested an approximation using the halfway vector that is more efficient.
The Halfway Vector

\( H \) is normalized vector halfway between \( L \) and \( V \).

Use its relationship to \( N \) to approximate \( \text{dot}(V,R) \).

Highlight occurs when \( H \) is near \( N \).

Note \( \beta = \frac{1}{2} \alpha \), so we need a new exponent.

\[
H = \frac{L + V}{|L + V|}
\]

Replace \((V \cdot R)^\alpha\)

with \((N \cdot H)^p\)
Putting it all together…

For each light source and each color component, the Phong model can be written as:

\[ I = k_d L_d \cdot n + k_s L_s (v \cdot r)^\alpha + k_a L_a \]

For each color component we add contributions from all sources.
Example

Only differences in these teapots are the parameters in the Phong model.
Lighting Rules of Thumb

Set light’s ambient term to 0

Set light’s diffuse and specular components to full color of the light (typically white light)

Set each material’s diffuse and ambient to full color of the object

Set material’s specular to some amount of white