**Illumination Models**

Test: surface reflection illuminates toy astronaut in shadow. In second image, foreground reflection is reduced with black paper.

Test: slanted surface alters shadow direction on Moon-rock model under one light. Identical ground slant can be detected in Moon photo on left.
Local Illumination

Example

Local Illumination in the projective rendering pipeline

Local Illumination
- only models light arriving directly from light source
  - interreflections and shadows
    - added through tricks, multiple rendering passes

Types of Models
- Simple, non-physical reflection models (Phong, Blinn)
- physically-based reflection models
  - BRDFs: Bidirectional Reflection Distribution Functions
Light Sources

Types of light sources
- `glLightfv(GL_LIGHT0, GL_POSITION, light[]))`
- Directional/parallel lights
  - E.g. sun
  - Homogeneous vector
- (Homogeneous) point lights
  - Same intensity in all directions
- Spot lights
  - Limited set of directions:
    - Point+direction+cutoff angle

Local Illumination

Commonly used model (simple, non-physical)

Combine diffuse, specular, ambient
- E.g. OpenGL / graphics hardware:

\[ I_{out}(x) = k_a \cdot I_a + k_d \cdot (\mathbf{l} \cdot \mathbf{n}) \cdot I_{diff} + k_s \cdot (\mathbf{h} \cdot \mathbf{n})^n \cdot I_{spec} \]

Materials

Ambient Light
- Incoming light component that is identical everywhere in the scene
- No direction
- Hack for replacing true global illumination (light bouncing off from other objects)

\[ I_{out}(x) = k_a \cdot I_a \]
**Diffuse component: Lambert’s Law**

**Johann Friedrich Lambert (1783):**
- Power per unit area arriving at some object point \( x \) also depends on the angle of the surface to the light direction
  - \( d\Omega \): differential surface area surrounding \( x \)
  - \( \mathbf{l} \): light direction (unit length!)

\[
I'(x) = \cos(\angle(n, \mathbf{l})) \cdot I_{in}(x) = (n \cdot \mathbf{l}) \cdot I_{in}(x)
\]

**Diffuse Component:**
- a more detailed look

- independent of viewing direction
Materials

Specular/Glossy
- Light is mostly reflected into the directions around the mirror direction $r_i$ of $l$

Phong Model (Phong Bui-Tuong, 1975)
- Use cosine power as heuristic

$$I_{spec} = k_s \cdot (v \cdot r_i)^n \cdot I_{in}$$

Phong model
- Polar plot

$$r_i = \frac{2(n \cdot l)}{n - l}$$
**Materials**

**Blinn-Phong model (Jim Blinn, 1977)**
- Variation with better physical interpretation
  - $h$: halfway vector; $n$: shininess

$$I_{out}(x) = k_s \cdot (h \cdot n)^n \cdot I_{in}(x); \text{ with } h = (I + v) / 2$$

**Commonly used model (simple, non-physical)**

**Combine diffuse, specular, ambient**
- E.g. OpenGL / graphics hardware:

$$I_{out}(x) = k_a \cdot I_a + k_d \cdot (I \cdot n) \cdot I_{diff} + k_s \cdot (h \cdot n)^n \cdot I_{spec}$$

**Lighting in OpenGL**
- Light source: amount of RGB light emitted
  - value represents percentage of full intensity, e.g., (1.0,0.5,0.5)
  - every light source emits ambient, diffuse, and specular light
- Materials: amount of RGB light reflected
  - value represents percentage reflected e.g., (0.0,1.0,0.5)
Lighting in OpenGL

```c
glLightfv(GL_LIGHT0, GL_AMBIENT, amb_light_rgba);
glLightfv(GL_LIGHT0, GL_DIFFUSE, dif_light_rgba);
glLightfv(GL_LIGHT0, GL_SPECULAR, spec_light_rgba);
glLightfv(GL_LIGHT0, GL_POSITION, position);
glEnable(GL_LIGHT0);

glMaterialfv( GL_FRONT, GL_AMBIENT, ambient_rgba );
glMaterialfv( GL_FRONT, GL_DIFFUSE, diffuse_rgba );
glMaterialfv( GL_FRONT, GL_SPECULAR, specular_rgba );
glMaterialfv( GL_FRONT, GL_SHININESS, n );
```

Flat Shading

```
Example:  
```

Gouraud Shading

```
Example:  
```

Example:
**Materials**

**Summary**
- Very simple reflection models
- Fast (dot products & exponentiation)
- No physical justification
- Not very good for modeling real world

**Light Sources**

**Quadratic falloff**
- Brightness of objects depends on power per unit area that hits the object
- The power per unit area for a point or spot light decreases quadratically with distance

\[ \text{Area } 4\pi r^2 \]
\[ \text{Area } 4\pi (2r)^2 \]

**Light Sources**

**Non-quadratic falloff:**
- Many systems allow for other fallofks
- Allows for faking of the effect of area light sources
- OpenGL / graphics hardware:
  - \( I_o \): intensity of light source
  - \( x \): object point
  - \( r \): distance of light from \( x \)

\[ I_{in}(x) = \frac{1}{ar^2 + br + c} \cdot I_0 \]

**Materials**

**Bi-directional Reflectance Distribution Function (BRDF):**
- Describes fraction of light reflected for all combinations of incoming (light) and outgoing (viewing) directions
- Color channels (R, G, B) are treated separately
  - Actually: wavelengths (see later in course)
**Materials**

**Bi-directional Reflectance Distribution Function (BRDF):**

\[ I_{out}(x) = f_r(\phi_{in}, \theta_{in}, \phi_{out}, \theta_{out}) \cdot I'_{in}(x) \]

\[ = f_r(l \rightarrow v) \cdot (n \cdot l) \cdot I_{in}(x) \]

- \( f_r(l \rightarrow v) \) is called BRDF
- \((t,n,b)\) is local coordinate frame (normal, tangent, binormal)

**Polar plot of BRDF**

- Fix incoming light direction \( l \)
- Plot \( f_r(l \rightarrow v) \cdot v \) for all viewing directions \( v \)
- Works for 2D and 3D plots
- Example: 2D polar plot for diffuse BRDF

**Light Sources**

**Area lights:**

- light sources with a finite area
- more realistic model of many light sources
- Not available with projective rendering pipeline, (i.e., not available with OpenGL)

**Gouraud Shading**

**Mach Bands:**

- Eye enhances discontinuity in first derivative
- Very disturbing, especially for highlights