Chapter 7

Illumination Models & Shading

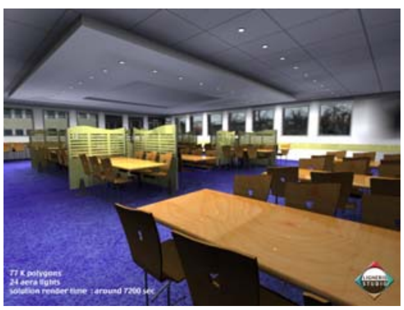
University of British Columbia

Shading Models

- Realistic interaction of light and objects
  - Simulate physical phenomena
- Fast - Fake it!!!
  - Ignore real physics, approximate the look
- Physically based reflection models
  - BRDFs: Bidirectional Reflection Distribution Functions

University of British Columbia

Photorealistic Illumination



[electricimage.com]

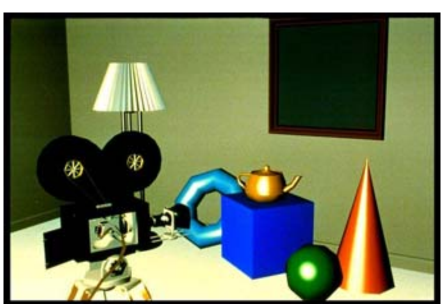
University of British Columbia

Local vs. Global Illumination Models

- Local model - interaction of each object with light
- Global model: interactions between objects

University of British Columbia

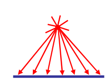
Fast Local Illumination



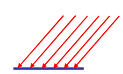
University of British Columbia

Light Sources

- Point source
  - light originates at a point
  - Rays hit planar surface at different angles



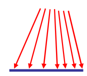
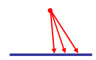
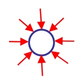
- Parallel source
  - light rays are parallel
  - Rays hit a planar surface at identical angles
  - May be modeled as point source at infinity
  - *Directional light*



University of British Columbia

Light Sources

- Area source
  - Light originates at finite area in space.
  - In-between point and parallel sources
- spotlights
  - position, direction, angle
- ambient light (environment light)

University of British Columbia

Light Sources - OpenGL

- Specify parameters

  glLightfv(GL_LIGHTi,GL_POSITION,light[])

  i – between 0 & 8 (or more)
- Directional $[x \quad y \quad z \quad 0]$
- Point source $[x \quad y \quad z \quad 1]$
- Spotlight has extra parameters:
  - GL_SPOT_DIRECTION, GL_SPOT_EXPONENT, GL_SPOT_CUTOFF
- Area source – too complex for projective pipeline (e.g. OpenGL)

University of British Columbia

Ambient Light

- non-directional light – environment light
- Object illuminated with same light everywhere
  - Looks like silhouette
- Illumination equation $I = I_a k_a$
  - I_a - ambient light intensity
  - k_a- fraction of this light reflected from surface
  - Defines object color

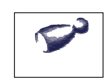
University of British Columbia

Light

- Light has color
- Interacts with object color (r,g,b)

$$I = I_a k_a$$
$$I_a = (I_{ar}, I_{ag}, I_{ab})$$
$$k_a = (k_{ar}, k_{ag}, k_{ab})$$
$$I = (I_r, I_g, I_b) = (I_{ar}k_{ar}, I_{ag}k_{ag}, I_{ab}k_{ab})$$

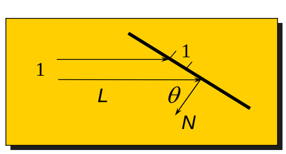
- Blue light on white surface?
- Blue light on red surface?



University of British Columbia

Diffuse Light

- Dull surfaces - such as solid matte plastic- reflect uniformly in all directions
- This is called diffuse or Lambertian reflection
- For light source normalized direction L & surface with normal N reflected light is proportional to LN



University of British Columbia

Diffuse Reflection

- Illumination equation is now:

$$I = I_a k_a + I_p k_d (N \cdot L) = I_a k_a + I_p k_d \cos\theta$$

- I_p - point/parallel source's intensity
- k_d - surface diffuse reflection coefficient

- Can we locate light source from shading?

University of British Columbia

Diffuse Reflection

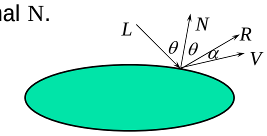
- Multiple lights I_p with directions L_p

$$I = I_a k_a + \sum_p I_p k_d (N \cdot L_p) = I_a k_a + \sum_p I_p k_d \cos\theta_p$$

University of British Columbia

Specular Reflection

- Shiny objects (e.g. metallic) reflect light in preferred direction R determined by surface normal N.



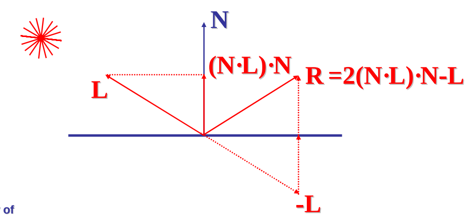
- Most objects are not ideal mirrors - reflect in the immediate vicinity of R

University of British Columbia

Specular

Phong Model (Phong Bui-Tuong, 1975)

- Assume exponential attenuation of form $\cos^n \alpha$
- Computing reflection direction **R** of **L**
- ***N*** and ***L*** are unit length!



University of British Columbia

Specular Reflection (Phong Model)

- Illumination equation:

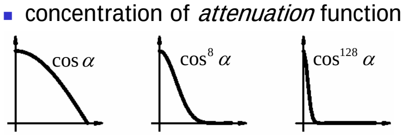
$$I = I_a k_a + I_p (k_d (N \cdot L) + k_s (R \cdot V)^n)$$

- k_s - Specular reflection coefficient
- n - Specularity exponent

University of British Columbia

Specular Reflection (cont'd)

- Exponent n of cosine controls
- concentration of *attenuation* function:



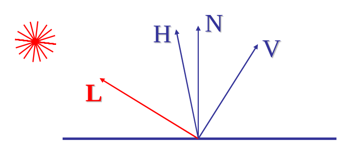
- No physical basis BUT looks good

University of British Columbia

Specular

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

$$I_{out}(\mathbf{x}) = k_s \cdot (H \cdot N)^n \cdot I_{in}(\mathbf{x}); \text{ with } H = (L+V)/2$$



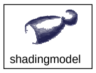
University of British Columbia

Illumination Equation

- For multiple light sources:

$$I = I_a k_a + \sum_p \frac{I_p}{d_p^2}(k_d(N \cdot L_p) + k_s(R_p \cdot V)^n)$$

- d_p- distance between surface and light source + distance between surface and viewer (Heuristic atmospheric attenuation)



University of British Columbia

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)

University of British Columbia

In OpenGL

- k_a, k_d, k_s - surface color (RGB)
- Modify by glMaterialfv(GL_FRONT_AND_BACK , *pname*, RGB[])
- *pname* - GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR
- Light source properties (also RGB) glLightfv(GL_LIGHTi,*pname*,light[])

University of British Columbia

Lighting in OpenGL

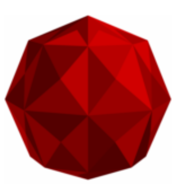
```
glLightfv(GL_LIGHT0, GL_AMBIENT, amb_light_rgba );
glLightfv(GL_LIGHT0, GL_DIFFUSE, dif_light_rgba );
glLightfv(GL_LIGHT0, GL_SPECULAR, spec_light_rgba );
glLightfv(GL_LIGHT0, GL_POSITION, position);
glEnable(GL_LIGHT0);

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 );
```

University of British Columbia

Flat Shading

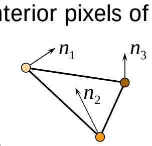
- Illumination value depends only on polygon normal
  - each polygon colored with uniform intensity
- Not adequate for polygons approximating smooth surface
- Looks non-smooth
  - worsened by Mach bands effect

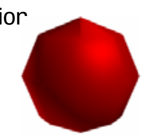


University of British Columbia

Gourard Shading

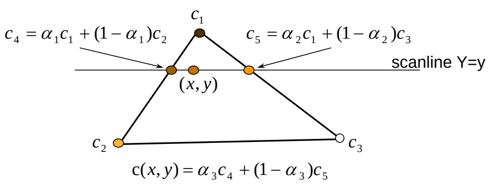
- Polyhedron - approximation of smooth surface
  - Assign to each vertex normal of original surface at point
  - If surface not available use estimate normal
- Compute illumination intensity at vertices using those normals
- Linearly interpolate vertex intensities over interior pixels of polygon projection





University of British Columbia

Gourard Shading (cont'd)

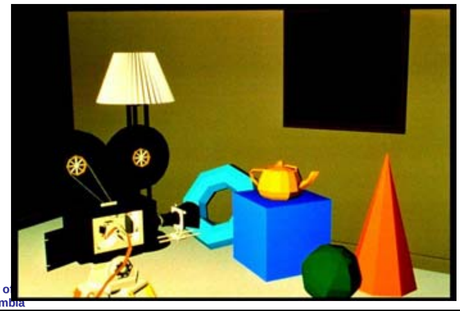


- Assign pixel color during scan conversion
- Can Gourard shading support specular reflection ?

University of British Columbia

Flat Shading

- Example:



University of British Columbia

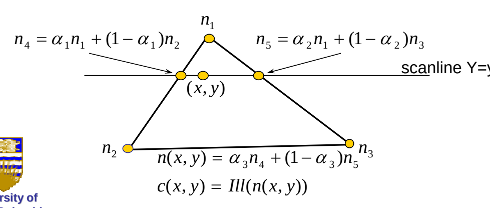
Gouraud Shading

- Example:

University of British Columbia

Phong Shading

- Interpolate (in image space) normal vectors instead of intensities
- Apply illumination equation for each interior pixel with its own normal



University of British Columbia

Shading

- Phong shading is clearly more expensive (why ?)- but well worth the effort
- Can achieve specular effects
- Both Gourard & Phong schemes are performed in the image plane ⇒ view dependent
- Can cause artifacts during animation

University of British Columbia

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)

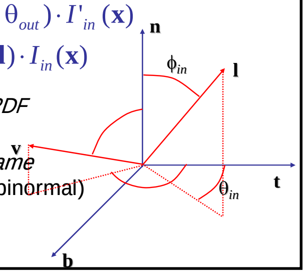
University of British Columbia

Materials

- Bi-directional Reflectance Distribution Function (BRDF):

$$I_{out}(\mathbf{x}) = f_r(\phi_{in}, \theta_{in}, \phi_{out}, \theta_{out}) \cdot I'_{in}(\mathbf{x})$$
$$= f_r(\mathbf{l} \to \mathbf{v}) \cdot (\mathbf{n} \cdot \mathbf{l}) \cdot I_{in}(\mathbf{x})$$

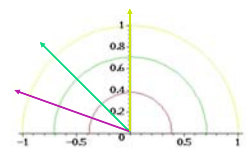
- $f_r(\mathbf{l} \to \mathbf{v})$ is called *BRDF*
- $(\mathbf{t},\mathbf{n},\mathbf{b})$ is *local coordinate frame* (normal, tangent, binormal)



University of British Columbia

Materials

- Polar plot of BRDF
  - Fix incoming light direction $\mathbf{l}$
  - Plot $f_r(\mathbf{l} \to \mathbf{v}) \cdot \mathbf{v}$ for all viewing directions $\mathbf{v}$
  - Works for 2D and 3D plots
  - Example: 2D polar plot for diffuse BRDF



University of British Columbia