Shading

Compute intensity at each pixel based on light sources mitigated by many things.

\[ I_{\text{Total}} = \frac{m_1 I_p + m_2 I_p + \ldots + m_n I_p + m_a I_a}{\text{each point source ambient light}} \]

\( m \) - mitigating coeff.

Too hard to use area light sources (mapped as point light sources)

Points are at infinite distance (parallel rays) for direction only

Infinite distance - caimes merge to single value

\( m_a I_a \) - cheat for the inter-reflection (like light reflected off the walls, floors, etc...)

Real intense reflections are modeled as point sources

\( m_a I_a \) - objects "glow" - artificial

- Set arbitrarily \( \to \) gets tweaked by visual observation

"It's hard to recognize porn, but I know it when I see it..."

Review equations for Intensity, reduction for next class
Shading Basic Illumination Model

\[ I_r = I_a k_a O_k + I_p k \cos(\alpha) \text{fatt} \text{OR} \]

( also \( I_a, I_p \))

- \( I_a \) - ambient (fake thing "glow")
- \( I_p \) - point light source
- \( k \) - material property
  - mitigating coefficient \( 0 \leq k < 1 \)
  - smooth close to 1, pitted (scatter) less than 1
  - determined empirically
- \( \cos \alpha \) - angle of light from light sources
- \( \text{fatt} \) - attenuation coefficient \( 0 < \text{fatt} < 1 \)
  - (light intensity diminishes as squared dist)
  - \( \frac{1}{d^2} \) but too steep for aesthetics, play with \( A, B \) to adjust

\( O_{R,k} \) - diffuse color coefficient \( 0 \leq 1 \)
  - \( R \) and one of the primary colors that is the color of the object
  - red obj under green light \( \rightarrow \) black obj
  - \( O_{R,k} = 1 \) for red object \( (O_a \cdot O_g = \Phi) \)

This is why we use 3-byte color
Do this after clipping / visible surface determination

If done properly, shadows will occur naturally (ray tracing)
- point light sources will be blocked

Special effects impact
- Specular reflection (shiny apple example)
  - Shiny surface - all light reflected at point
  - around light angle \( \cos(\alpha) \) close to 1
  - material property \( n \) (shinier object \( \rightarrow \) small n)
  - \( \cos(\alpha) \)
  - choose a limit for spec reflection \( \text{e.g.} \cos(\alpha) > 0.9 \)
  - must occur under white light.
  - Phong model (basis on empirical data)
  - apply this first, use it instead of Basic III model \( (I_a \rightarrow I_{ap}) \)
Polygon Shading - (interpolated shading)
- hard to make edges go away
  - calculated once - use for all facets
  \[ I_p \]
  - only thing changes is \( \cos(\alpha) \)
    - facets will be treated constantly across extent
  - need to interpolate across polygon surface
    - average surface normal at shared vertices

Two methods

Gouraud shading
- only calculate \( I \) for each vertex
- interpolate down each edge \( I_2 \) double interpolation
- interpolate along scan line \( I_3 \) interpolation

\( \Rightarrow \) lots "cheaper" than computing each pixel using BIM

Problems:
  shading of surface will change just by rotation
  - attitude problem

\( \Rightarrow \) Vertex illumination calculation must be done before clipping, back face culling
  - need this info to perform interpolations (same for Phong)

Phong shading - more calc for each pixel
- interpolate surface normal values instead of Illuminations
  and use BIM

Shading is hard, but necessary, for realism

"mock-banding for eye"

non-representational

"half vertex"