Shading

- Shading models
- Approximate light reflection on illuminated surfaces
Shading model

- Compute light reflected toward camera / eye
  - $v$ - view direction
  - $l$ - light direction
  - $n$ - surface normal
  - surface parameters
    - color, shininess, etc.

$v, l, n$ must be normalized!!
Diffuse

- Light scattered uniformly in all directions
- surface color same for all viewing directions

Diffuse

- Lambertian shading model
- energy from light source depends on angle to light source
  - max illumination - surface directly toward light source
  - min illumination - surface tangent to light source
**Diffuse**

- Lambert’s cosine law

  Top face of cube receives a certain amount of light
  Top face of 60° rotated cube intercepts half the light
  In general, light per unit area is proportional to \( \cos \theta = L \cdot n \)

---

**Diffuse**

- Lambertian shading
- Shading independent of view direction

\[
L_d = k_d I \max(0, n \cdot l)
\]

- \( L_d \) - diffusely reflected light
- \( k_d \) - diffuse coefficient
- \( I \) - illumination from source
Diffuse

- Lambertian shading
  - matte appearance

\[ k_d \]

Diffuse

- Light scattered uniformly in all directions
  - surface color same for all viewing directions
Specular

- Blinn-Phong
  - Intensity depends on view direction
  - Highlights

Specular

- Blinn-Phong
  - reflection brightest when \( \mathbf{v} \) and \( \mathbf{i} \) are symmetric across surface normal
Specular

- Blinn-Phong
  - $L_s$ - specularly reflected light
  - $k_s$ - specular coefficient
  - $p$ - Phong exponent $> 1$

$$h = \text{bisector}(v, l) \quad L_s = k_s I \max(0, \cos \alpha)^p$$

$$= \frac{v + l}{\|v + l\|} \quad = k_s I \max(0, n \cdot h)^p$$

Fig. 16.9 Different values of $\cos^p \alpha$ used in the Phong illumination model.
Specular

$\mathbf{L} = k_d \mathbf{I}_{max}(0, \mathbf{n} \cdot \mathbf{l}) + k_s \mathbf{I}_{max}(0, \mathbf{n} \cdot \mathbf{h})^p$

**Diffuse + Specular**
Specular

- Lambertian
  - view independent
- Blinn-Phong
  - view dependent

Ambient

- Independent of everything
  - add constant color
  - fill in black shadows

\[ L_a = k_a I_a \]

- \( L_a \) - reflected ambient light
- \( k_a \) - ambient coefficient
• Ambient + diffuse + specular

\[ L = L_a + L_d + L_s \]
\[ = k_a I_a + k_d I \max(0, \mathbf{n} \cdot \mathbf{l}) + k_s I \max(0, \mathbf{n} \cdot \mathbf{h})^p \]

---

**Demo**

• Your task
  - Create a 5x5x5 array of spheres
  - Modify specular, diffuse, and ambient properties along each axis
  - In DropBox create a folder named “ClassWork” and submit a screenshot of your spheres