Why we need shading

• Suppose we build a model of a sphere using many polygons and color each the same color. We get something like

• 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 (normal)
The most general approach is based on physics - using principles such as conservation of energy. A surface either emits light (e.g., light bulb) or reflects light for other illumination sources, or both. Light interaction with materials is recursive. The rendering equation is an integral equation describing the limit of this recursive process.
Fast local shading models

- the rendering equation can't be solved analytically
- numerical methods aren't fast enough for real-time
- for our fast graphics rendering pipeline, we'll use a local model where shade at a point is independent of other surfaces
- use Phong reflection model
  - shading based on local light-material interactions
Local shading model
Global Effects

shadow

multiple reflection

translucent surface

[Angel and Shreiner]
Light-material interactions

at a surface, light is absorbed, reflected, or transmitted

specular

diffuse

translucent
General light source

Illumination function:

\[ l(x, y, z, \theta, \phi, \lambda) \]

integrate contributions from all sources to shade the point
Idealized light sources

- Ambient light
- Point light
- Spotlight
- distant (directional) light

luminance: \( L = \begin{bmatrix} L_r \\ L_g \\ L_b \end{bmatrix} \)
Ambient light source

- achieve a uniform light level
- no black shadows
- ambient light intensity at each point in the scene

\[ L_a = \begin{bmatrix} L_{ar} \\ L_{ag} \\ L_{ab} \end{bmatrix} \]
Point light source

\[ L(p_0) = \begin{bmatrix} L_r(p_0) \\ L_g(p_0) \\ L_b(p_0) \end{bmatrix} \]

illumination intensity at \( p \):

\[ l(p, p_0) = \frac{1}{|p - p_0|^2} L(p_0) \]
Point light source

Most real-world scenes have large light sources

Point light sources alone aren’t too realistic
- add ambient light to mitigate high contrast
Point light source

Most real-world scenes have large light sources

Point light sources alone aren’t too realistic
- drop off intensity more slowly

\[ l(p, p_0) = \frac{1}{d^2} L(p_0) \]

\[ l(p, p_0) = \frac{1}{a + bd + cd^2} L(p_0) \]
Spotlights

\[ \cos(\phi) \]
Spotlights

\[ \cos^e(\phi) \ l(p, p_s) \]
Distant light source characterized by direction
Lambertian Reflection Model
Lambertian Reflection Model

\[ I \propto \cos \theta \]

**color intensity**

Lambert’s cosine law

**direct**: maximum light intensity

**indirect**: reduced light intensity
Lambertian Reflection Model

\[ I \propto n \cdot l \]

color intensity

**Lambert’s cosine law**

- **Direct**: maximum light intensity
- **Indirect**: reduced light intensity
Lambertian Reflection Model

\[ I \propto Rn \cdot l \]

**direct**: maximum light intensity

**indirect**: reduced light intensity

**Lambert’s cosine law**
Lambertian Reflection Model

\[ I = LR \cdot n \cdot I \]

- **Illumination**
- **Color intensity**
- **Reflectance**

Lambert's cosine law:

- **Direct**: maximum light intensity
- **Indirect**: reduced light intensity
Lambertian Reflection Model

\[ I = LR \max(0, \mathbf{n} \cdot \mathbf{l}) \]
Lambertian Reflection Model

\[ I = LR |\mathbf{n} \cdot \mathbf{l}| \]

two-sided lighting
Ambient Reflection

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

Surfaces facing away from the light will be totally **black**
Ambient Reflection

\[ I = L_a R_a + L_d R_d \max(0, \mathbf{n} \cdot \mathbf{l}) \]

All surfaces get same amount of ambient light
Phong Reflection Model
Phong Reflection Model

- Efficient, reasonably realistic
- 3 components
- 4 vectors

Ambient + Diffuse + Specular = Phong Reflection Model

[Brad Smith, Wikimedia Commons]
Phong Reflection Model

\[ I = I_a + I_d + I_s \]
\[ = R_a L_a + R_d L_d \max(0, l \cdot n) + R_s L_s \max(0, \cos \phi)^\alpha \]
Ambient reflection

Different ambient coefficients for different colors

\[ I_a = R_a L_a, \quad 0 \leq R_a \leq 1 \]
Diffuse reflection

![Diagram showing ambient, diffuse, and specular components combining to form Phong reflection.](image)
Diffuse reflection

\[ I_d = R_d L_d \max(0, l \cdot n) \]

**Lambert’s cosine law**

- **Direct**: maximum light intensity
- **Indirect**: reduced light intensity
Specular reflection

Ideal reflector

\[ \theta_i = \theta_r \]

\( \mathbf{r} \) is the mirror reflection direction
Specular reflection is strongest in mirror reflection direction.
Specular reflection

specular reflection drops off with increasing angle $\phi$

$$I_s = R_s L_s \cos^{\alpha} \phi$$

specular reflection coefficient
Phong exponent
Specular reflection

\[ I_s = R_s L_s \max(0, \cos \phi)^\alpha \]

\( \alpha = 5..10 \quad \text{plastic} \)

\( \alpha = 100..200 \quad \text{metal} \)
Phong Reflection Model

\[ I = I_a + I_d + I_s \]
\[ = R_a L_a + R_d L_d \max(0, l \cdot n) + R_s L_s \max(0, v \cdot r)^\alpha \]

- Ambient
- Diffuse
- Specular
Alternative: Blinn-Phong Model

\[ h = \frac{l + v}{|l + v|} \]

\[ I = I_a + I_d + I_s \]
\[ = R_a L_a + R_d L_d \max(0, l \cdot n) + R_s L_s \max(0, h \cdot n)^\alpha \]

Ambient \hspace{1cm} Diffuse \hspace{1cm} Specular
$\alpha$

10: eggshell
100: shiny
1000: glossy
10000: mirror-like