CSE452 Computer Graphics

Lecture 10: Illumination
If there is no light...
What is light?

- Electromagnetic radiation that is visible to the human eye
  - Carried by “photons” but also exhibits wave behaviors
- Properties
  - Speed (constant)
  - Direction (straight)
  - Wavelength (“color”)
What is light?

- Electromagnetic radiation that is visible to the human eye
  - Carried by “photons” but also exhibits wave behaviors
- Properties
  - Speed (constant)
  - Direction (straight)
  - Wavelength (“color”)
  - Amplitude (“intensity”)

![Diagram showing the relationship between wavelength, amplitude, and intensity](image)
Where is purple or pink?

Intensity

Color
Where is purple or pink?
You may not have fully understood.

Important thing is “color is deep.”
How is a surface lit?

• By the (color and intensity of) light that is transmitted from the surface in the direction towards our eye
  – Active: the surface is emitting light (i.e., a light source)
  – Passive: the light originates from somewhere else
Passive Lighting

- When a photon hits a surface it may
  - Get absorbed (turned into heat or other energy)
Passive Lighting

- When a photon hits a surface it may
  - Get absorbed
  - Reflect
Passive Lighting

- When a photon hits a surface it may
  - Get absorbed
  - Reflect
  - Refract
Passive Lighting

- When a photon hits a surface it may
  - Get absorbed
  - Reflect
  - Refract
Passive Lighting

- When a photon hits a surface it may
  - Get absorbed
  - Reflect
  - Refract
  - Scatter
Passive Lighting

- When a photon hits a surface it may
  - Get absorbed
  - Reflect
  - Refract
  - Scatter

- What happens depends on:
  - The color and intensity of light
  - Color and material of surface
  - Orientation of surface with respect to the light source and eye

May change direction, color, and intensity
Illumination (in Computer Graphics)

• Given
  – Light sources, object surfaces and the camera

• Compute
  – Color of each pixel on the screen
    • As intensity of photons that come towards the camera in that viewing direction
Computer Representation

- **Light sources**
  - Directional light (e.g., the sunlight)
    - Emitting photons in one direction
  - Point/Area light (e.g., light bulb)
    - Emitting photons in all directions from a single source
  - Spot light (e.g., a flashlight)
    - Emitting photons from a single source forming a cone

- **Light attenuation**
  - Intensity falls off with distance
Computer Representation

• Surface
  – A geometric surface can be
    • Discrete: consisting of polygons (e.g., triangles), or
    • Continuous: parametric surface (e.g., the sphere)
  – Each surface element is locally represented by
    • The point location
    • The normal vector of the tangent plane
      – Discrete: polygon normal
      – Continuous: first derivatives
Local Illumination
Local Illumination

- Light interaction with one surface
  - Only considers direct reflection of the light from the source
  - Assuming the path between the light source and the surface is unblocked
- Pros: Fast
- Cons: Missing many effects
  - Shadow
  - Refraction
  - Multi-hop reflection
Global Illumination

- Light interaction with all surfaces
  - Reflection/refraction involving multiple surfaces
  - Considers shadows (when the path between the light source and surface is blocked)
    - Complete (*Umbra*) or incomplete (*Penumbra*)

- Pros: Realistic
- Cons: Expensive
Umbra/Penumbra from Sun
Local vs. Global Illumination
If you want to know more...

Light and Color Song by ParrMr
[http://www.youtube.com/watch?v=X1hlQvKbQDE]
Overview

- Local illumination
  - Local lighting model (this lecture)
  - Drawing polygonal models (lecture 11)

- Global illumination
  - Ray tracing (lecture 12, 13)
  - Radiosity (lecture 14)
Local Lighting Model

- Factors in computing reflected light:
  - Geometric configuration (between light source, surface and camera)
  - Light properties (source type, color, attenuation)
  - Surface material (color, shininess, etc.)
  - Others (polarization, fluorescence, phosphorescence, etc.)

- Lighting model: the math that computes reflected light
  - Physical model
    - Computes actual energy transmitted, very expensive
  - Non-physical model (OpenGL)
    - “Close enough”, “looks good”, but fast
**Local Lighting Model**

- Point light source
- Sum of three terms
  - Diffuse light
    - Diffusive reflection
  - Specular light
    - Highlights
  - Ambient light
    - Global, environment light
Local Lighting Model

\[ \text{Diffuse} + \text{Specular} + \text{Ambient} \]
Parameters

- **Geometry**
  - Surface normal (unit vec): \( N = \{ N_x, N_y, N_z \} \)
  - Direction to light source (unit vec): \( L = \{ L_x, L_y, L_z \} \)
  - Distance to light source: \( d_L \)
  - Direction to camera: \( V = \{ V_x, V_y, V_z \} \)
Parameters

• **Geometry**
  - Surface normal (unit vec): \[ N = \{ N_x, N_y, N_z \} \]
  - Direction to light source (unit vec): \[ L = \{ L_x, L_y, L_z \} \]
  - Distance to light source: \[ d_L \]
  - Direction to camera: \[ V = \{ V_x, V_y, V_z \} \]

• **Light properties**
  - Diffuse/Specular Light
    \[ I_L = \{ I^r_L, I^g_L, I^b_L \} \]
  - Ambient light: \[ I_A = \{ I^r_A, I^g_A, I^b_A \} \]
  - Attenuation coefficients: \[ c_0, c_1, c_2 \]
Parameters

- **Geometry**
  - Surface normal (unit vec): \( \mathbf{N} = \{N_x, N_y, N_z\} \)
  - Direction to light source (unit vec): \( \mathbf{L} = \{L_x, L_y, L_z\} \)
  - Distance to light source: \( d_L \)
  - Direction to camera: \( \mathbf{V} = \{V_x, V_y, V_z\} \)

- **Light properties**
  - Diffuse/Specular Light
    \( \mathbf{I}_L = \{I^r_L, I^g_L, I^b_L\} \)
  - Ambient light:
    \( \mathbf{I}_A = \{I^r_A, I^g_A, I^b_A\} \)
  - Attenuation coefficients: \( c_0, c_1, c_2 \)

- **Surface material**
  - Diffuse coefficients:
    \( \mathbf{k}_d = \{k^r_d, k^g_d, k^b_d\} \)
  - Specular coefficients:
    \( \mathbf{k}_s = \{k^r_s, k^g_s, k^b_s\} \) and exponent: \( n \)
  - Ambient coefficients:
    \( \mathbf{k}_a = \{k^r_a, k^g_a, k^b_a\} \)
Diffuse Reflection

- Simulates reflection on matte surfaces
  - Independent of view direction
- Lambert’s Cosine Law

\[ I_{\text{diff}} = I_L \ k_d \ \cos[\theta] = I_L \ k_d \ (N \cdot L) \]
\[ (\theta < \pi / 2, \ \text{or} \ N \cdot L \geq 0) \]
Diffuse Reflection

- Simulates reflection on matte surfaces
  - Independent of view direction
- Lambert’s Cosine Law

\[
I_{\text{diff}} = I_L k_d \cos(\theta) = I_L k_d (N \cdot L)
\]

\((\theta < \pi/2, \text{ or } N \cdot L \geq 0)\)

- Compute for each color component :

\[
I_{\text{diff}}^R = I_L^R k_d^R (N \cdot L)
\]
\[
I_{\text{diff}}^G = I_L^G k_d^G (N \cdot L)
\]
\[
I_{\text{diff}}^B = I_L^B k_d^B (N \cdot L)
\]
Specular Reflection

- Simulates highlight on shiny surfaces
  - Dependent on the viewing direction
- Phong’s approximation

\[
I_{\text{spec}} = I_L k_s \cos[\alpha]^n = I_L k_s (R \cdot V)^n
\]
\[
(N \cdot L \geq 0)
\]
- \(R\): reflected light direction
- \(n\): specular exponent
Specular Reflection

- Simulates highlight on shiny surfaces
  - Dependent on the viewing direction
- Phong’s approximation

\[
I_{\text{spec}} = I_L k_s \cos[\alpha]^n = I_L k_s (R \cdot V)^n \quad (N \cdot L \geq 0)
\]

- \(R\): reflected light direction
- \(n\): specular exponent

Larger \(n\), faster drop-off
Specular Reflection

- Phong’s approximation

\[ I_{\text{spec}} = I_L k_s \cos[\alpha]^n = I_L k_s (R \cdot V)^n \]
\[ (N \cdot L \geq 0) \]

- Often, \( k_s \) is independent of object
Ambient Reflection

• Simulates global illumination
  – Lights bounced off other objects
• Constant light (a simple hack)

\[ I_{\text{amb}} = I_A k_a \]
Light Attenuation

- Simulates decrease of light energy over distance
  
  \[ I_L \leftarrow f_{\text{att}} I_L \]
  
  - Does not affect ambient light

- Inverse square law of energy fall-off

  \[ f_{\text{att}} = \frac{1}{d_L^2} \]

- In practice

  \[ f_{\text{att}} = \frac{1}{c_0 + c_1 d_L + c_2 d_L^2} \]
Putting Together

- Local (OpenGL) lighting model

\[ I = I_{amb} + I_{diff} + I_{spec} \]
\[ = I_A k_a + I_L f_{att} (k_d (N \cdot L) + k_s (R \cdot V)^n) \]

- Compute for each color component
Example: Varying Parameter
Example: Varying Parameter
Example: Varying Parameter

- **Ambient Strength:**
  - AmbientStrength = 0
  - AmbientStrength = 0.15
  - AmbientStrength = 0.25
  - AmbientStrength = 0.75

- **Specular Strength:**
  - SpecularStrength = 0

- **Diffuse Strength:**
  - DiffuseStrength = 0.5
  - DiffuseStrength = 1.0
  - DiffuseStrength = 2.0

- **Specular Exponent:**
  - Specular Exponent = 1
  - Specular Exponent = 10
  - Specular Exponent = 100
Example: Attenuation

No attenuation:

Linear attenuation:

Quadratic attenuation: