Shading

- Compute interaction of light with surfaces
- Requires simulation of physics
- “Global illumination”
  - Multiple bounces of light
  - Computationally expensive, minutes per image
  - Movies, architectural design, etc.

Global illumination

- Rendering algorithms, Winter 2010!

Interactive applications

- No physics based simulation
- Simplified models
- Reproduce perceptually most important effects
- Local illumination
  - Only one bounce of light between light source and viewer

One bounce of light

Rendering pipeline

- Different options for shading in rendering pipeline
- Determine colors of vertices
  - Per vertex shading
- Shading while drawing triangles
  - Per pixel shading
Today

Shading
• Introduction
• Local shading models
• Light sources
• Shading strategies

Local illumination
• What is giving a material its color?
• How is light reflected by a
  - Mirror
  - White sheet of paper
  - Blue sheet of paper
  - Glossy metal

Radiometry
• Physical units to measure light energy
• Based on the ray optics model
• Light modeled as rays
  - Ray is idealized narrow beam of light
• No wave effects, like interference or diffraction

Radiance
• “Energy carried along a narrow beam of light”
• Limit of energy passing through a small area in a small bundle of directions, divided by area and solid angle spanned by bundle of directions
  - “Ray density”
• Spectral radiance: Energy at each wavelength
• Units
  \[ W \cdot sr^{-1} \cdot m^{-2} \]

Irradiance
• Energy per area: “energy going through a small area, divided by size of area“
• “Radiance summed up over all directions“
• Units
  \[ W \cdot m^{-2} \]

Local illumination
• Goal: model reflection of light at surfaces
• Bidirectional reflectance distribution function (BRDF)
  - Given light direction, viewing direction, obtain fraction of light reflected towards the viewer
  - For any pair of light/viewing directions!
  - For different wavelengths (or R, G, B) separately
• BRDF completely describes appearance of material
  “For each pair of light/view direction, BRDF gives fraction of reflected light”
BRDFs

- Given incident and outgoing directions
- BRDF is fraction of incident irradiance arriving from small beam of directions over radiance reflected in outgoing direction
- Units \( \frac{W \cdot m^{-2}}{W \cdot sr^{-1} \cdot m^{-2}} \) = \( \frac{1}{sr} \)

---

BRDFs

- How to define and store BRDFs that represent physical materials?
  - Physical measurements
    - Gonioreflectometer: robot with light source and camera
    - Measures reflection for each light/camera direction
    - Store measurements in table
  - Too much data, too slow for interactive applications

---

BRDFs

- Analytical models
  - Try to describe physical properties of materials using mathematical expressions
  - Many models proposed in graphics
    - [Bidirectional reflectance distribution function](http://en.wikipedia.org/wiki/Bidirectional_reflectance_distribution_function)
  - Most of them too complicated for interactive rendering

---

Simplified model

- BRDF is sum of diffuse, specular, and ambient components
- Each is simple analytical function
- Covers a large class of real surfaces
- Model is not physically accurate!

---

Diffuse reflection

- Ideal diffuse material reflects light equally in all directions
  - View-independent
    - Surface looks the same independent of viewing direction
  - Matte, not shiny materials
    - Paper
    - Unfinished wood
    - Unpolished stone

---

Diffuse sphere

- Ideal diffuse material reflects light equally in all directions
  - View-independent
    - Surface looks the same independent of viewing direction
  - Matte, not shiny materials
    - Paper
    - Unfinished wood
    - Unpolished stone
**Diffuse materials**

Beam of parallel rays shining on a surface
- Area covered by beam varies with the angle between the beam and the normal
- The larger the area, the less incident light per area
- Incident light per unit area is proportional to the cosine of the angle between the normal and the light rays
- Object darkens as normal turns away from light
- Diffuse surfaces are also called Lambertian surfaces

**Lambert’s cosine law**

- Lambert’s cosine law
  - Area covered by beam varies with the angle between the beam and the normal
  - The larger the area, the less incident light per area
  - Incident light per unit area is proportional to the cosine of the angle between the normal and the light rays
  - Object darkens as normal turns away from light
  - Diffuse surfaces are also called Lambertian surfaces

**Diffuse reflection**

- Given
  - Unit surface normal $\mathbf{n}$
  - Unit light direction $\mathbf{L}$
  - Material diffuse reflectance (material color) $k_d$
  - Light color (intensity) $c_l$
- Diffuse color
  \[ c_d = c_l k_d (\mathbf{n} \cdot \mathbf{L}) \]
  - Cosine between normal and light

**Notes**

- Parameters $k_d, c_l$ are r,g,b vectors
- Compute r,g,b values of diffuse color $c_d$ separately
- Parameters in this model have no precise physical meaning
  - $c_l$: strength, color of light source
  - $k_d$: fraction of reflected light, material color

**OpenGL/jogl**

- Lights (glLight*)
  - Values for light: $0,0,0 \leq c_l \leq 1,1,1$
  - $(0,0,0)$ is black, $(1,1,1)$ is white
- OpenGL
  - Values for diffuse reflection: $0,0,0 \leq k_d \leq 1,1,1$
  - Fraction of reflected light
- Consult OpenGL book
  - Online: [http://fly.cc.fer.hr/~unreal/theredbook/](http://fly.cc.fer.hr/~unreal/theredbook/)
**Simplified model**
- BRDF is sum of diffuse, specular, and ambient components
- Each is simple analytical function
- Covers a large class of real surfaces
- Model is not physically accurate!

**Specular reflection**
- Shiny surfaces
  - Polished metal
  - Glossy car finish
  - Plastics
- Specular highlight
  - Blurred reflection of the light source
  - Position of highlight depends on viewing direction

**Glossy materials**
- Ideal specular reflection is mirror reflection
  - Perfectly smooth surface
  - Incoming light ray is bounced in single direction
  - Angle of incidence equals angle of reflection

**Law of reflection**
- Angle of incidence equals angle of reflection
  \[ R + L = 2 \cos \theta \hat{n} = 2(|L| \hat{n})\hat{n} \]
  \[ R = 2(|L| \hat{n})\hat{n} - L \]

**Glossy materials**
- Many materials not quite perfect mirrors
- Glossy materials have blurry reflection of light source

Glossy teapot with highlights from many light sources
**Physical model**

- Assume surface composed of small mirrors with random orientation (microfacets)
- Smooth surfaces
  - Microfacet normals close to surface normal
  - Sharp highlights
- Rough surfaces
  - Microfacet normals vary strongly
  - Leads to blurry highlight

<table>
<thead>
<tr>
<th>Polished</th>
<th>Smooth</th>
<th>Rough</th>
<th>Very rough</th>
</tr>
</thead>
</table>

**Phong model**

- Simple “implementation” of the physical model
- Specular reflectance coefficient $k_s$
- Phong exponent $p$
  - Higher $p$, smaller (sharper) highlight

$$c = k_s L_0 \cdot e^p$$

**Blinn model (Jim Blinn, 1977)**

- Alternative to Phong model
- Define unit halfway vector $h = \frac{L + e}{||L + e||}$
- Halfway vector represents normal of microfacet that would lead to mirror reflection to the eye

**Blinn model**

- The larger the angle between microfacet orientation and normal, the less likely
- Use cosine of angle between them
- Shininess parameter $s$
- Very similar to Phong

$$c = k_s L_0 \cdot h^s$$
**Simplified model**
- BRDF is sum of diffuse, specular, and ambient components
- Each is simple analytical function
- Covers a large class of real surfaces
- Model is not physically accurate!

**Ambient light**
- In real world, light is bounced all around scene
- Could use global illumination techniques to simulate
- Simple approximation
  - Add constant ambient light at each point $k_d c_a$
  - Ambient light $c_a$
  - Ambient reflection coefficient $k_d$
- Areas with no direct illumination are not completely dark

**Complete model**
- Blinn model with several light sources $i$
  \[ c = \sum_i c_i (k_d (L_i \cdot n) + k_s (h_i \cdot n)^n) + k_a c_a \]

**Notes**
- All colors, reflection coefficients have separate values for R,G,B
- Usually, ambient = diffuse coefficient
- For metals, specular = diffuse coefficient
  - Highlight is color of material
- For plastics, specular coefficient = $(x,x,x)$
  - Highlight is color of light

**Today**
**Shading**
- Introduction
- Local shading models
- Light sources
- Shading strategies

**Light sources**
- Light sources can have complex properties
  - Geometric area over which light is produced
  - Anisotropy in direction
  - Variation in color
  - Reflective surfaces act as light sources
- Interactive rendering is based on simple, standard light sources
### Light sources

- At each point on surfaces need to know
  - Direction of incoming light (the $L$ vector)
  - Strength of incoming light (the $c_i$ values)
- Standard light sources in OpenGL
  - Directional: from a specific direction
  - Point light source: from a specific point
  - Spotlight: from a specific point with intensity that depends on the direction

### Directional light

- Light from a distant source
  - Light rays are parallel
  - Direction and strength constant everywhere in 3D scene
  - As if the source were infinitely far away
  - Good approximation to sunlight
- Specified by a unit length direction vector, and a color

### Point lights

- Simple model for light bulbs
- Point that radiates light in all directions equally
  - Light vector varies across the surface
  - Intensity drops off proportionally to the inverse square of the distance from the light
  - Intuition for inverse square falloff?

### Point lights

- Incident light direction
  \[ \mathbf{L} = \frac{\mathbf{p} - \mathbf{v}}{||\mathbf{p} - \mathbf{v}||} \]
- Strength
  \[ c_I = \frac{c_{src}}{||\mathbf{p} - \mathbf{v}||^2} \]

### Attenuation

- Sometimes, it is desirable to modify the inverse square falloff behavior of point lights
  - Common (OpenGL) model for distance attenuation
    \[ c_I = \frac{c_{src}}{k_c + k_p ||\mathbf{p} - \mathbf{v}|| + k_l ||\mathbf{p} - \mathbf{v}||^2} \]
  - Not physically accurate

### Spotlights

- Like point source, but intensity depends on direction
- Specified by a unit length direction vector, and a color

#### Parameters
- Position, the location of the source
- Spot direction, the center axis of the light
- Falloff parameters
  - how broad the beam is (cone angle)
  - how light tapers off at edges of the beam (cosine exponent)
Spotlights

\[ \mathbf{L} = \frac{\mathbf{p} - \mathbf{v}}{\|\mathbf{p} - \mathbf{v}\|} \]
\[ q_i = \begin{cases} 0 & \text{if } -\mathbf{L} \cdot \mathbf{d} \leq \cos(\theta_{\text{max}}) \\ \ell_{\text{arc}} (-\mathbf{L} \cdot \mathbf{d})' & \text{otherwise} \end{cases} \]

Photograph of spotlight

Spotlights in OpenGL

Today

Shading
- Introduction
- Local shading models
- Light sources
- Shading strategies

Per-triangle, -vertex, -pixel shading
- May compute shading operations
  - Once per triangle
  - Once per vertex
  - Once per pixel

Per-triangle shading
- Known as flat shading
- Evaluate shading once per triangle using per-triangle normal
- Advantages
  - Fast
- Disadvantages
  - Faceted appearance

Per-vertex shading
- Known as Gouraud shading (Henri Gouraud 1971)
- Per-vertex normals
- Interpolate vertex colors across triangles
- OpenGL default
- Advantages
  - Fast
  - Smoother than flat shading
- Disadvantages
  - Problems with small highlights
Per-pixel shading

- Also known as Phong interpolation (not to be confused with Phong illumination model)
  - Rasterizer interpolates normals across triangles
  - Illumination model evaluated at each pixel
  - Implemented using programmable shaders (next week)
- Advantages
  - Higher quality than Gouraud shading
- Disadvantages
  - Much slower

Gouraud vs. per-pixel shading

- Gouraud has problems with highlights
- Could use more triangles...

What about shadows?

- Standard shading assumes light sources are always visible
  - Does not determine if light is blocked
  - Does not produce shadows
- Shadows require additional work
- Later in the course

What about textures?

- How to combine „colors“ stored in textures and lighting computations?
  - Interpret textures as shading coefficients
  - Usually, texture used as ambient and diffuse reflectance coefficient \( k_d, k_a \)
- Textures provide spatially varying BRDFs
  - Each point on surface has different BRDF parameters, different appearance

Summary

- Local illumination (single bounce) is computed using BRDF
- BRDF captures appearance of a material
  - Amount of reflected light for each pair of light/viewing directions
- Simplified model for BRDF consists of diffuse + Phong/Blinn + ambient
  - Lambert’s law for diffuse surfaces
  - Microfacet model for specular part
  - Ambient to approximate multiple bounces
- Light source models
  - Directional
  - Point, spot, inverse square fall-off
- Different shading strategies
  - Per triangle, Gouraud, per pixel

Next time

- Programmable shaders