Announcements

- Assignment 0: due this Friday
- Homework 1: due this Thursday
- Assignment 1: due February 14th
**Today**

- Local Illumination & Shading
  - The BRDF
  - Simple diffuse and specular approximations
  - Shading interpolation: flat, Gouraud, Phong
  - Some miscellaneous tricks

---

**Local Shading**

- Local: consider in isolation
  - 1 light
  - 1 surface
  - The viewer
- Recall: lighting is linear
  - Almost always...

Counter example: photochromatic materials
Local Shading

- Examples of non-local phenomena
  - Shadows
  - Reflections
  - Refraction
  - Indirect lighting

The BRDF

- The Bi-directional Reflectance Distribution Function
- Given
  - Surface material
  - Incoming light direction
  - Direction of viewer
  - Orientation of surface
- Return:
  - Fraction of light that reaches the viewer
  - We'll worry about physical units later...
The BRDF

\[ \rho(v, l, n) \]

- Spatial variation capture by "the material"
- Frequency dependent
  - Typically use separate RGB functions
  - Does not work perfectly
  - Better:
    \[ \rho = \rho(\theta_v, \theta_L, \lambda_{\text{in}}, \lambda_{\text{out}}) \]

Obtaining BRDFs

- Measure from real materials

Images from Marc Levoy
## Obtaining BRDFs

- Measure from real materials
- Computer simulation
  - Simple model + complex geometry
- Derive model by analysis
- Make something up

## Beyond BRDFs

- The BRDF model does not capture everything
  - e.g. Subsurface scattering (BSSRDF)

Images from Jensen et al. SIGGRAPH 2001
## Beyond BRDFs

- The BRDF model does not capture everything
  - e.g. Inter-frequency interactions

\[
\rho = \rho(\theta_V, \theta_L, \lambda_{\text{in}}, \lambda_{\text{out}}) \text{ This version would work...}
\]

## A Simple Model

- Approximate BRDF as sum of
  - A diffuse component
  - A specular component
  - A "ambient" term
Diffuse Component

- Lambert’s Law
  - Intensity of reflected light proportional to cosine of angle between surface and incoming light direction
  - Applies to “diffuse,” “Lambertian,” or “matte” surfaces
  - Independent of viewing angle
  - Use as a component of non-Lambertian surfaces

\[ k_d I (\hat{\mathbf{l}} \cdot \hat{\mathbf{n}}) \]

\[ \max(k_d I (\hat{\mathbf{l}} \cdot \hat{\mathbf{n}}), 0) \]
Diffuse Component

- Plot light leaving in a given direction:

- Plot light leaving from each point on surface

Specular Component

- Specular component is a mirror-like reflection
- Phong Illumination Model
  - A reasonable approximation for some surfaces
  - Fairly cheap to compute
  - Depends on view direction
Specular Component

\[ k_s I (\hat{r} \cdot \hat{v})^p \]

\[ k_s I \max(\hat{r} \cdot \hat{v}, 0)^p \]

\[ \hat{r} = -\hat{l} + 2(\hat{l} \cdot \hat{n})\hat{n} \]
Specular Component

• “Half-angle” approximation for specular

\[ \hat{\mathbf{h}} = \frac{\hat{\mathbf{i}} + \hat{\mathbf{v}}}{||\hat{\mathbf{i}} + \hat{\mathbf{v}}||} \]

different specular term \[ k_s I (\hat{\mathbf{h}} \cdot \hat{\mathbf{n}})^p \]

*Don’t use half-angle approximation in your assignments!

Specular Component

• Plot light leaving in a given direction:

• Plot light leaving from each point on surface

~
### Specular Component

- Specular exponent sometimes called “roughness”

<table>
<thead>
<tr>
<th>n=1</th>
<th>n=2</th>
<th>n=4</th>
<th>n=8</th>
<th>n=16</th>
<th>n=32</th>
<th>n=64</th>
<th>n=128</th>
<th>n=256</th>
</tr>
</thead>
</table>

### Ambient Term

- Really it’s a cheap hack
- Accounts for “ambient, omnidirectional light”
- Without it everything looks like it’s in space
Summing the Parts

\[ R = k_d I + k_a I \max(\hat{I} \cdot \hat{n}, 0) + k_s I \max(\hat{r} \cdot \hat{v}, 0)^p \]

- Recall that the \( k \) are by wavelength
  - RGB in practice
  - Sum over all lights

Anisotropy
<table>
<thead>
<tr>
<th>Metal -vs- Plastic</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Metal Ball" /> <img src="image2.png" alt="Plastic Ball" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metal -vs- Plastic</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image3.png" alt="Metal Sphere" /> <img src="image4.png" alt="Plastic Sphere" /> <img src="image5.png" alt="Metal Sphere" /> <img src="image6.png" alt="Plastic Sphere" /></td>
</tr>
</tbody>
</table>
Other Color Effects

Images from Godal et al. | P80
Measured BRDFs

BRDFs for automotive paint

BRDFs for aerosol spray paint
Measured BRDFs

BRDFs for house paint

BRDFs for lucite sheet
Details Beget Realism

- The “computer generated” look is often due to a lack of fine/subtle details... a lack of richness.

Direction - vs - Point Lights

- For a point light, the light direction changes over the surface
- For “distant” light, the direction is constant
- Similar for orthographic/perspective viewer
**Falloff**

- Physically correct: $1/r^2$ light intensity falloff
  - Tends to look bad (why?)
  - Not used in practice
  - Sometimes compromise of $1/r$ used

**Spot and Other Lights**

- Other calculations for useful effects
  - Spot light
  - Only light certain objects
  - Negative lights
  - etc.
<table>
<thead>
<tr>
<th>Ugly....</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ugly....</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image2.png" alt="Image" /></td>
</tr>
</tbody>
</table>

03-Shading.key - January 28, 2014
### Surface Normals

- The normal vector at a point on a surface is perpendicular to all surface tangent vectors.

- For triangles normal given by right-handed cross product.

### Flat Shading

- Use constant normal for each triangle (polygon)
  - Polygon objects don't look smooth
  - Faceted appearance very noticeable, especially at specular highlights
  - Recall mach bands...
### Smooth Shading

- Compute “average” normal at vertices
- Interpolate across polygons
- Use threshold for “sharp” edges
  - Vertex may have different normals for each face

![Smooth Shading Diagram]
### Gouraud Shading

- Compute shading at each vertex
  - Interpolate colors from vertices
  - Pros: fast and easy, looks smooth
  - Cons: terrible for specular reflections

![Flat vs Gouraud](image)

Note: Gouraud was hardware rendered...

### Phong Shading

- Compute shading at each pixel
  - Interpolate normals from vertices
  - Pros: looks smooth, better speculars
  - Cons: expensive

![Gouraud vs Phong](image)

Note: Gouraud was hardware rendered...