Light Reflection and Illumination

CS 4620 Lecture 35
Shading

• Variation in observed color across an object
  – strongly affected by lighting
  – present even for homogeneous material
• caused by how a material reflects light
  – depends on
    • geometry
    • lighting
    • material
  – therefore gives cues to all 3
Shading for Computer Graphics

• Need to compute an image
  – of particular geometry
  – under particular illumination
  – from a particular viewpoint

• Basic question: how much light reflects from an object toward the viewer?
Diffuse + Phong shading
Mirror reflection

• Consider perfectly shiny surface
  – there isn’t a highlight
  – instead there’s a reflection of other objects
• Can render this using recursive ray tracing
  – to find out mirror reflection color, ask what color is seen
    from surface point in reflection direction
  – already computing reflection direction for Phong…
• “Glazed” material has mirror reflection and diffuse

\[ L = L_a + L_d + L_m \]
  – where \( L_m \) is evaluated by tracing a new ray
Mirror reflection

- Intensity depends on view direction
  - reflects incident light from mirror direction

\[ r = v + 2((n \cdot v)n - v) \]
\[ = 2(n \cdot v)n - v \]
Diffuse + mirror reflection (glazed)

(glazed material on floor)
Fancier shading

- Diffuse + Phong has long been the heuristic baseline for surface shading
- Newer/better methods are more based on physics
  - when writing a shader, think like a bug standing on the surface
  - bug sees an *incident distribution* of light that is arriving at the surface
  - physics question: what is the *outgoing distribution* of light?
Simple materials

metal

dielectric
Adding microgeometry
Classic reflection behavior

ideal specular (mirror)

glossy specular

Lambertian
Specular reflection

- Smooth surfaces of pure materials have ideal specular reflection (said this before)
  - Metals (conductors) and dielectrics (insulators) behave differently
- Reflectance (fraction of light reflected) and angle of reflection/refraction depend on angle
Refraction at boundary of media
Snell’s Law

- Tells us where the refracted ray goes
- Computation
  - ratio of sines is ratio of in-plane components
  - project to surface; scale by eta ratio; recompute normal-direction component
  - total internal reflection

\[ \eta_1 \sin \theta_1 = \eta_2 \sin \theta_2 \]
Computing Ray Directions

\[ s_2 = \left( \frac{\eta_1}{\eta_2} \right) s_1 \]
Total Internal Reflection

• Occurs when $s_2 > 1$
• All light is reflected; no refraction
Ray tracing dielectrics

• Like a simple mirror surface, use recursive ray tracing
• But we need two rays
  – One reflects off the surface (same as mirror ray)
  – The other crosses the surface (computed using Snell’s law)
    • Doesn’t always exist (total internal reflection)
• Splitting into two rays, recursively, creates a ray tree
  – Very many rays are traced per viewing ray
  – Ways to prune the tree
    • Limit on ray depth
    • Limit on ray attenuation
Specular reflection from metal

- Reflectance does depend on angle
  - but not much
  - safely ignored in basic rendering
Specular reflection from glass/water

- Dependence on angle is dramatic!
  - about 4% at normal incidence
  - always 100% at grazing
  - remaining light is transmitted

- This is important for proper appearance
Fresnel reflection

constant reflectance

Fresnel reflectance
Fresnel’s formulas

- They predict how much light reflects from a smooth interface between two materials
  - usually one material is empty space

\[
F_p = \frac{\eta_2 \cos \theta_1 - \eta_1 \cos \theta_2}{\eta_2 \cos \theta_1 + \eta_1 \cos \theta_2}
\]

\[
F_s = \frac{\eta_1 \cos \theta_1 - \eta_2 \cos \theta_2}{\eta_1 \cos \theta_1 + \eta_2 \cos \theta_2}
\]

\[
R = \frac{1}{2} \left( F_p^2 + F_s^2 \right)
\]

- \( R \) is the fraction that is reflected
- \( (1 - R) \) is the fraction that is transmitted
Schlick’s approximation

- For graphics, a quick hack to get close with less computation:

\[ \tilde{R} = R_0 + (1 - R_0)(1 - \cos \theta)^5 \]

- \( R_0 \) is easy to compute:

\[
F_p = \frac{\eta_2 - \eta_1}{\eta_2 + \eta_1} \\
F_s = \frac{\eta_1 - \eta_2}{\eta_1 + \eta_2} \\
R_0 = \left( \frac{\eta_2 - \eta_1}{\eta_2 + \eta_1} \right)^2
\]
Fresnel reflection
<table>
<thead>
<tr>
<th></th>
<th>diffuse</th>
<th>glossy</th>
<th>mirror</th>
</tr>
</thead>
<tbody>
<tr>
<td>indirect</td>
<td>soft indirect illumination</td>
<td>blurry reflections of other objects</td>
<td>reflected images of other objects</td>
</tr>
<tr>
<td>environment</td>
<td>soft shadows</td>
<td>blurry reflection of environment</td>
<td>reflected image of environment</td>
</tr>
<tr>
<td>area</td>
<td>soft shadows</td>
<td>shaped specular highlight</td>
<td>reflected image of source</td>
</tr>
<tr>
<td>point/directional</td>
<td>hard shadows</td>
<td>simple specular highlight</td>
<td>point reflections</td>
</tr>
</tbody>
</table>

= easy to include in “classic” ray tracer
BRDF

\[ f_r(\theta_i, \phi_i, \theta_e, \phi_e) \ ; \ f_r \geq 0 \]
Reciprocity

- Interchanging arguments
- Physical requirement
Energy Conservation

- Reflected power < incident power
- Physical requirement