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) depends on angle

Snell’s Law

- Tells us where the refracted ray goes

\[ \eta_1 \sin \theta_1 = \eta_2 \sin \theta_2 \]
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

- Black glazed sphere
  - Reflection from glass surface
  - Transmitted ray is discarded

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:

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

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

\[
R_0 = \left( \frac{\eta_2 - \eta_1}{\eta_2 + \eta_1} \right)^2
\]
Basic ray tracing

- Many advanced methods build on the basic ray tracing paradigm
- Basic ray tracer: one sample for everything
  - one ray per pixel
  - one shadow ray for every point light
  - one reflection ray, possibly one refraction ray, per intersection

Discontinuities in basic RT

- Perfectly sharp object silhouettes in image
  - leads to aliasing problems (stair steps)
- Perfectly sharp shadow edges
  - everything looks like it’s in direct sun
- Perfectly clear mirror reflections
  - reflective surfaces are all highly polished
- Perfect focus at all distances
  - camera always has an infinitely tiny aperture
- Perfectly frozen instant in time (in animation)
  - motion is frozen as if by strobe light
Soft shadows

Cause of soft shadows

point lights cast hard shadows

area lights cast soft shadows

Glossy reflection
**Cause of glossy reflection**

- smooth surfaces produce sharp reflections

**Cause of focusing effects**

- what lenses do (roughly)

- rough surfaces produce soft (glossy) reflections
**Cause of focusing effects**

- **Point aperture produces always-sharp focus**

**Cause of focusing effects**

- **Finite aperture produces limited depth of field**

**Motion blur**

(source: Cook, Porter, Carpenter 1984)

**Cause of motion blur**

- **Image point sees different object points at different times**
Creating soft shadows

- For area lights: use many shadow rays
  - and each shadow ray gets a different point on the light
- Choosing samples
  - general principle: start with uniform in square

![Diagram of shadow rays]

Creating glossy reflections

- Jitter the reflected rays
  - Not exactly in mirror direction; add a random offset
  - Can work out math to match Phong exactly
  - Can do this by jittering the normal if you want

![Diagram of glossy reflections]

Depth of field

- Make eye rays start at random points on aperture
  - always going toward a point on the focus plane

![Diagram of depth of field]

Motion blur

- Caused by finite shutter times
  - strobing without blur
- Introduce time as a variable throughout the system
  - object are hit by rays according to their position at a given time
- Then generate rays with times distributed over shutter interval

![Diagram of motion blur]
Generating samples

- A complicated question in general
- Basic idea: start with random points in a square
- Monte Carlo methods—CS 667