Shadows
Shadows

for each pixel do
  compute viewing ray
  if ( ray hits an object with t in [0, inf] ) then
    compute n
    evaluate shading model and set pixel to that color
  else
    set pixel color to the background color
for each pixel do
compute viewing ray
if ( ray hits an object with \( t \) in \([0, \infty]\) ) then
    compute \( n \)
    evaluate shading model and set pixel to that color
else
    set pixel color to the background color
for each pixel do
    compute viewing ray
    if ( ray hits an object with t in [0, inf] ) then
        compute n
        // e.g., phong shading
        for each light
            add light’s ambient component
        compute shadow ray
        if ( ! shadow ray hits an object )
            add light’s diffuse and specular components
        else
            set pixel color to the background color
Reflections
for each pixel do
compute viewing ray
if ( ray hits an object with t in [0, inf] ) then
    compute n
    evaluate shading model and set pixel to that color
else
    set pixel color to the background color
for each pixel do
  compute viewing ray
  if ( ray hits an object with t in [0, inf] ) then
    compute n
    evaluate shading model and set pixel to that color
  else
    set pixel color to the background color

Reflections
for each pixel do
  compute viewing ray
  pixel color = \texttt{cast\_ray}(viewing ray)

\texttt{cast\_ray:}
  \textbf{if} ( ray hits an object with t in [0, \text{inf}] ) \textbf{then}
    compute \textit{n}
    return color = \texttt{shade\_surface}
  \textbf{else}
    return color = to the background color

\texttt{shade\_surface:}
  color = \ldots
  compute reflected ray
  return color = color + k * \texttt{cast\_ray}(reflected ray)
ray tracer extensions

- refraction
- more complex geometry
  - instancing
  - CSG
- distribution ray tracing (Cook et al., 1984)
  - antialiasing
  - soft shadows
  - depth of field
  - fuzzy reflections
  - motion blur
Transparency and Refraction
Transparency and Refraction

Snell’s Law

\[ n_1 \sin \theta = n_2 \sin \phi \]

Example values of \( n \):
- air: 1.00;
- water: 1.33–1.34;
- window glass: 1.51;
- optical glass: 1.49–1.92;
- diamond: 2.42.
Transparency and Refraction

Snell’s Law

Additional effects
- varying reflectivity
  - Fresnel equations
- attenuation of light intensity
  - Beer’s Law
Object Instancing

instance of circle with 3 transformations applied

ray intersection problem in the two spaces are simple transforms of each other
Constructive Solid Geometry (CSG)

use set operations to combine solid shapes

intersection with composite object
Distribution Ray Tracing
Anti-aliasing

[Shirley and Marschner]
Soft Shadows

[Shirley and Marschner]
Soft Focus

lens

focus plane

Shirley and Marschner
Fuzzy Reflections
Motion Blur

objects move while camera aperture is open
Motion Blur

to simulate, choose random time within open aperture interval for each view ray
Acceleration Structures
Acceleration Structures

[Shirley and Marschner]
Bounding boxes

- Bounding box:
  - Contains multiple shapes.
- Ray:
  - Intersects the bounding box.

Mathematical expressions:
- $t \in [t_{\text{min}}, t_{\text{max}}]$:
  - Indicates the range of $t$ values.
- $t \in [t_{\text{min}}, t_{\text{max}}] \cap [t_{\text{min}}, t_{\text{max}}]$:
  - Intersection of two ranges.
Uniform Spatial Partitioning

[Diagram of a ray intersecting a grid with shaded regions, possibly representing intersections or occlusions]
Bounding Volume Hierarchy

[Shirley and Marschner]