CS 665
Advanced Interactive Rendering

Fall 2004

Kavita Bala

Computer Science

Cornell University

Information

- Instructor: Kavita Bala kb@cs.cornell.edu
- AA: Cindy Robinson cindy@cs.cornell.edu
- Tue and Thu 10:10-11:25
  - Moving location Rhodes 484
  - Instead of Phillips 219

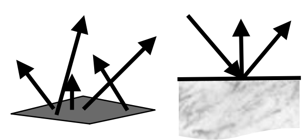
What is this course about

- What does image generation mean?
  - Physics of light
- How to generate images?
  - Global illumination algorithms
- How do we do this efficiently?
  - Interactive rendering, data structures, image-based rendering etc.


RTGI, Program of Computer Graphics

Physics of light

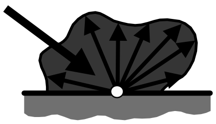
- What is light?
- How does it behave?
  - Does it bend?



- What effects are visible due to different behaviors of light?
  - Polarization, interference, …
- Radiometry

Materials

- How does light interact with materials?

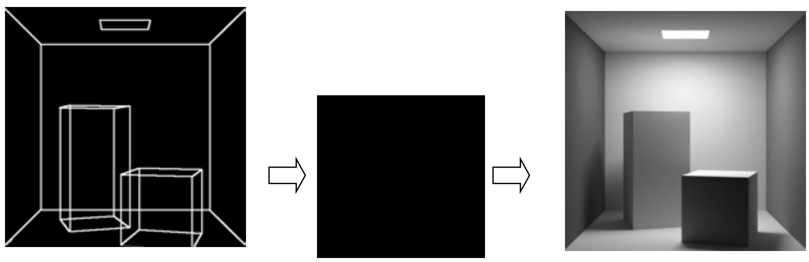


How do we generate images?

- How does light propagate in 3D?
- Rendering Equation: mathematical formulation of global illumination problem

Global Illumination

- GI algorithms solve the rendering equation
- Generate 2D image from 3D scene



Global Illumination

Emission (light sources)

BRDFs (materials)

Geometry (objects)

Global Illumination

- Ray tracing (Whitted)
- Radiosity (Finite Element)
- Monte Carlo rendering

Classic Ray Tracing

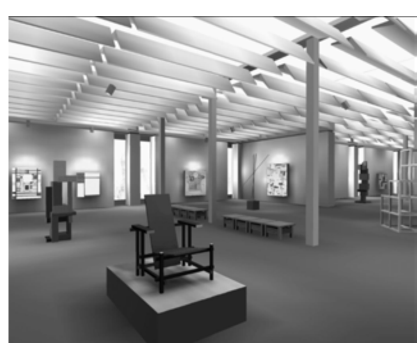
- Introduced in 1980 by Turner Whitted
- First global illumination algorithm
- Many advances through the 80s
- Widely available in commercial and public-domain software
  - Rayshade, Radiance

Ray tracing

Classic Radiosity

- Introduced in 1984
- Diffuse inter-reflections
- Widely available
  - Lightscape

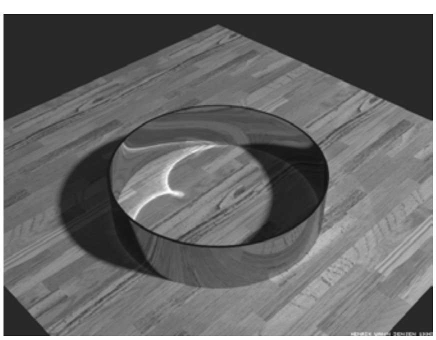
Radiosity Pictures



Advanced Global Illumination

- Classic ray tracing and classic radiosity are basic building blocks
- More realistic materials than just perfect specular / diffuse
- We want ***accurate*** solutions

Global Solutions

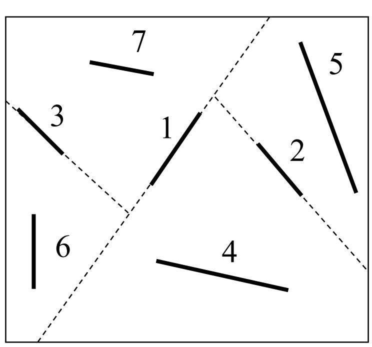


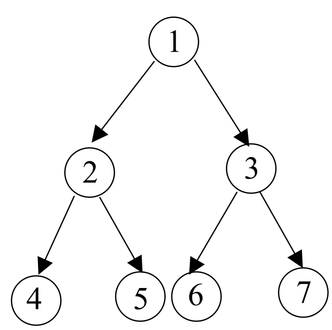
Complexity

- How do we handle complexity?
  - Many objects
  - Many lights
  - Complex BRDFs
  - Global illumination
  - Dynamic scenes

Acceleration Structures

- Octrees, kd-trees, bounding volume hierarchies, BSP trees

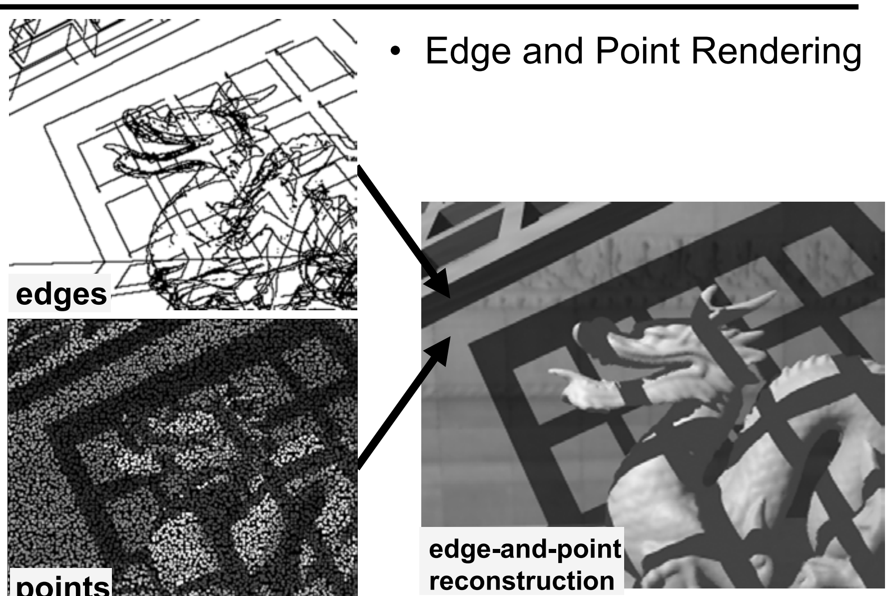




Fast Rendering

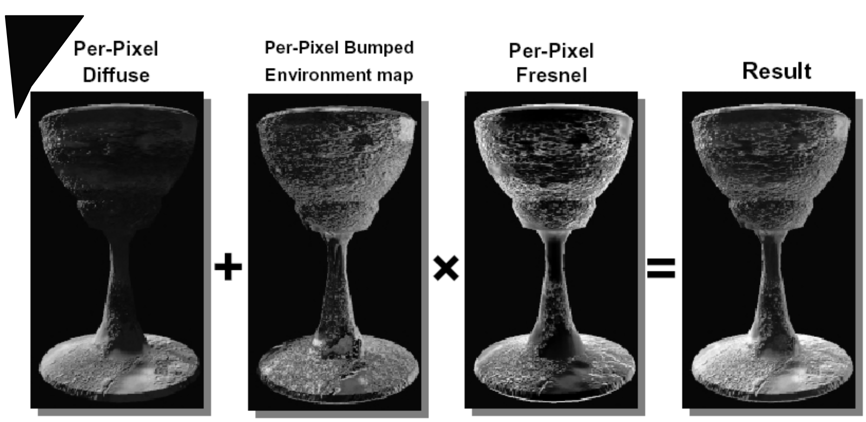
Intelligent Sampling

- Edge and Point Rendering

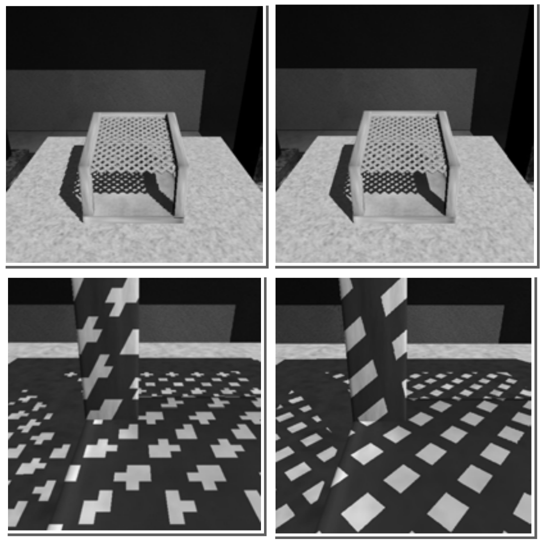


Dynamic Scenes

Hardware Rendering



Shadows



Fernando, Fernandez, Bala, Greenberg [SIG01]

Shadows

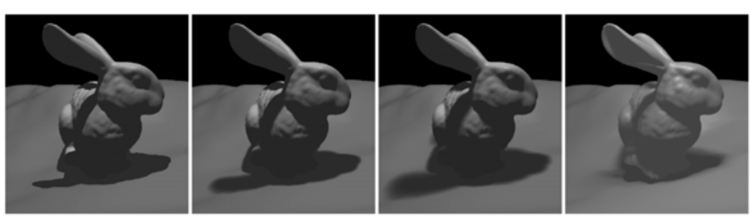


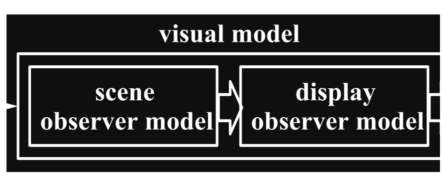
Figure 7: *Comparison of the Stanford Bunny with shadow maps (left), penumbra maps with two different sized lights (center), and a pathtraced shadow using the larger light (right). For this data set, we generate shadows using a 10k polygon model and render the shadows onto the full (~70k polygon) model.*

Figure 8: *Using a standard shadow map results in hard shadows (left), add a penumbra map to get soft shadows (right). Using a 10k polygon dragon model for the shadows and a 50k polygon model to render, we get 14.5 fps at 1024x1024.*

Display the image ...

- GI computes radiance. How to display radiance to user?
- How to transform radiometric units to RGB screen values?
- Model the ***Human Visual System*** **(HVS)**

Realistic image display



tone reproduction operator

- The tone reproduction problem

Modeling visual adaptation

10,000:1 dynamic range

before
linear mapping

after
visual mapping

Perceptually-based rendering

Understanding the human visual system
to *decide what is important to render*

Stuart Little, Dreamworks, 1999

Complexity

- How do we handle complexity?
  - Many objects
  - Many lights
  - Complex BRDFs
  - Global illumination
  - Dynamic scenes

NPR

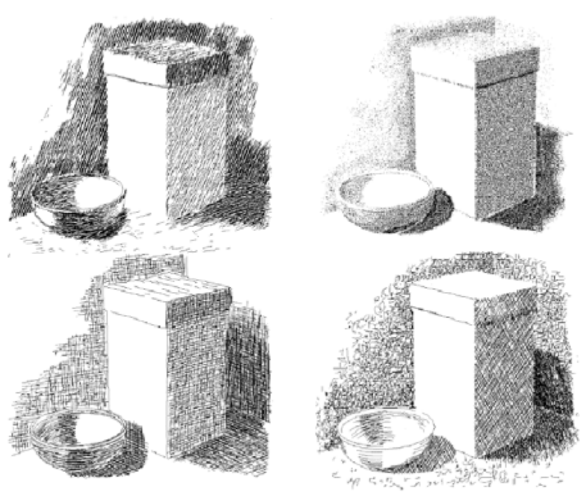


Image-Based Rendering

- Use photographs to capture complex scenes

By the end of the course…

Fundamental understanding of:

- Algorithms for generating images
  - Photorealistic and NPR
- Efficient techniques for high-quality rendering

Pre-requisites

- An introductory graphics course
- Talk to me if you have not taken the Cornell introductory courses

Administration

- 3-4 assignments
  – Written exercises
  – Programming assignments
- Final project
  – Groups?
- Focus on understanding concepts
- 1 mid-term

Academic Integrity

- Can work in groups
- Don’t copy from Web

Books

- Advanced Global Illumination
  Dutre, Bekaert, Bala
  – Rendering
  – Monte Carlo techniques
  – Current areas

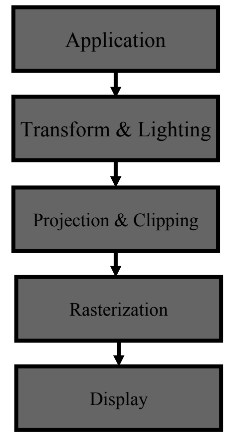
Information

- www.cs.cornell.edu/courses/cs665/2004fa/
  - Tentative schedule
  - Homeworks, lecture notes, will be on-line
  - Check for updates and announcements

Questions?

Traditional Graphics Pipeline

- Modeling Transformation
  - World space transformations
- Lighting
  - Local shading model
- Rasterize pixels
- Interpolate color/depth/etc.
- Z-buffer for hidden surface elimination



Shading Model

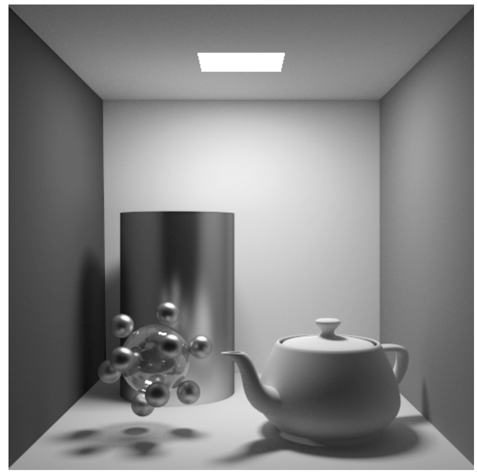
$$I(x,y) = \sum_{i=1}^{Nlights} (k_d (N.L) + k_s (N.H)^n) V_i \quad + \quad I_a$$

diffuse specular ambient

- Illumination at surface equals
  - Ambient +Diffuse +Specular highlights
- With programmable GPUs
  - Can have arbitrary shading model

But what about other cues?

- Lighting: Shadows
- Lighting: Shading
  - Glossy
  - Transparency
- Color bleeding

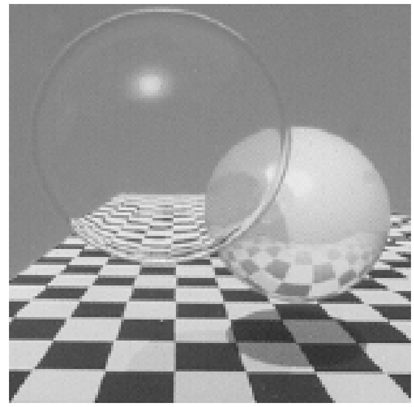


Classic Ray Tracing

- Introduced in 1980 by Turner Whitted
- Existing rendering:
  - Phong shading
  - Local illumination (specular, diffuse)

Insights

- Trace rays from eye into scene
  - Backward ray tracing
- Find visible objects
- Shade visible points
  - Shadows
  - Reflections
  - Refractions

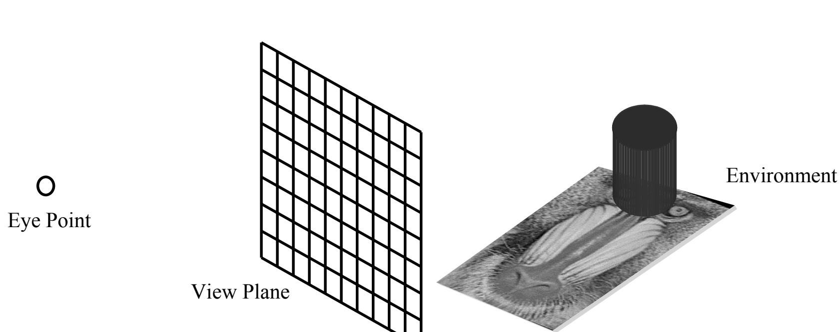


Whitted 1980: First ray traced image

- First global illumination algorithm!

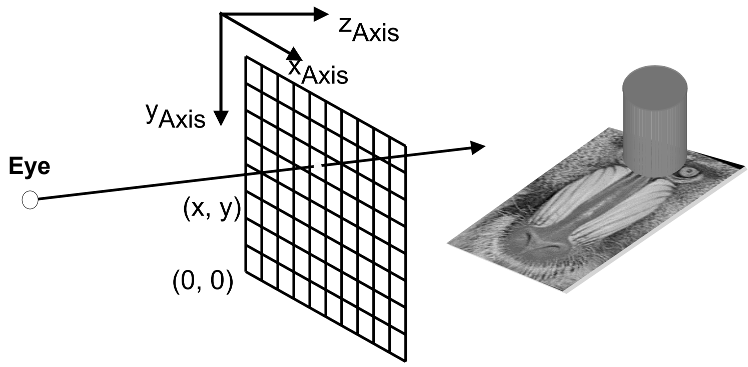
Basic Algorithm - View Setup

- Synthetic camera defined by “eye point” and “view plane” in world coordinates
- View plane divided into pixels corresponding to the image dimensions



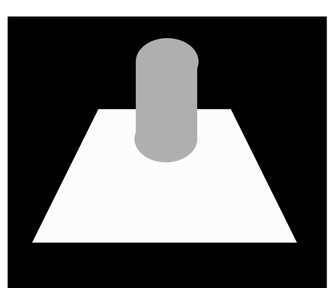
Basic Algorithm - View Rays

- Rays are cast from the eye point through each pixel in the image



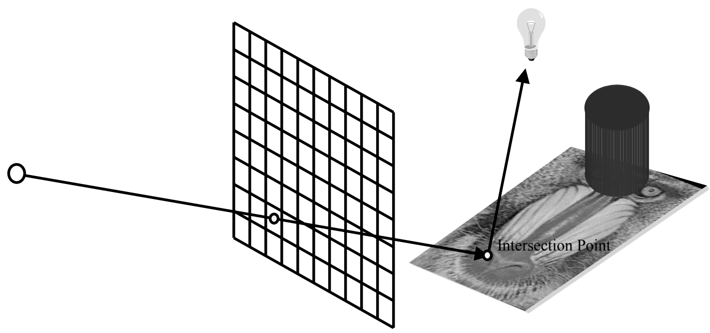
Visibility Determination

- Intersect eye ray with all objects in scene
  - Find closest object
  - Z-buffer was existing algorithm
- No intersection? Show background color



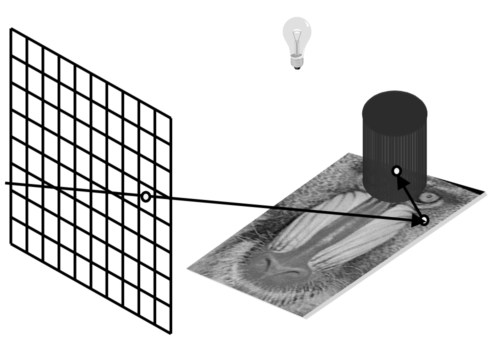
Basic Algorithm - Shadows

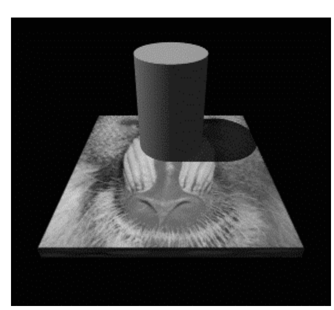
- Cast ray from the surface point to each light source: shadow rays



Basic Algorithm - Shadows, cont.

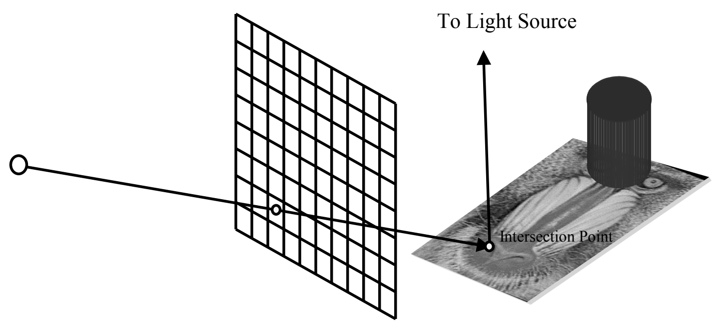
- Shadow ray is blocked = shadow





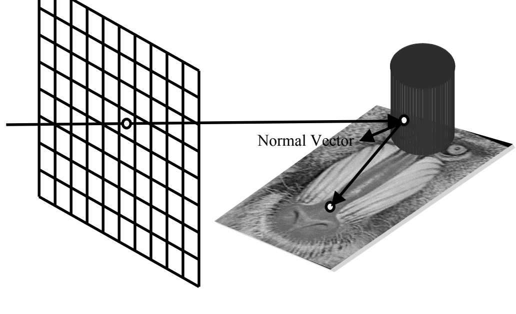
Basic Algorithm – Shadow Rays

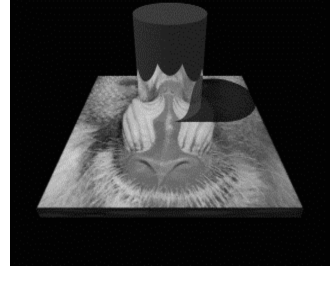
- If shadow ray not blocked, calculate radiance based on shading model



Basic Algorithm - Reflections

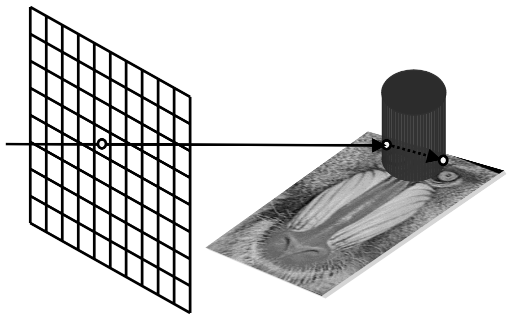
- If object specular, shoot secondary reflected rays

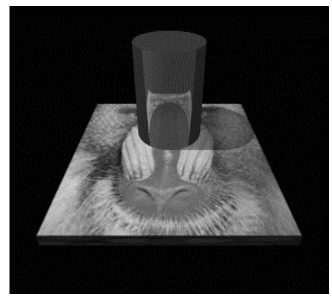




Basic Algorithm - Refractions

- If object transparent, shoot secondary refracted rays





Whitted RT Shading Model

$$I(x,y) = \sum_{i=1}^{Nlights} (k_d (N.L) + k_s (N.H)^n) V_i \quad +$$

diffuse specular

$$I_a \quad + \quad k_s I_r \quad + \quad k_t I_t$$

ambient reflected refracted

- Illumination at surface equals
  - Ambient +Diffuse +Specular highlights +
  - Secondary specular reflections and transmissions

- Equivalent to Blinn-Phong plus contributions from specularly reflected and transmitted rays

High-level algorithm

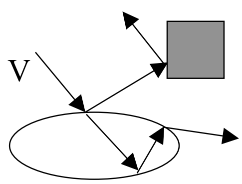
```
For each pixel (x,y) {
    eye ray e = ray through pixel (x,y)
    color of pixel (x,y) = Trace (e, scene)
}
Trace (Ray eyeray, Scene scene) {
    o = intersect (eyeray, scene)
    if (o != null) {
        Shade (o, p, N, scene, e,…)
    } else return background color
}
```

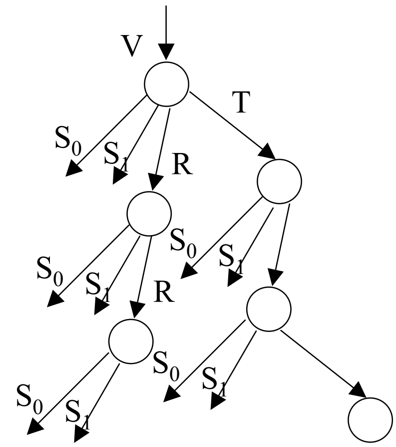

Shade

Shade (Object o, Point pt, Normal N, Scene scene, Ray ray, …) {
  for each light {
    if (!intersect (shadowray, scene))
      color += diffuse+specular // not in shadow
  }
  if (o.specular) color += k_sTrace (reflected ray)
  if (o.transparent) color += k_tTrace (refracted ray)
}

Basic Algorithm - Recursion

- Reflected and/or transmitted recursively spawn more rays
  - Ray tree
- Depth cutoff
- Weight cutoff





Classic Ray Tracing

- Image-based
- Gathering approach
  - from the light sources (direct illumination)
  - from the reflected direction (perfect specular)
  - from the refracted direction (perfect specular)
- All other contributions are ignored!
  - Not a complete solution

Whitted RT Assumptions

- Light Source: point light source
  - Hard shadows
  - Single shadow ray direction
- Material: Blinn-Phong model
  - Diffuse with specular peak
- Light Propagation
  - Occluding objects
  - Specular interreflections only
    - trace rays in mirror reflection direction only

History

- Problems with classic ray tracing:
  - Not realistic: only perfect specular and perfect refraction/reflection between surfaces
  - View-dependent
- Radiosity (1984)
  - Global Illumination in diffuse scenes
  - Discretize scene
- Monte Carlo Ray Tracing (1986)
  - Global Illumination for any environment