


Now Playing:

JOHN COLTRANE GIANT STEPS




Giant Steps  
From *Giant Steps*  
Recorded May 4-5, 1959  
John Coltrane - Tenor Sax  
Tommy Flanagan - Piano  
Paul Chambers - Bass  
Art Taylor - Drums

## Luxo Jr. (Pixar, 1986)



## Real Cameras and Ray Tracing

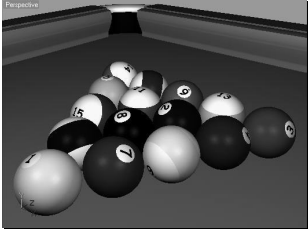


COMP 575  
August 28, 2007

## Last Time

- Overview of the first half of the semester
- Talked about displays
  - Raster vs. Vector
  - Cathode Ray Tube (CRT) displays
  - Liquid Crystal Displays (LCDs)
- Talked about the graphics pipeline
  - How to turn 3D geometry into screen pixels

## Standard Rasterization



- (NOTE: You can do better with fancy shaders, etc.)
- Notice anything?

## Lights, Cameras, Surfaces

- Last time I talked briefly about “lights”, “surfaces”, and “cameras” from the perspective of the graphics pipeline
  - “Surfaces” are the 3D geometry in the scene
  - “Lights” determine the color of the surface
  - The “camera” projects the 3D geometry onto the view plane

## Lights, Cameras, Surfaces

- How are these connected in the real world?
- Light!
- More precisely, photons
  - Light sources emit photons
  - Surfaces reflect & absorb photons
  - Cameras measure photons

## Photons

- The quanta of electromagnetic energy
  - “Particles of light”
- Appear to be particles and waves simultaneously
  - We ignore wave behavior (i.e. diffraction)
  - We simulate particle behavior

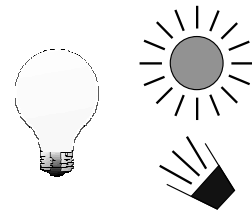
## Photons as Rays

- We model photons as “rays” of light
  - Rays in the mathematical sense
    - Defined by a starting point and a vector



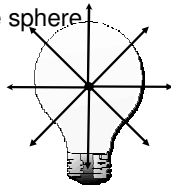
## Lights

- There are several kinds of lights (or light sources)
  - Light bulbs
  - The sun
  - Spot Lights
  - Ceiling Lights
- These are different because they emit photons differently



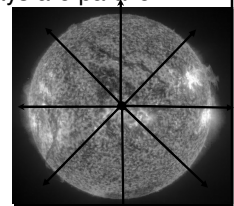
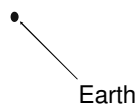
## Point Lights (i.e. Light Bulbs)

- Emit light evenly in all directions
  - That is, photons (rays) from a point light all originate from the same point, and have directions evenly distributed over the sphere



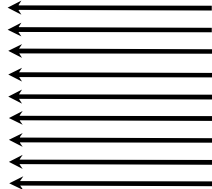
## Directional Lights (i.e. the Sun)

- Point light sources at an infinite (or near infinite) distance
  - Can assume that all rays are parallel



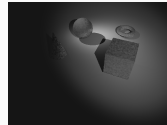
## Directional Lights (i.e. the Sun)

- Point light sources at an infinite (or near infinite) distance
- Can assume that all rays are parallel



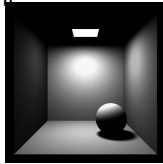
## Spot Lights (i.e., well, Spot Lights)

- Similar to point lights, but intensity of emitted light varies by direction
- Can think of it as only emitting rays over a patch on the sphere



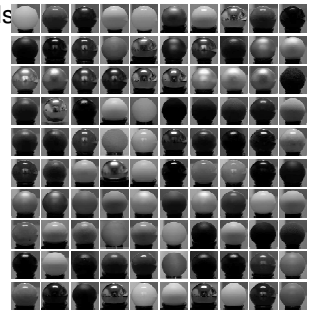
## Area Lights (i.e. Ceiling Lights)

- Emits light in every direction from a surface
- Can think of it as a set of point lights, or a patch on which every point is a point light

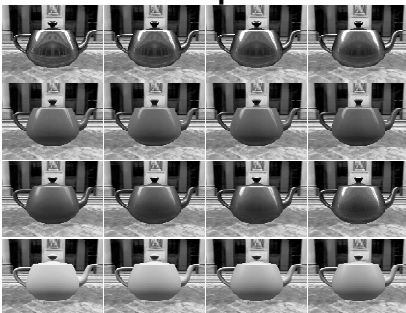


## Surfaces

- Surface materials can be:
  - Dull
  - Shiny
  - Bumpy
  - Smooth
  - Transparent
  - Translucent...



## Materials on the Utah Teapot



## Surfaces

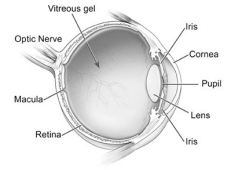
- So what do surfaces do?
  - They selectively react to incoming photons
    - Absorb
    - Reflect
    - Refract
    - Fluoresce
    - etc.

## Surfaces

- Surfaces make up the “interesting” things in a scene, but from a light transport point of view, they are just intermediaries

## Cameras

- Almost all cameras work in a similar fashion
- Use a lens to map rays from the scene (i.e. the world) onto an image plane

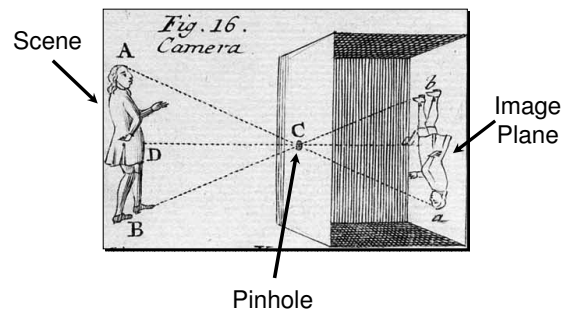


## Cameras

- Lenses are a distraction
- Let's simplify:
  - Camera Obscura (dark room)



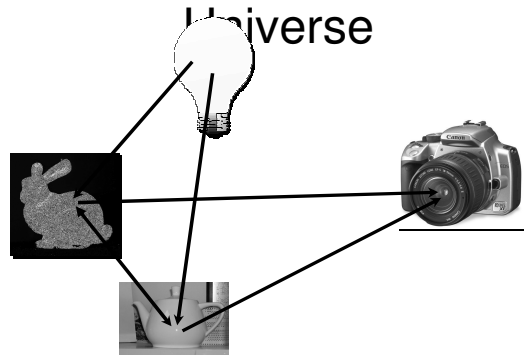
## Camera Obscura

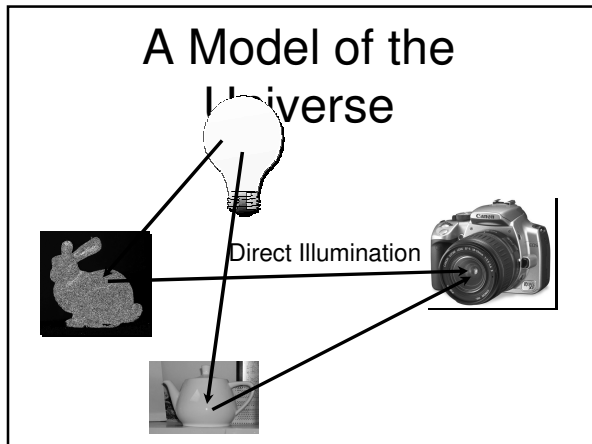


## Recap

- So now we know the basics of how lights, surfaces, and cameras interact
  - Lights emit rays
  - Surfaces change rays
  - Cameras capture rays
- We can use this to build a rendering system!

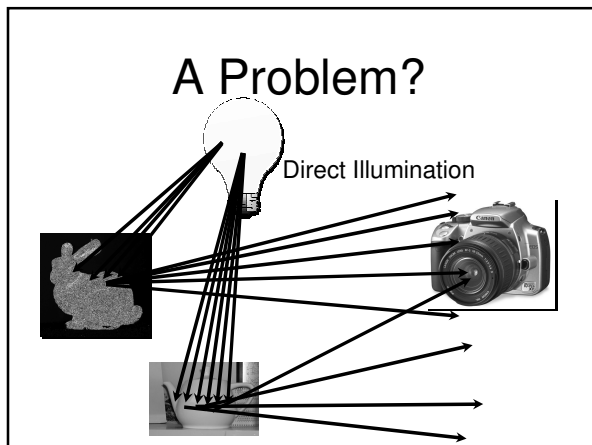
## A Model of the Universe





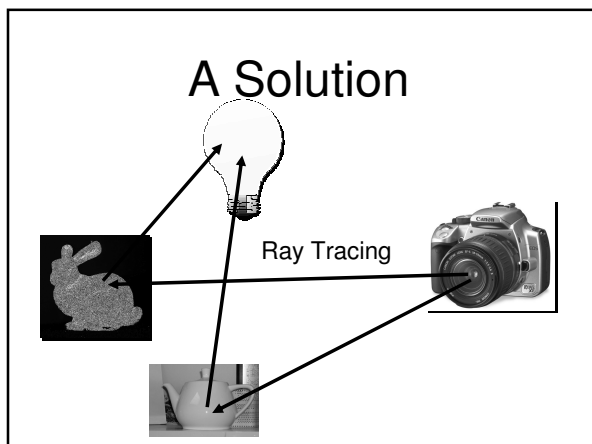
### A Problem?

- It is perfectly reasonable to try to implement an algorithm based on the model discussed so far
- However, it would be VERY inefficient
- Why?
  - Many (probably most) light rays in a scene would never hit the image plane



### A Solution

- Instead of shooting rays from the light, shoot rays from the camera
- Shoot one (or, realistically, many) rays per pixel
- Can stop when you hit a light and/or a non-reflective surface
- This is the basis of the ray tracing algorithm



### Ray-Tracing Algorithm

- for each pixel / subpixel
  - shoot a ray into the scene
  - find nearest object the ray intersects
  - if surface is (nonreflecting OR light)
    - color the pixel
  - else
    - calculate new ray direction
    - recurse

## Ray-Tracing Algorithm

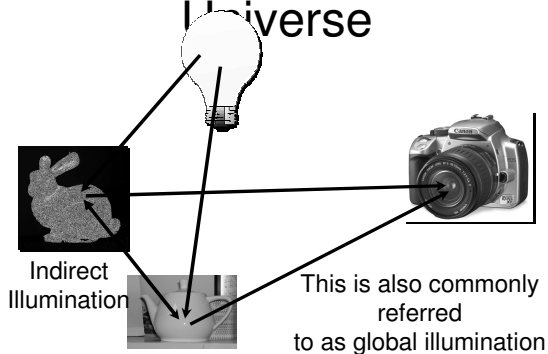
- for each pixel / subpixel  
shoot a ray into the scene  
find nearest object the ray intersects  
if surface is (nonreflecting OR light)  
color the pixel  
else  
calculate new ray direction  
recurse



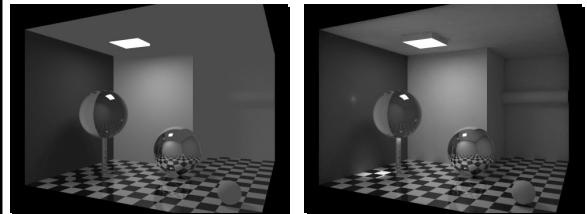
## Ray-Tracing Components

1. Generate the rays that are seen by the eye
  - One (or more) for each pixel
  - Need to determine ray origin / directions
2. Figure out what (if anything) those rays hit
  - Compute the nearest intersection
  - Ray-object intersections
3. Determine the color of the pixel

## A Model of the Universe



## Direct Illumination vs. Global Illumination



Direct Illumination Only

With Global Illumination

## Class Schedule

- Weeks 10 & 11
  - Real Cameras & Light Transport
  - Casting Rays
- Weeks 12 & 13
  - Ray Tracing
  - Radiosity
- Week 14
  - Photon Mapping

## Class Schedule

- Week 15
  - Special Topics (tentative)
    - High Dynamic Range (HDR) Rendering
    - Image-Based Rendering
    - Suggestions?
- Week 16
  - Course / Final Exam Review