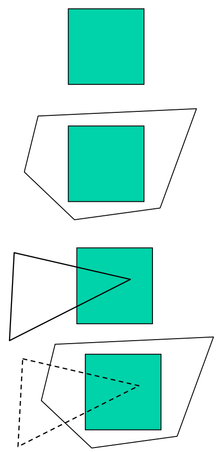
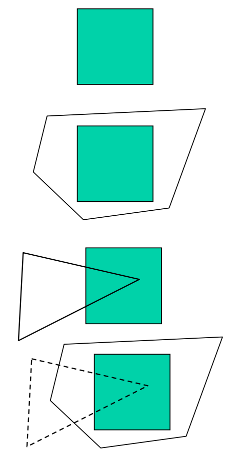
Area subdivision

- Four tractable cases for a given some region in image plane:
  - no surfaces project to the region
  - only one surface completely surrounds the region
  - only one surface is completely inside the region or overlaps the region
  - a polygon is completely in front of everything else in that region determined by considering depths of the polygons at the corners of the region



Area subdivision

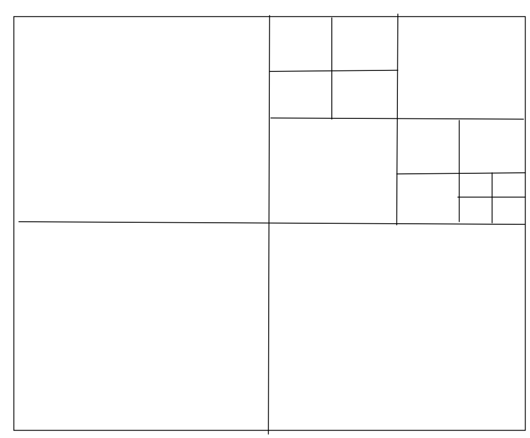
- Four tractable cases for a given some region in image plane:
  - no surfaces project to the region (paint background if needed, otherwise do nothing)
  - only one surface completely surrounds the region (paint surface)
  - only one surface is completely inside the region or overlaps the region (paint background if applicable and then scan convert region)
  - a polygon is completely in front of everything else in that region determined by considering depths of the polygons at the corners of the region (paint surface)



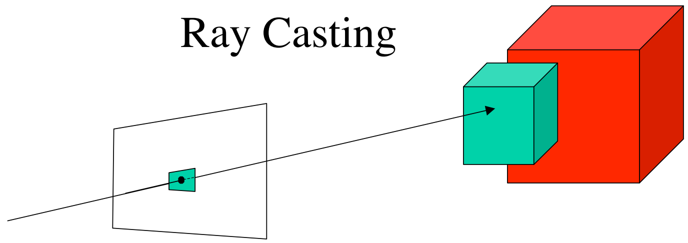
Area subdivision

- Algorithm:
  - subdivide each region until one of these cases is true or until region is very small
  - if case is true, deal with it
  - if region is small, choose surface with smallest depth.
  - determining cases quickly makes use of the same ideas in depth sort (depth sorting, bounding boxes, tests for which side of the plane), and the difficult cases are deferred by further subdivision.
- Advantages:
  - can be very efficient
  - no over rendering
  - anti-aliases well (subdivide a bit further)

One Subdivision Strategy



Ray Casting



- Image precision algorithm
- For each pixel cast a ray into the world
  - For each surface
    - determine intersection point with ray
  - Render pixel based on closest surface

Ray Casting

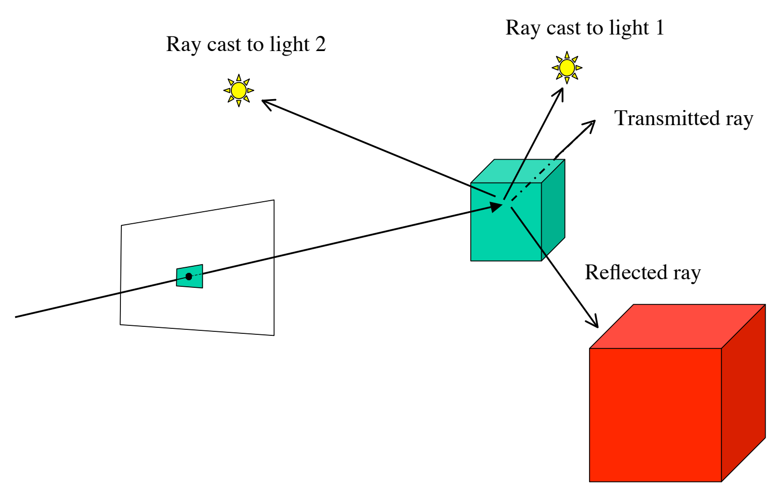
- First step in ray tracing algorithm
- Expensive
- Good performance usually requires clever data structures such as bounding volumes for object groups or storing world occupancy information in octrees.
- Other main problem is computing intersection.
- For polygons, we can use the standardized orthographic space where we can work in 2D.
- Spheres are easy (a bit more difficult in perspective space).
- Useful for “picking”--not expensive here (why?)

Ray Tracing--teaser

- Idea is very simple--follow light around
- Following **all** the light around is intractable, so we follow the light that makes **the most** difference
- Work backwards from what is seen
- Simple ray tracer
  - Cast a ray through each pixel (as in ray casting for visibility)
  - From intersection point cast additional rays to determine the color of the pixel.
    - For diffuse component, must cast rays to the lights
    - We may also add in some “ambient” light
    - For mirrors, must cast ray in mirror direction (recursion--what is the stopping condition)

Recursive ray tracing

H&B, page 597



Current state of intro students graphic's ability

Know how to draw polygons

Know about cameras

Know how to map 3D polygons onto the screen

Know how to draw the bits closest to the cameras

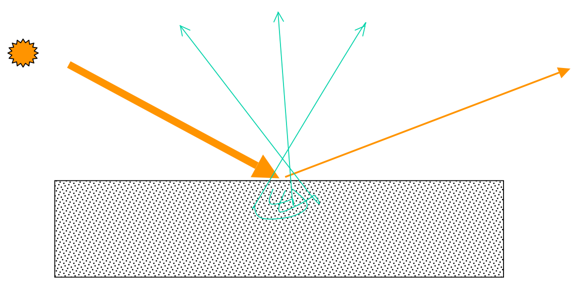
Issues

- Should we live in a polygonal world?
- How do you get polygons for complex objects?
- What color should each pixel be?

Light interacting with the world

- The signal reaching your eye from a surface is the result of the surface interacting with the light following on it.
- Many effects when light strikes a surface. It could be:
  - absorbed
  - transmitted
  - reflected
  - scattered (in a variety of directions!)

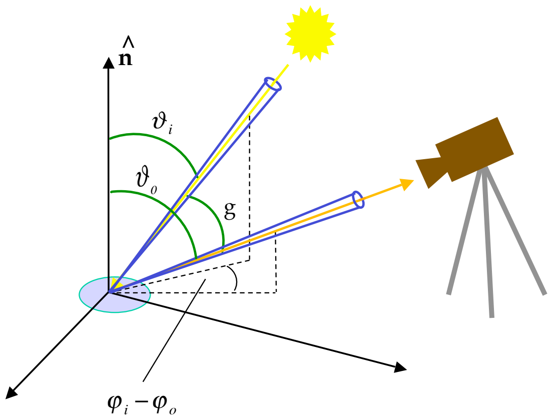
This shows some possibilities that can happen to the line from **one** direction.



Bidirectional Reflectance Distribution Function (BRDF)

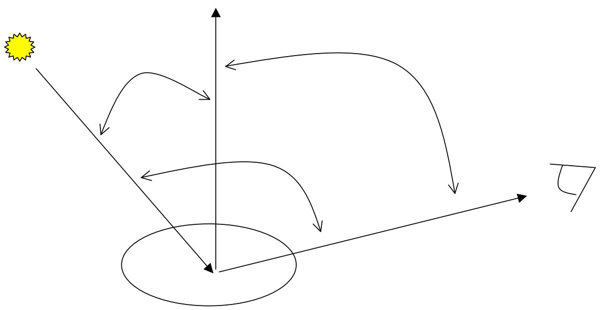
- The BRDF is a technical way of specifying how light from sources interacts with the matter in the world
- Understanding images requires understanding that this varies as a function of materials. The following "look" different
  - mirrors
  - white styrofoam
  - colored construction paper
  - colored plastic
  - gold
- The BRDF is the **ratio** of what comes out to what came in
- What comes out <--> "radiance"
- What goes in <--> "irradiance"
- Details on the BRDF available as supplementary material

This shows angular effects. There are also spectral (color effects).



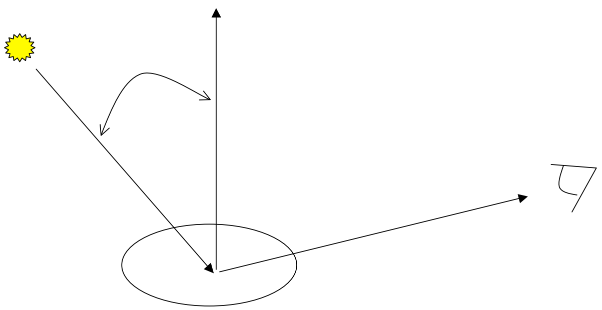
Isotropic surfaces

The BRDF for many surfaces can be well approximated as a function of 3 variables (angles), not 4. In this case, turning the surface around the normal has no effect. The surface is said to be *isotropic*.



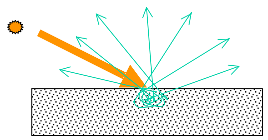
Lambertian surfaces

- Even simpler case--the BRDF does not depend on the viewing (output) direction (e.g., Lambertian).



Lambertian surfaces

- Simple special case of reflectance: ideal diffuse or matte surface--e.g. cotton cloth, matte paper.
- Surface appearance is independent of viewing angle.
- Typically such a surface is the result of lots of scattering---the light “forgets” where it came from, and it could end up going in any random direction.



- What counts is how much light power reaches the surface

Lambertian surfaces and albedo

- We will refer later to “radiosity” as a unit to describe light leaving the surface taken as whole
  - Technically, it is the total power leaving a point on the surface, per unit area on the surface (Wm^{-2})
- Recall that for a Lambertian surface, the direction that light leaves is not an issue.
- Percentage of light leaving the surface compared with that falling onto it, is often called diffuse reflectance, or *albedo* for a Lambertian surface.

Lambertian surfaces

The Lambertian assumption leads to very simple rule to shade an object. Specifically, we attenuate brightness by

$$\mathbf{n} \bullet \mathbf{s}$$

Must know this

Surface normal

Light source direction