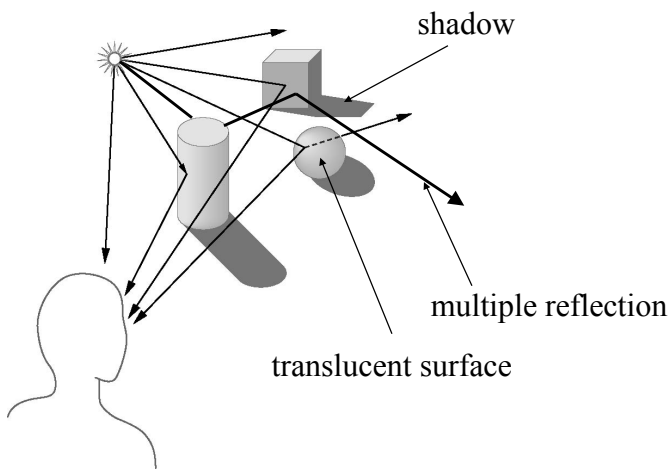
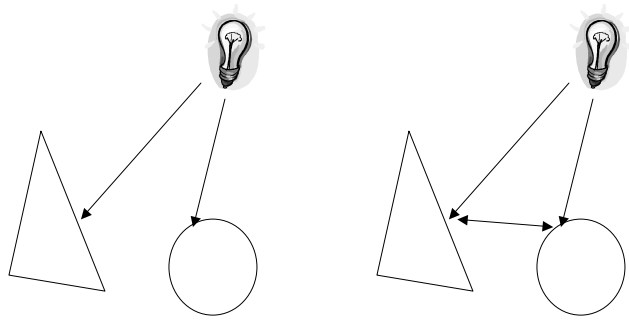
Global Illumination

CS5500 Computer Graphics
May 29, 2006

Global Effects



Local vs. Global



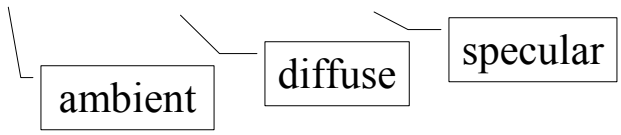
Local vs Global Rendering

- Correct shading requires a global calculation involving all objects and light sources
  - –Incompatible with pipeline model which shades each polygon independently (local rendering)
- However, in computer graphics, especially real time graphics, we are happy if things “look right”
  - –Exist many techniques for approximating global effects

Local Reflection Models

Phong Reflection Model

- I = Ka*Ia + kd*Id + Ks*Is



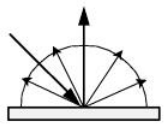
- Not completely correct, but good enough.

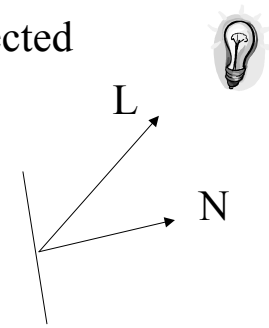
Ambient Component

- Accounting for light scatter around.
- Ia is constant.

Diffuse Component

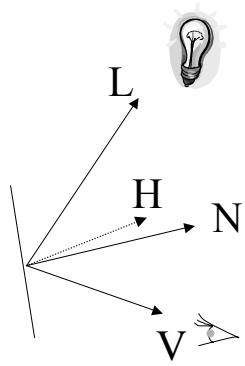
- $Ia = Ii * N \cdot L$
- Not affected by viewing direction.
  - i.e., incoming light is reflected to all directions.





Specular Component (Phong Reflection Model)

- To model imperfect reflection.
- $Is = Ii(N \cdot H)^n$



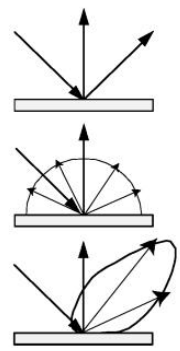
Specular Component (Cook & Torrance Model)

- Consider specular reflection as perfect reflection of micro-facets.
- Specular=DGF/($N \cdot V$)

  D: Distribution term

  G: Geometry (shadowing and masking) term

  F: Fresnel term

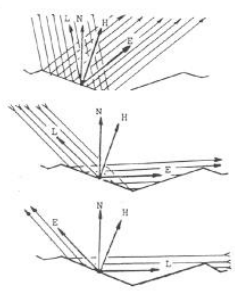


D Term (Cook & Torrance)

- Modeling the distribution of micro-geometry.
- Gaussian distribution can be used:

$$D = k\, e^{-(\alpha/m)2}$$

G Term (Cook & Torrance)



$$G = \min(G_a, G_b, G_c)$$

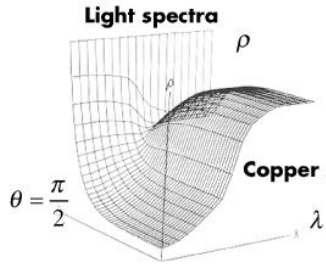
$$G_a = 1$$

$$G_b = \frac{2(\mathbf{N} \bullet \mathbf{H})(\mathbf{N} \bullet \mathbf{E})}{(\mathbf{H} \bullet \mathbf{E})}$$

$$G_c = \frac{2(\mathbf{N} \bullet \mathbf{H})(\mathbf{N} \bullet \mathbf{L})}{(\mathbf{H} \bullet \mathbf{L})}$$

The Fresnel Term

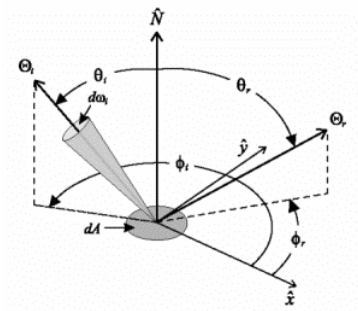
- Color of reflected light varies with viewing angle.

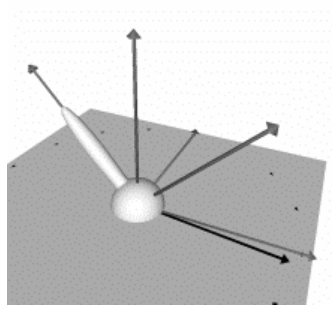


Reflectance of Copper as a function of wavelength and angle of incidence

BRDF

- BRDF=f(θ_{in}, ϕ_{in}, θ_{ref}, ϕ_{ref})=f(Light,View)





Why Not Always Using BRDF?

- Difficult to find a “closed form” representation of BRDF.
- The Phong model and Cook & Torrance model are approximation of BRDF.
  - They are not 100% matche of BRDF, but they are easy to compute.

Exercise

- Q1: Those reflectance models give us local illumination only. Why?

  Hint: What are the parameters? Do they include the other objects?
- Q2: What typical effects are missing?

  Shadow, reflection, and refraction…

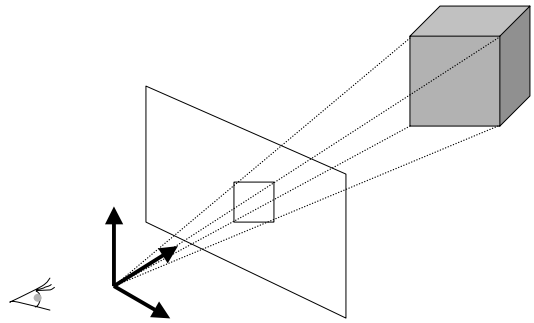
Object-Order vs. Screen-Order Rendering

How Do You Draw a Picture (Without a Computer)?

- What is your subject?
- Viewing Parameters:
  - Camera, Picture Frames, Resolutions
- Many ways to specify it:

(1) eye, focus length, image plane

(2) eye, direction, FOV, up vector



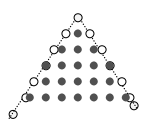
Method A: Object Order Algorithm
(Process one polygon at a time.)

3D to 2D Projection

- OK, so we can map a 3D point (or vertex) to 2D image.
- But what about a 3D surface?
- Polygons are made from points.
- Actually, we only need triangles!

Scan Conversion

- Also called rasterization.
- The 3D to 2D Projection gives us 2D vertices (points).
- We need to fill in the interior.



Shading

- Interpolation between any two vertices:
  - Vertex colors (lit or unlit?)
  - Normal vectors
  - Z
  - w (or 1/w) for perspective correction
  - Texture coordinates

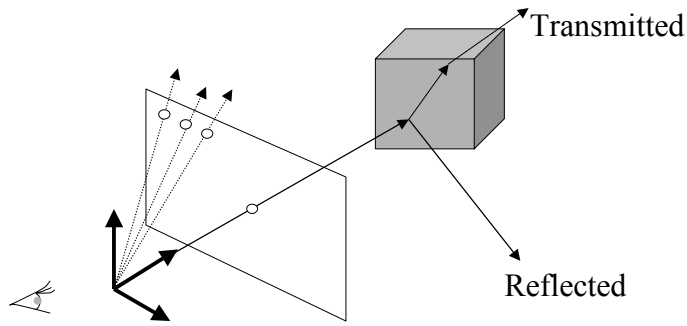
Note that we usually apply only local illumination in object order algorithm (because you have to go beyond the current object to obtain the global effect.)

Method B: Screen Order Algorithm

(Process one pixel at a time.)

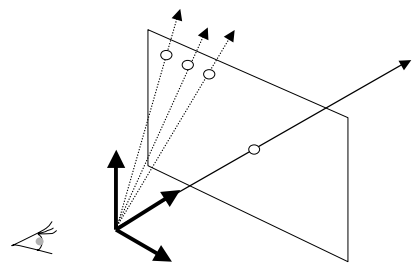
Ray Tracing Algorithm

- More detail in Watt's 10.3.1 (pp.284-286) and 12.2-12.4 (pp.342-354)



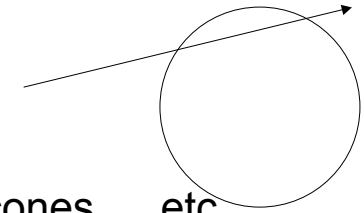
Creating a Ray

- Parameters:
  - Image Plane (position, size, and resolution)
  - Viewpoint
  - Which ray (x, y)?



Ray-Object Intersection

- For example: sphere
  $(x-x_0)^2+(y-y_0)^2+(z-z_0)^2=r^2$
- Ray: $(x,y)=(x_1,y_1)+t(x_d,y_d)$
- Find t that satisfy
  $(x-x_0)^2+(y-y_0)^2+(z-z_0)^2=r^2$
- Normal vector?
- Also easy for planes, cones, …etc.

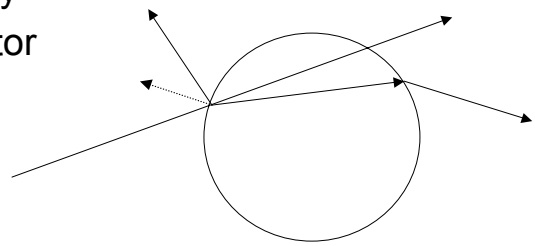


Shading Models

- Pixel color = ambient + diffuse + specular + reflected + transmitted
- The weight of each is determined by the surface properties.

Reflection and Refraction

- Reflected ray is determined by:
  - incoming ray and normal vector.
- Refracted ray is determined by:
  - Incoming ray
  - Normal vector
  - And density



Recursive Algorithm

- The reflected ray and refracted ray are traced recursively.
- Termination condition:
  - Depth of trace
  - Weight (to the final pixel color) of ray
- Pseudo code in Watt's page 349.

Advantage

- We get all the following automatically:
  - Hidden surface removal
  - Shadow
  - Reflection
  - Transparency and refraction

Disadvantage

- Slow. Many rays are spawned.
- Slow. Ray-object intersection for every ray and every object. (Space partition helps a lot!).
- The lighting is still not completely right!

Can you get this with ray tracing?

Other Global Illumination Techniques

- Radiosity
- Monte Carlo path tracing, for examples:
  - Distributed ray tracing
  - Metropolis light transport
  - Photon map…etc.
- Want to know more? Take Digital Image Synthesis next semester!

What is Radiosity

- Borrowed from radiative heat transfer.
- Assuming diffuse reflectance.
- View independent solution.

RADIANCE

© 1994 by Charles Ehrlich, Mark Mack

Photon Map

Metropolis Light Transport

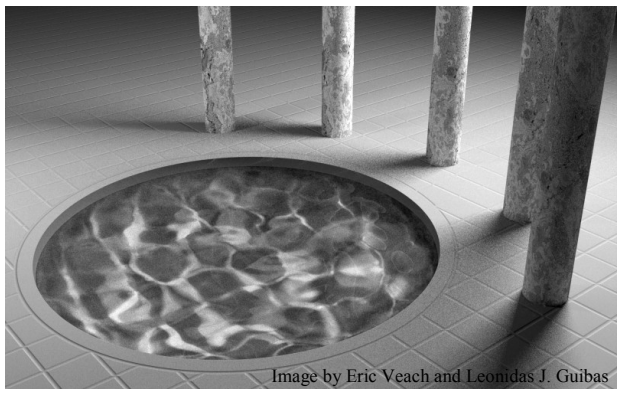


Image by Eric Veach and Leonidas J. Guibas