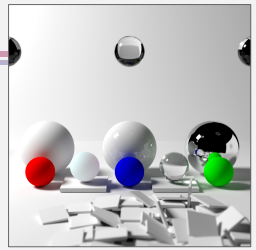
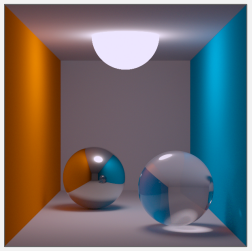
Path Tracing

Ingredients –

- rendering equation
- generalized notion of *material*
  - OpenGL materials only capture certain aspects of how light interacts with a surface
- path tracing algorithm
  - to approximate a solution to the rendering equation



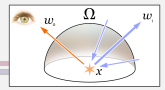


https://commons.wikimedia.org/wiki/File:Path_tracing_001.png
https://commons.wikimedia.org/wiki/File:Box_-_Path_Tracing_High.png

Principles of Optics

- principle of global illumination
  - in a closed environment, every object must contribute illumination to every other object
- principle of equivalence
  - no distinction between light emitted from a light source and light reflected from a surface
- principle of direction
  - light scattered in a particular direction is a function of the incoming and outgoing directions

Rendering Equation



light energy leaving a point in a particular direction =
light energy emitted by the point in that direction +
light energy arriving at the point which is scattered in that direction

$$\underbrace{L_o(\mathbf{x}, \omega_o, \lambda, t)}_{\text{light leaving in direction } \omega_o \text{ at wavelength } \lambda \text{ and time } t} = \underbrace{L_e(\mathbf{x}, \omega_o, \lambda, t)}_{\text{light emitted in direction } \omega_o} + \int_\Omega \underbrace{f_r(\mathbf{x}, \omega_i, \omega_o, \lambda, t)}_{\text{bidirectional reflectance distribution function}} \underbrace{L_i(\mathbf{x}, \omega_i, \lambda, t)}_{\text{light incident from direction } \omega_i \text{ at wavelength } \lambda \text{ and time } t} (\omega_i \cdot \mathbf{n}) \, \mathrm{d}\omega_i$$

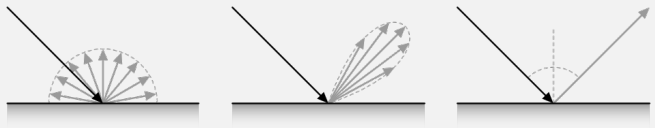
can be generalized to include transmission (bidirectional scattering distribution function)

- introduced by Jim Kajiya in 1986

https://en.wikipedia.org/wiki/Rendering_equation
https://en.wikipedia.org/wiki/Jim_Kajiya#/media/File:Jim_Kajiya.jpg

Materials

- when light hits a surface
  - some light is absorbed
  - the rest is scattered in various directions (some reflected, some transmitted)
- OpenGL materials capture certain aspects of this



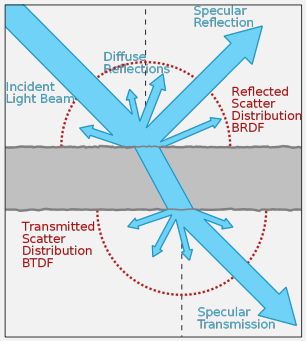
diffuse color | specular color shininess | reflection coefficient

https://en.wikipedia.org/wiki/Bidirectional_reflectance_distribution_function

BSDFs

Bidirectional Scattering Distribution Functions

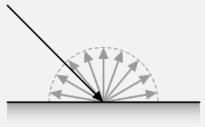
- BSDFs generalize the notion of material
  - BSDF = bidirectional scattering distribution function
    - combination of BRDF (bidirectional reflectance distribution function) and BTDF (bidirectional transmission distribution function)
  - expresses the proportion of incoming light scattered in various directions
    - function of incoming and outgoing directions



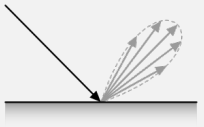
BSDFs

Bidirectional Scattering Distribution Functions

- OpenGL's notion of material can be expressed with BSDFs



$f(\omega_i,\omega_o)$
= md if $N\cdot\omega_i > 0$, $N\cdot\omega_o > 0$
= 0 otherwise



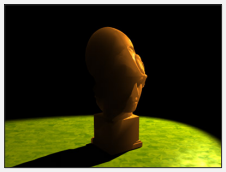
$f(\omega_i,\omega_o)$
= ms $(R\cdot V)^{mh}$ if $R\cdot V > 0$
= 0 otherwise

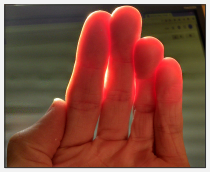
$f(\omega_i,\omega_o)$
= mr if $N\cdot\omega_i = N\cdot\omega_o > 0$
= 0 otherwise

BSDFs

Bidirectional Scattering Distribution Functions

- not limited to surfaces – any point in space can be assigned a BSDF
  - for empty space, $f(\omega_i,\omega_o) = 1$ if $\omega_i=\omega_o$
    $= 0$ otherwise
  - allows for materials like fog and clouds
  - allows for subsurface scattering in materials like marble, skin, milk





https://commons.wikimedia.org/wiki/File:Subsurface_scattering.jpg
https://commons.wikimedia.org/wiki/File:Skin_Subsurface_Scattering.jpg

Path Tracing

- Monte Carlo method for solving the rendering equation
  - uses repeated random sampling to approximate a solution
- basic algorithm – for solid objects
  - cast ray from viewer through pixel into scene
  - at each intersection, choose the next ray to follow according to the BSDF distribution for that point
  - repeat until ray hits a light source (which only emits), a skybox, or nothing

  } repeat many times for each pixel and average the colors

  - each point's color is determined by the color of the emitter or skybox, colors of surfaces hit along the way, and angles at which light hits each surface
    = emitted light + incoming light × fraction reaching surface × fraction reflected in outgoing direction

Volumetric Path Tracing

- can extend the rendering equation to include a scattering term
- idea
  - use BSDF for the medium to determine a probability distribution on how far light will travel before being scattered
  - choose the next ray direction and length according to the BSDF for that point