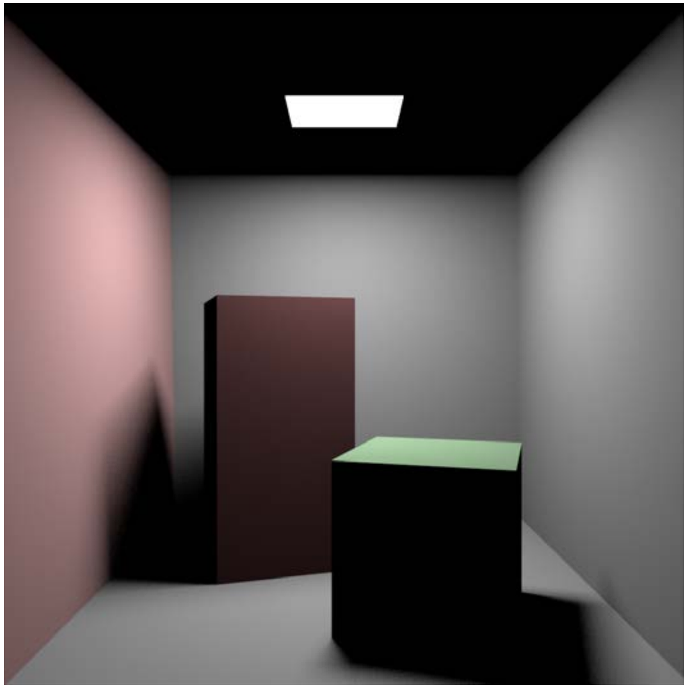
CT5503: Photorealistic Rendering

Global Illumination

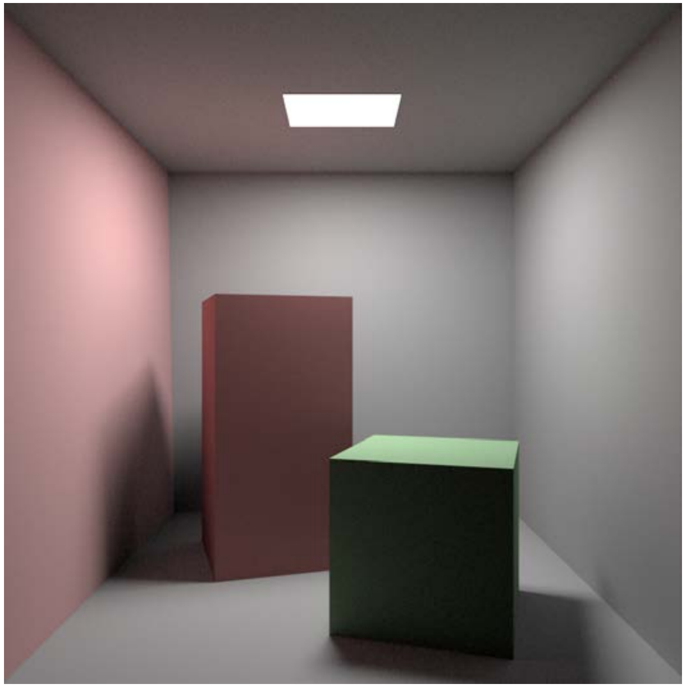
BOCHANG MOON

Global Illumination

- Global illumination methods consider both direct and indirect lighting



Without indirect lighting



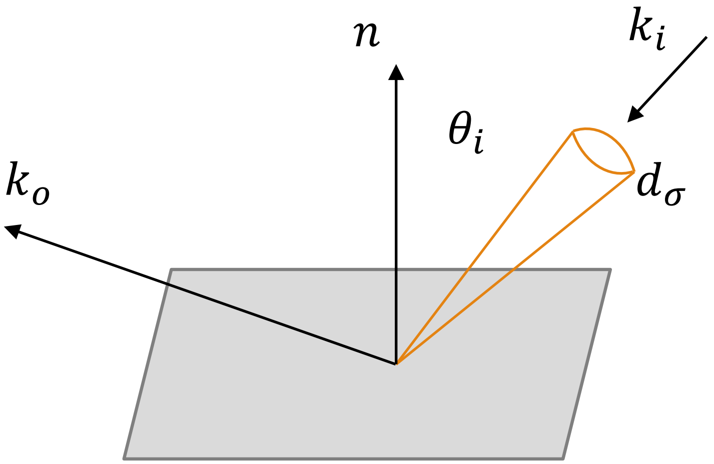
With indirect lighting

Radiosity

- $L_s(k_o) = \int_{all\ k_i} \rho(k_i, k_o) L_f(k_i) cos\theta_i d\sigma_i$

- Lambertian surfaces (ideal diffuse surface)
  - $\rho(k_i, k_o) = \frac{R}{\pi}$
  - R: diffuse reflectance

- Assumption: all surfaces are Lambertian
- $L_s(k_o) = \int_{all\ k_i} \rho(k_i, k_o) L_f(k_i) cos\theta_i d\sigma_i$
- $\quad = \frac{R}{\pi} \int_{all\ k_i} L_f(k_i) cos\theta_i d\sigma_i$

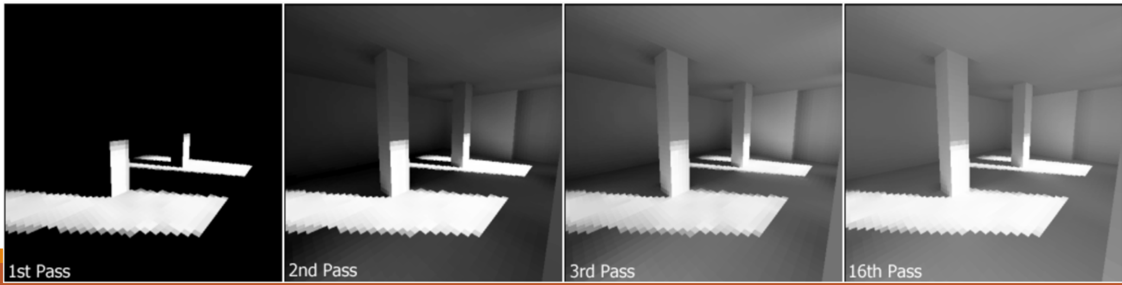


Radiosity

- $L_s(k_o) = \frac{R}{\pi} \int_{all\ k_i} L_f(k_i) cos\theta_i d\sigma_i$

- Finite element methods
  - Divide the scene into N small surfaces (patches) with unknown surface radiance L_i, reflectance R_i, and emitted radiance E_i
  - Then, the integral can be approximated with the N linear equations below:
  - $L_i = E_i + \frac{R_i}{\pi} \sum_{j=1}^{N} k_{ij} L_j$
  - k_{ij}: a constant related to the integral (form factor)
    - Fraction of light leaving a patch i arriving at a patch j
  - This results in N constant-colored polygons
  - Called *radiosity*

from wikipedia



Path Tracing

- $L_s(k_o) = L_e(k_o) + \int_{all\ k_i} \rho(k_i, k_o) L_f(k_i) cos\theta_i d\sigma_i$
- Monte Carlo integration
  - $\int_{x \in S} g(x) d\mu \approx \frac{1}{N} \sum_{i=1}^{N} \frac{g(x_i)}{p(x_i)}$
  - When N=1,
  - $L_s(k_o) \approx L_e(k_o) + \frac{\rho(k_i, k_o) L_f(k_i) cos\theta_i d\sigma_i}{p(k_i)}$
  - Need to do:
    - Select a random direction k_i
    - Evaluate $L_f(k_i)$

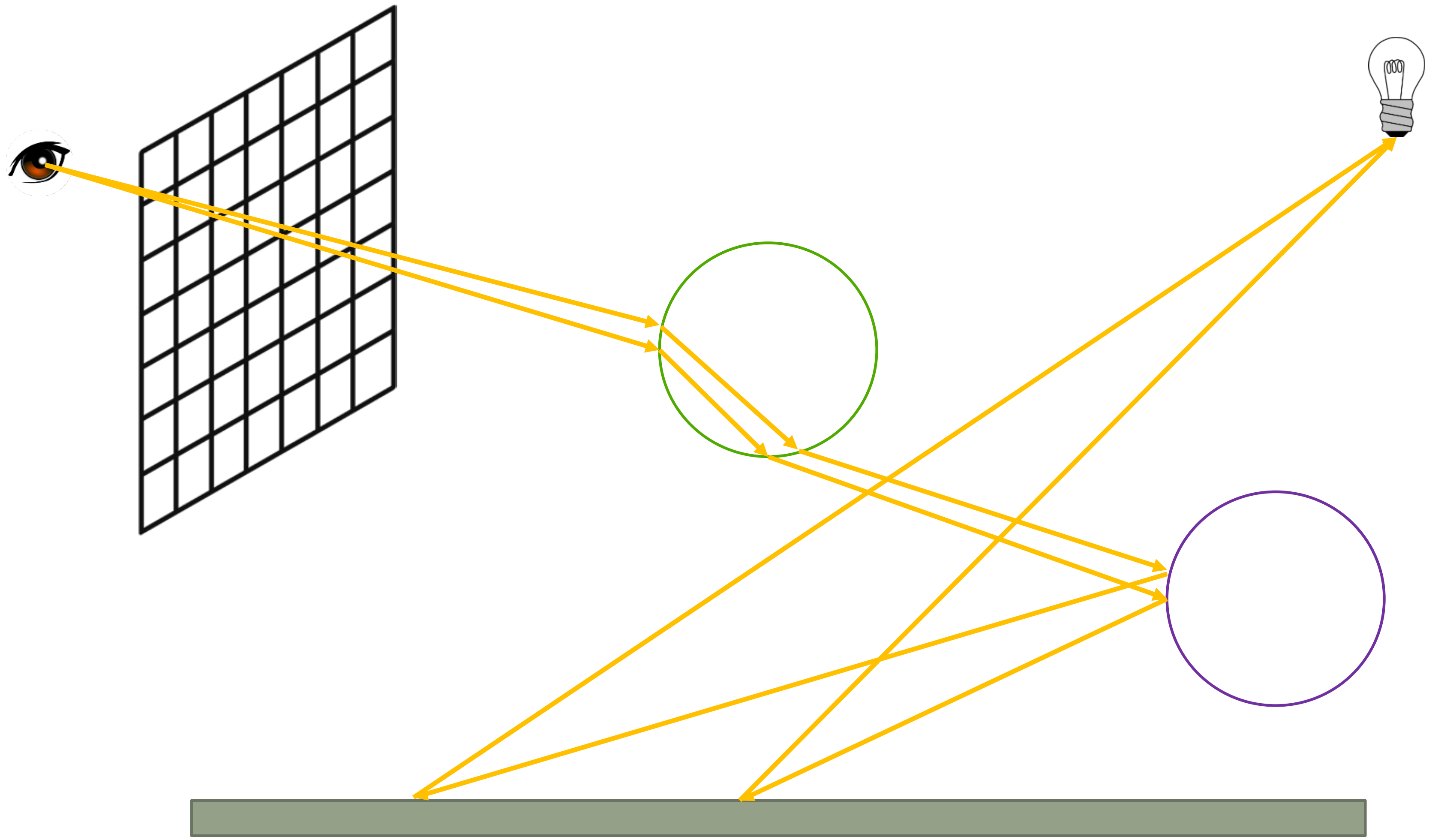
Path Tracing

- $L_s(k_o) \approx L_e(k_o) + \frac{\rho(k_i, k_o) L_f(k_i) \cos\theta_i d\sigma_i}{p(k_i)}$
- In case of the ideal diffuse surface:
  - $\rho = \frac{R}{\pi}$
  - When we choose a density function $p(k_i) = \frac{\cos\theta_i}{\pi}$
    - $L_s(k_o) \approx L_e(k_o) + R L_f(k_i)$
    - Note that we can cancel out the cosign terms

Path Tracing

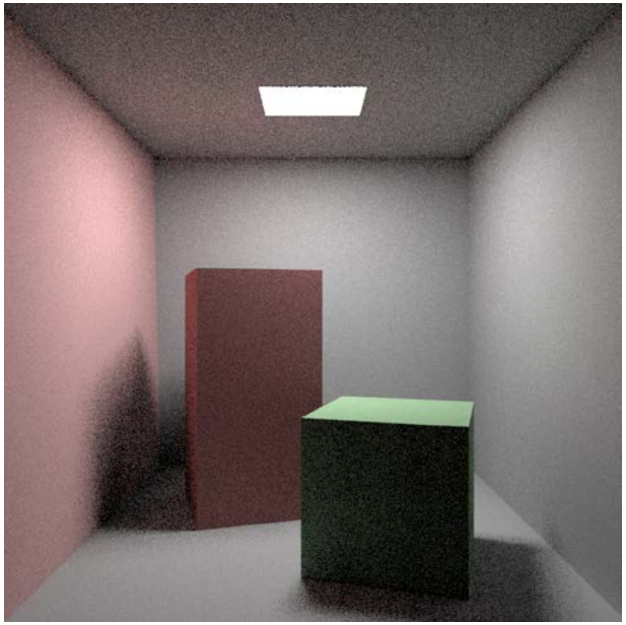
- Procedure
  - $RGB\ trace(ray\ a + tb, int\ depth)$
  - $if\ (ray\ hits\ at\ a\ point\ c)$
    - $RGB\ c = L_e(-b)$
    - $if\ (depth < maxdepth)$
      - $compute\ a\ random\ direction\ d$
      - $return\ c + R \times trace(c + sd, depth + 1)$
  - $else$
    - $return\ background\ color$

Path Tracing

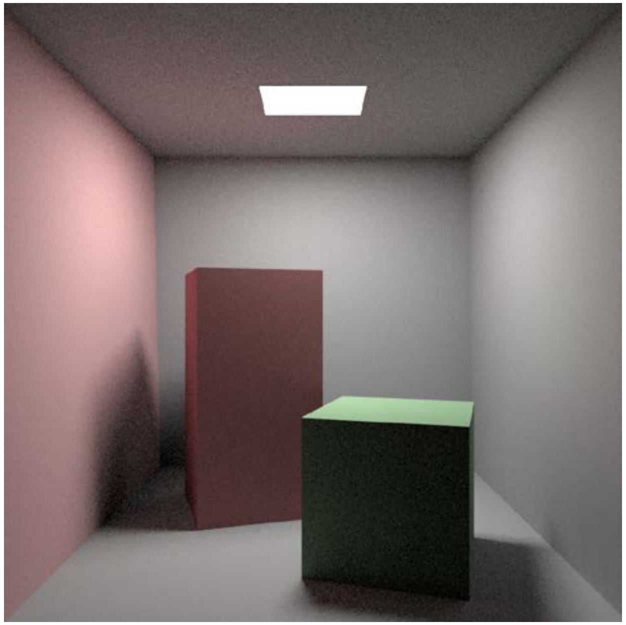


Path Tracing

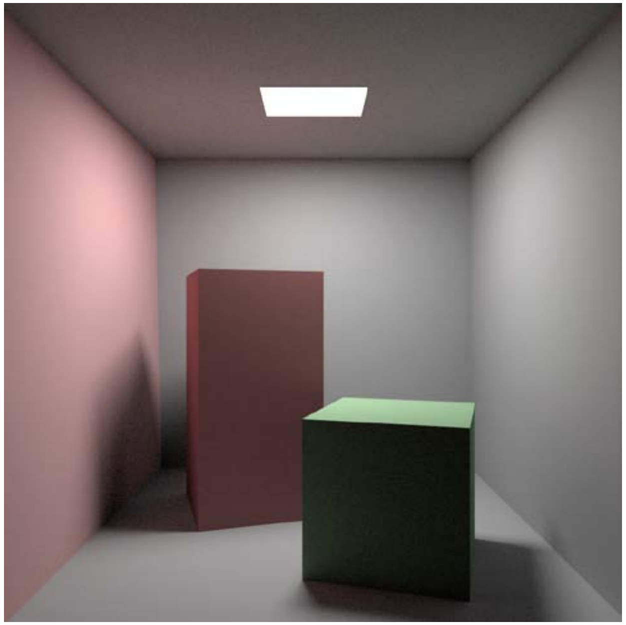
- A general rendering method that solves the full light transport equation (i.e., rendering equation)
- For each pixel color, it makes multiple ray paths, then averages the colors from the ray paths



4 samples / pixel (1.25 secs)



16 samples / pixel (5 secs)



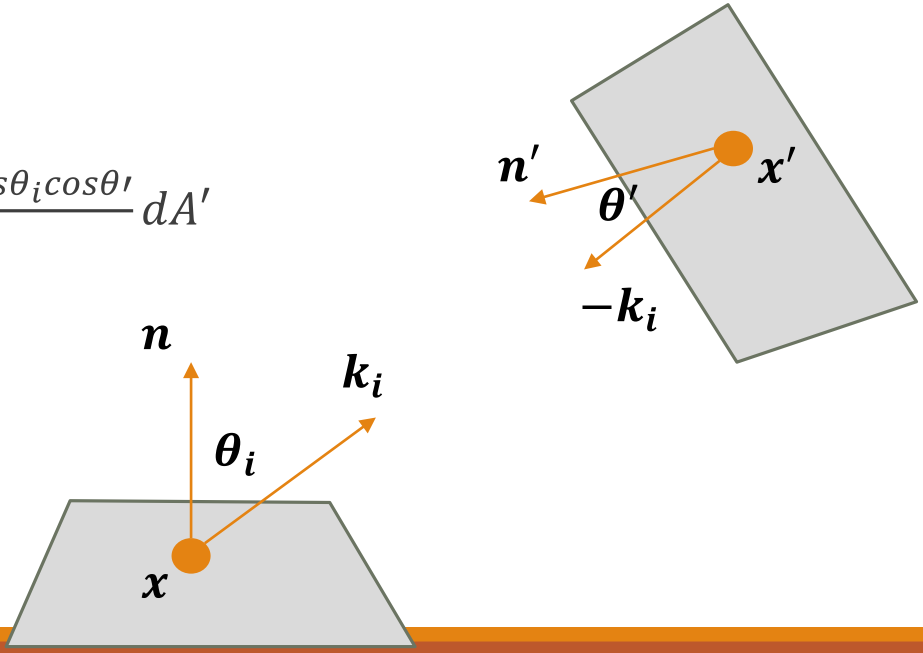
64 samples / pixel (20 secs)

Direct Lighting

- GI methods include the direct lighting as well as the indirect lighting

- Full light transport
  - $L_s(x, k_o) = \int_{all\ x'} \frac{\rho(k_i,k_o)L_s(x',x-x')v(x,x')cos\theta_i cos\theta'}{||x-x'||^2} dA'$

- Light transport for direct lighting
  - $L_s(x, k_o) = \int_{all\ x'} \frac{\rho(k_i,k_o)L_e(x',-k_i)v(x,x')cos\theta_i cos\theta'}{||x-x'||^2} dA'$



Direct Lighting

- $L_s(x, k_o) = \int_{all\ x'} \frac{\rho(k_i,k_o)L_e(x',-k_i)v(x,x')cos\theta_i cos\theta\prime}{||x-x'||^2} dA'$
- Sample a point x' on a luminaire with density function p ($x' \sim p$)
- $L_s(x, k_o) \approx \frac{\rho(k_i,k_o)L_e(x',-k_i)v(x,x')cos\theta_i cos\theta\prime}{p(x')||x-x'||^2}$
- Pick a uniform random point x' from the luminaire
  - $p = \frac{1}{A}$ (A is the area of the luminaire)
  - $L_s(x, k_o) \approx \frac{\rho(k_i,k_o)L_e(x',-k_i)v(x,x')cos\theta_i cos\theta' A}{||x-x'||^2}$

Direct Lighting

- $L_s(x, k_o) \approx \frac{\rho(k_i,k_o)L_e(x',-k_i)v(x,x')cos\theta_i cos\theta' A}{||x-x'||^2}$

- Procedure for a planar (e.g.., rectangular) light
  - $RGB\ directLight(x, k_o, n)$
  - $Sample\ a\ random\ point\ x' with\ normal\ vector\ n' on\ a\ light$
  - $d = x' - x$
  - $k_i = \frac{d}{||d||}$
  - $if\,(ray\ x + td\ does\ not\ hit\ any\ objects\ between\ the\ origin\ and\ light)$
    - $return\ \rho(k_i, k_o)L_e(x', -k_i)max(0, n \cdot d)max(0, -n' \cdot d)/||d||^4$ (note: $n \cdot d = ||d||cos\theta$)
  - $else$
    - $return$ 0