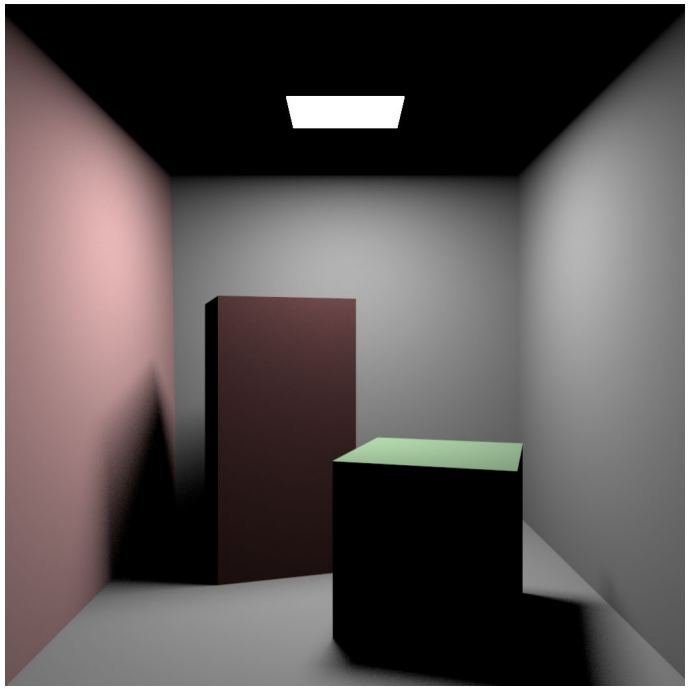
Lecture slides (AI3501/CT4201/EC4215 – Computer Graphics)

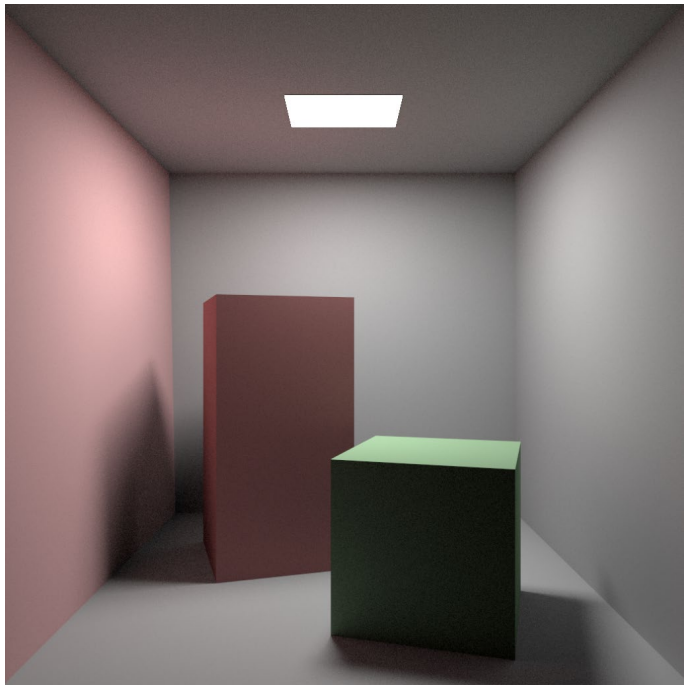
Global Illumination Methods

Lecturer: Bochang Moon

Global Illumination

- Distributed ray tracing introduces a noticeable improvement on rendering quality, but it still misses some rendering effects

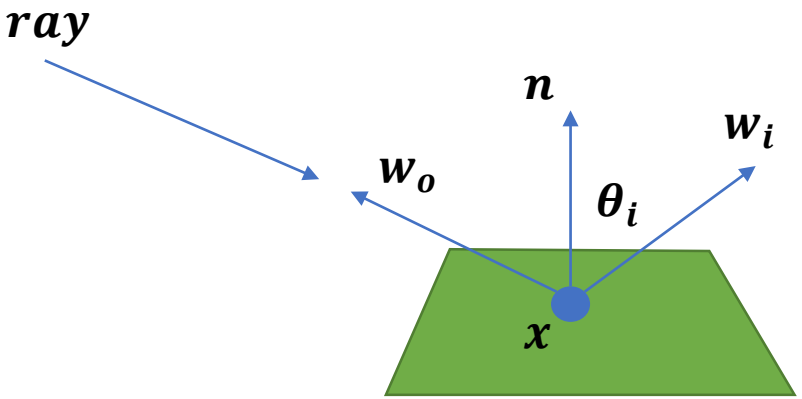


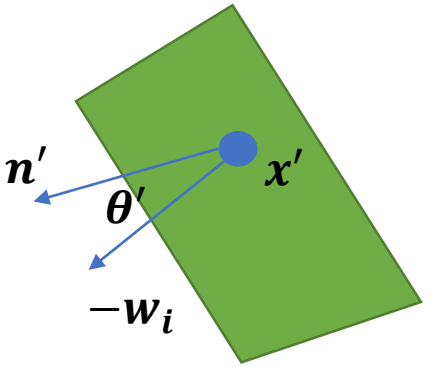


Global Illumination

- Rendering equation [Kajiya 86] (another form: [Immel and Cohen 86])

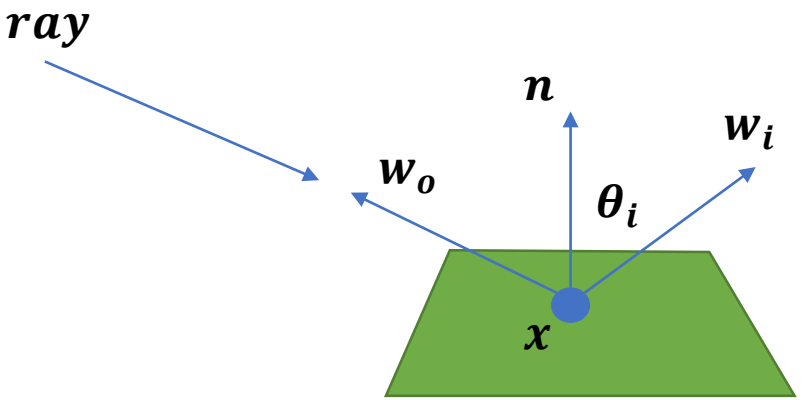
- $L_s(\boldsymbol{x}, w_o) = \int_{all\ x'} \frac{\rho(\boldsymbol{w_i},\boldsymbol{w_o}) L_s(\boldsymbol{x'},\boldsymbol{x}-\boldsymbol{x'}) v(\boldsymbol{x},\boldsymbol{x'}) cos\theta_i cos\theta'}{\|\boldsymbol{x}-\boldsymbol{x'}\|^2} dA$

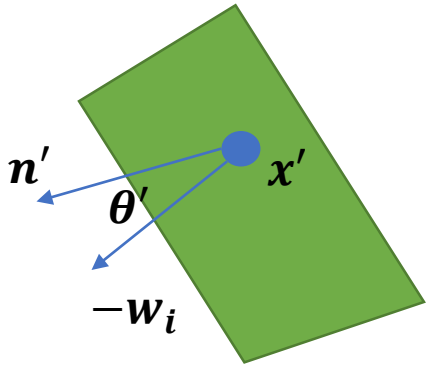




Global Illumination

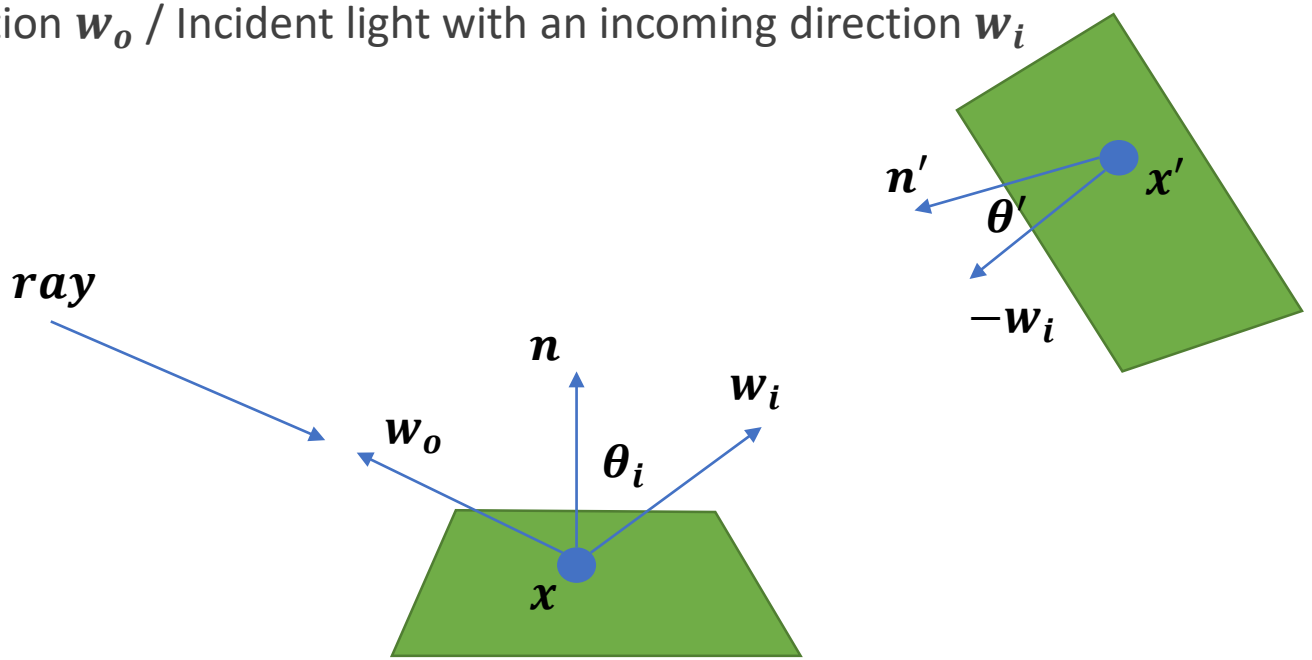
- $L_s(\boldsymbol{x}, w_o) = \int_{all\ x'} \frac{\rho(\boldsymbol{w_i}, \boldsymbol{w_o}) L_s(\boldsymbol{x'}, \boldsymbol{x}-\boldsymbol{x'}) v(\boldsymbol{x}, \boldsymbol{x'}) cos\theta_i cos\theta'}{\|\boldsymbol{x}-\boldsymbol{x'}\|^2} dA$
- $v(\boldsymbol{x}, \boldsymbol{x'})$: visibility term
  - 1 if two points are mutually visible
  - 0 otherwise





Global Illumination

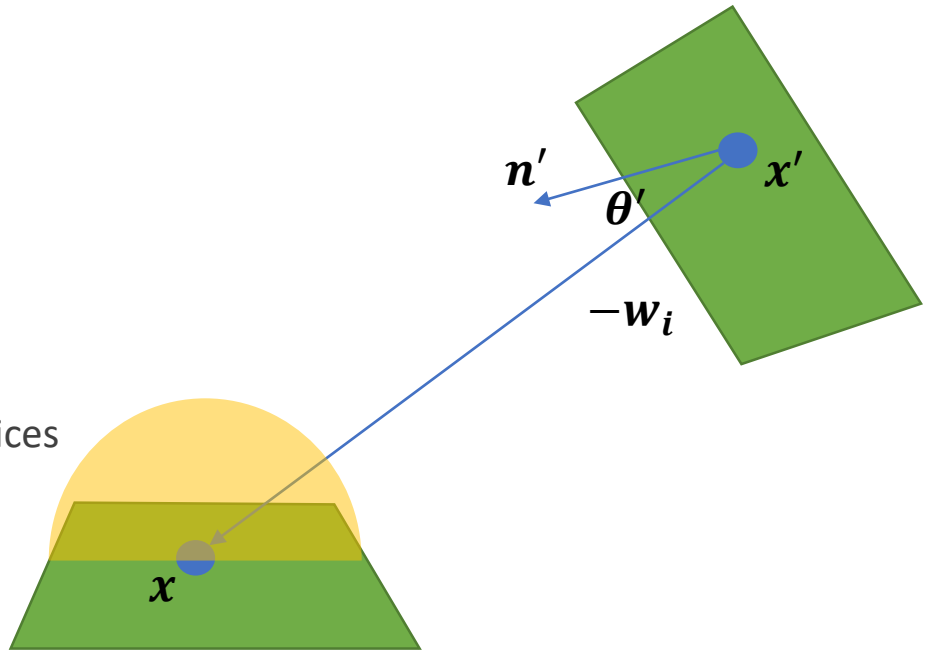
- $L_s(\boldsymbol{x}, w_o) = \int_{all\ \boldsymbol{x'}} \frac{\rho(\boldsymbol{w_i},\boldsymbol{w_o}) L_s(\boldsymbol{x'},\boldsymbol{x}-\boldsymbol{x'}) v(\boldsymbol{x},\boldsymbol{x'}) cos\theta_i cos\theta'}{\|\boldsymbol{x}-\boldsymbol{x'}\|^2} dA$
- $\rho(\boldsymbol{w_i}, \boldsymbol{w_o})$: bidirectional reflectance distribution function (BRDF)
  - Reflected light with an outgoing direction $\boldsymbol{w_o}$ / Incident light with an incoming direction $\boldsymbol{w_i}$
  - Determine appearance of a surface



Global Illumination

- $L_S(\boldsymbol{x}, w_o) = \int_{all\ \boldsymbol{x'}} \frac{\rho(\boldsymbol{w_i},\boldsymbol{w_o}) L_S(\boldsymbol{x'},\boldsymbol{x}-\boldsymbol{x'}) v(\boldsymbol{x},\boldsymbol{x'}) cos\theta_i cos\theta'}{\|\boldsymbol{x}-\boldsymbol{x'}\|^2} dA$

- $\rho(\boldsymbol{w_i}, \boldsymbol{w_o})$: bidirectional reflectance distribution function (BRDF)
  - Reflected light with an outgoing direction $\boldsymbol{w_o}$ / Incident light with an incoming direction $\boldsymbol{w_i}$
  - Determine appearance of a surface
  - e.g. ideal diffuse surface
    - $\rho(\boldsymbol{w_i}, \boldsymbol{w_o}) = \rho$ (constant)
  - e.g. other surfaces?
    - Utilize a mathematical form for well-known materials
    - Build the 4D function by measuring the fraction using actual devices



Global Illumination

- $L_s(\boldsymbol{x}, w_o) = \int_{all\ \boldsymbol{x'}} \frac{\rho(\boldsymbol{w_i},\boldsymbol{w_o}) L_s(\boldsymbol{x'},\boldsymbol{x}-\boldsymbol{x'}) v(\boldsymbol{x},\boldsymbol{x'}) cos\theta_i cos\theta'}{\|\boldsymbol{x}-\boldsymbol{x'}\|^2} dA$

- How can we solve the integral?

Monte Carlo Integration

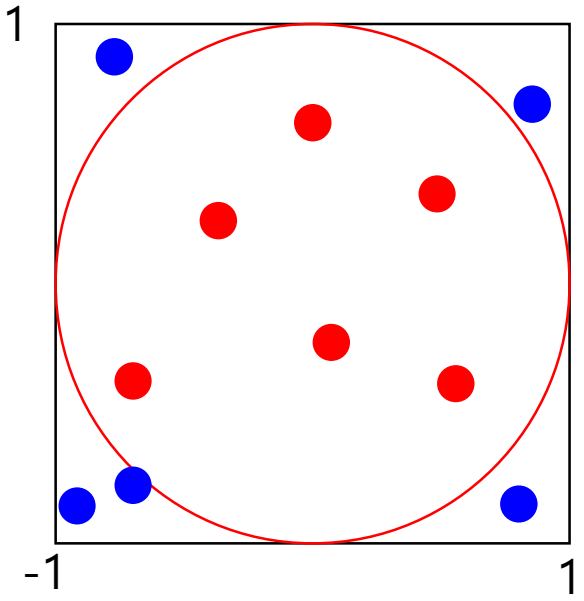
- A sampling method based on probability
- Draw random numbers and approximate an integral
- e.g., evaluate a 1D integral: $\int_a^b f(x)dx$
- $F_n = \frac{b-a}{n}\sum_{i=1}^{n} f(x_i)$
- $x_i \in [a,b]$: uniform random variable, $p(x) = \frac{1}{b-a}$
- $E[F_n] = \frac{b-a}{n}\sum_{i=1}^{n} E[f(x_i)] = \frac{b-a}{n}\sum_{i=1}^{n}\int_a^b f(x)p(x)dx = \frac{1}{n}\sum_{i=1}^{n}\int_a^b f(x)dx$
- $= \int_a^b f(x)dx$

Monte Carlo Integration

- A sampling method based on probability
- Draw random numbers and approximate an integral
- e.g., evaluate a 1D integral: $\int_a^b f(x)dx$
- $F_n = \frac{b-a}{n}\sum_{i=1}^{n} f(x_i)$
- $x_i \in [a, b]$: from a probability density function $p(x)$
  - $F_n = \frac{1}{n}\sum_{i=1}^{n} \frac{f(x_i)}{p(x_i)}$
  - Indicate that we can control sampling density by adjusting $p(x)$
  - Require a good sampling strategy (e.g., *important sampling* – a research area)
  - The approximation error decreases as the number of samples increases

Monte Carlo Integration

- $f(x,y) = 1$ if $x^2 + y^2 \leq 1$
- $= 0$ otherwise



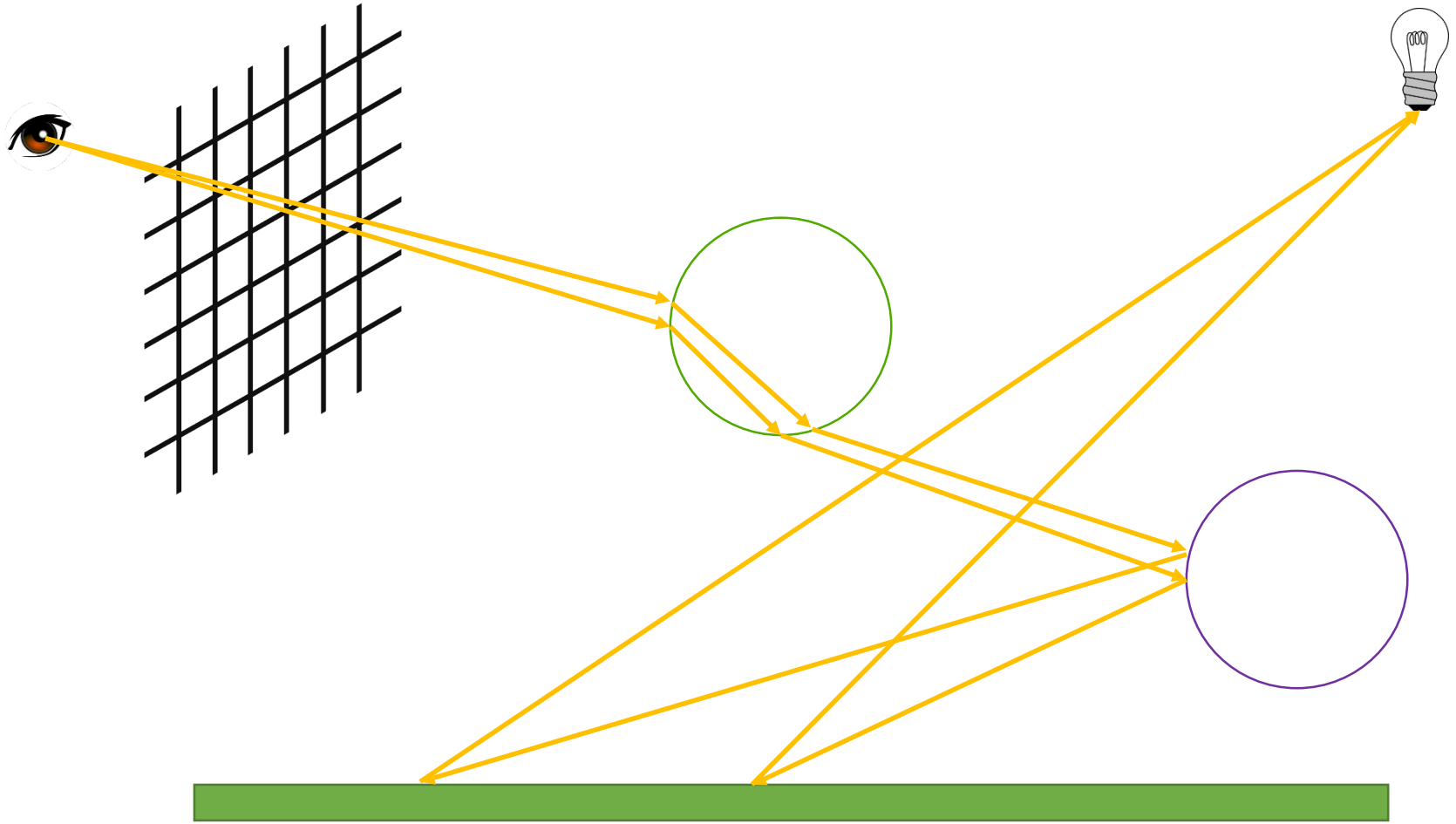
- $I = \int_{\Omega} f(x,y)dxdy = \pi$
- $\Omega = [-1,-1] \times [1,1]$
- $I \approx \frac{V}{n}\sum_{i=1}^{n} f(x_i, y_i)$
- V is the volume of the Ω
  - $V = 4$
- (x_i, y_i) has a uniform pdf

Monte Carlo Integration

- $L_s(\boldsymbol{x}, w_o) = \int_{all\ \boldsymbol{x'}} \frac{\rho(\boldsymbol{w_i},\boldsymbol{w_o}) L_s(\boldsymbol{x'},\boldsymbol{x}-\boldsymbol{x'}) v(\boldsymbol{x},\boldsymbol{x'}) cos\theta_i cos\theta'}{\|\boldsymbol{x}-\boldsymbol{x'}\|^2} dA$

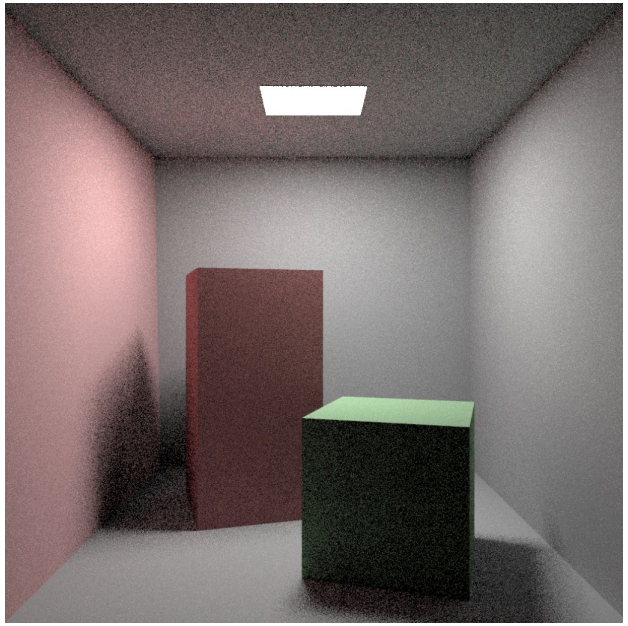
- Monte Carlo ray tracing
  - e.g., distributed ray tracing, path tracing
  - A random sample x_i is a high-dimensional vector
  - $x_i = [random\ numbers\ on\ image\ plane, lens, time, area\ lights, for\ secondary\ rays]$
  - Essentially the integral is a high-dimensional integral
  - The pixel color is an output by averaging the colors from multiple light paths

Path Tracing

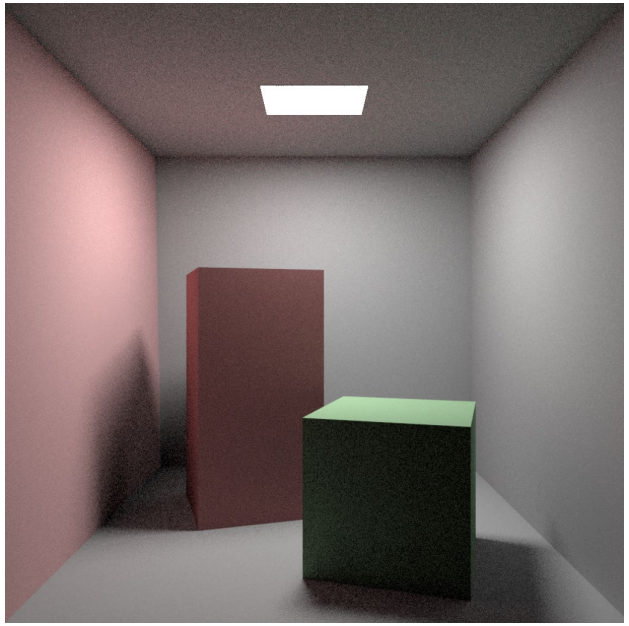


Path Tracing

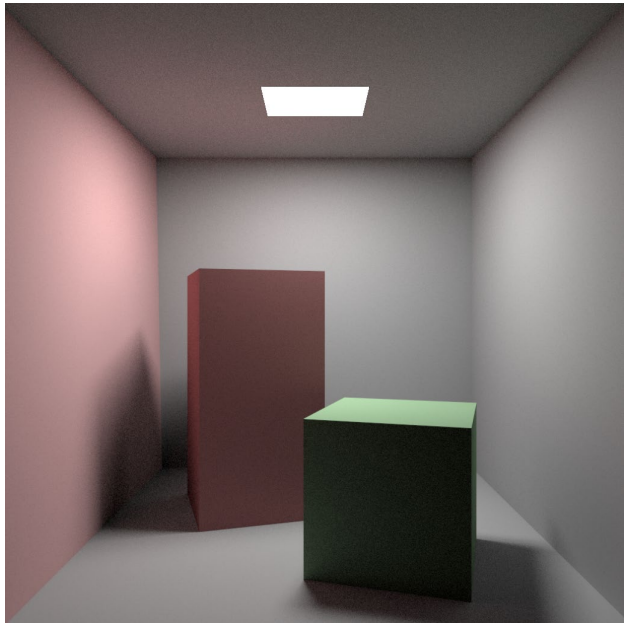
- A general rendering method that solves the full light transport equation (i.e., rendering equation)
- For each pixel color, it makes multiple ray paths, then averages the colors from the ray paths



4 samples / pixel (1.25 secs)



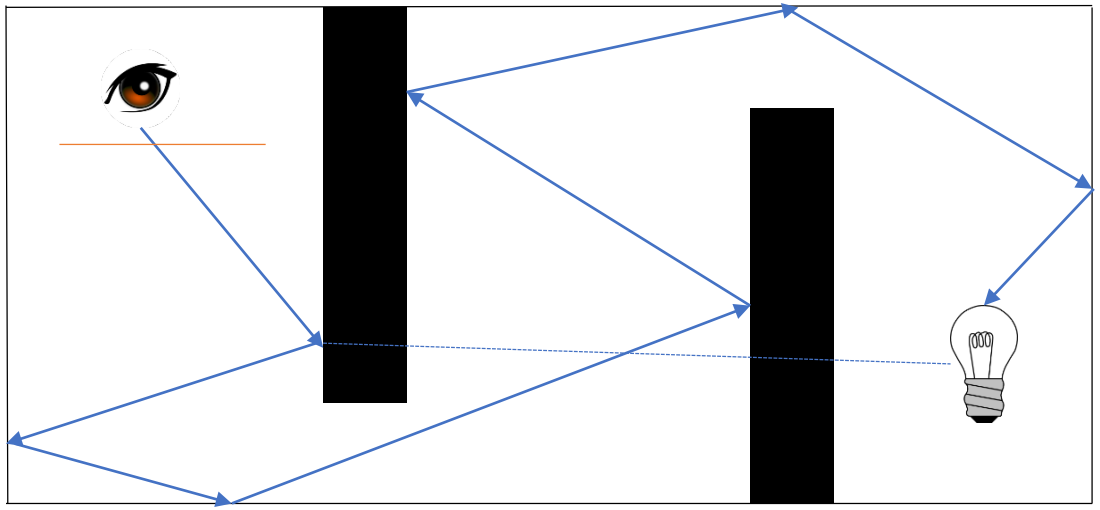
16 samples / pixel (5 secs)



64 samples / pixel (20 secs)

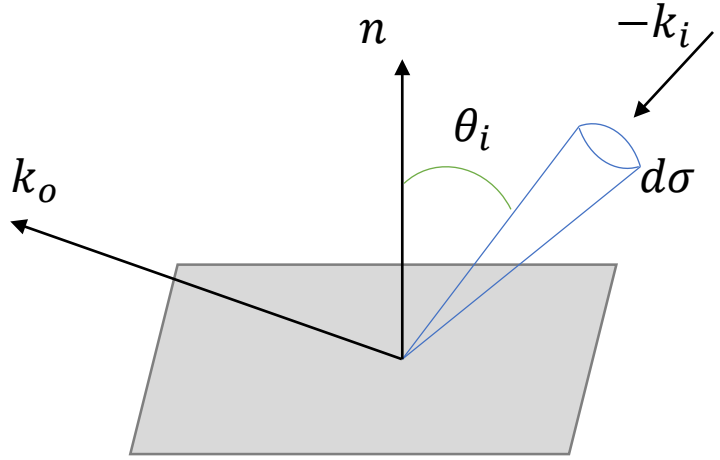
Path Tracing

- Find each light path starting from eye
  - Can be inefficient when light paths are difficult to reach from the eye



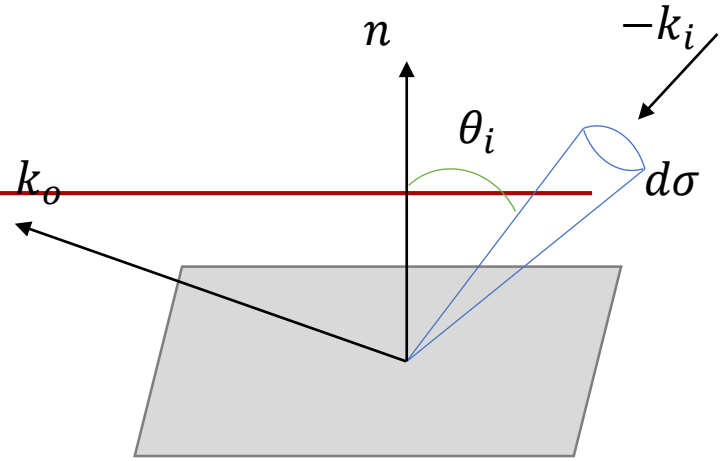
Rendering Equation in another form

- $L_s(k_o) = \int_{all\ k_i} \rho(k_i, k_o) L_f(k_i) cos\theta_i d\sigma_i$
  - $L_f(k_i)$: field radiance from k_i direction
  - $L_s(k_o)$: surface radiance measured in k_o direction
  - $L_s(-k_i) = L_f(k_i)$
  - *Rendering equation* [Immel, Cohen & Greenberg, 1986]

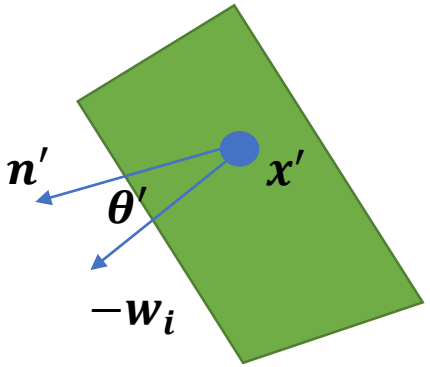


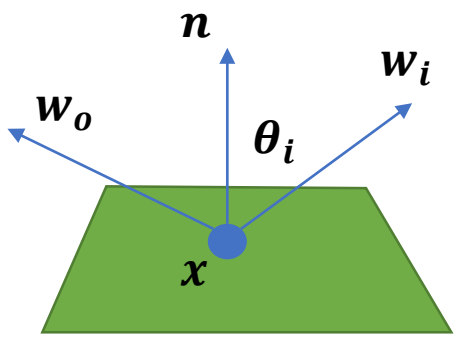
  - We can also write the equation with surface radiances only [Kajiya, 1986]

Rendering Equation



- $L_s(k_o) = \int_{all\ k_i} \rho(k_i, k_o) L_f(k_i) cos\theta_i d\sigma_i$
- $L_s(\boldsymbol{x}, w_o) = \int_{all\ \boldsymbol{x'}} \frac{\rho(\boldsymbol{w_i}, \boldsymbol{w_o}) L_s(\boldsymbol{x'}, \boldsymbol{x}-\boldsymbol{x'}) v(\boldsymbol{x}, \boldsymbol{x'}) cos\theta_i cos\theta'}{\|\boldsymbol{x}-\boldsymbol{x'}\|^2} dA$
- Discussion
  - Both equations are the same in theory.
  - When those are approximated via a sampling method (Monte Carlo ray tracing) , the sampling domain is different.
    - e.g., Generating a ray with k_i or toward x'





Monte Carlo Integration

- Given a function $f: S \rightarrow \mathbb{R}$, a random variable $x \sim p$
  - The expected value of $f(x)$
    - $E\big(f(x)\big) = \int_{x \in S} f(x)p(x)d\mu \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$
  - By substituting $g = fp$,
    - $E\big(f(x)\big) = \int_{x \in S} g(x)d\mu \approx \frac{1}{N}\sum_{i=1}^{N} \frac{g(x_i)}{p(x_i)}$
    - p should be positive for nonzero g
  - How can we apply this technique to the transport equation?
    - $L_s(k_o) = \int_{all\ x'} \frac{\rho(k_i,k_o)L_s(x',x-x')v(x,x')cos\theta_i cos\theta'}{||x-x'||^2} dA'$

Comparisons of Estimators

- Monte Carlo ray tracing (together with other image processing) is simply a pixel estimator, which produces an estimate $\hat{\mu}_c$ of the unknown μ_c at pixel *c*.
- Actual error: $\|\hat{\mu}_c - \mu_c\|^2$
  - Knowing the actual error is equivalent to knowing the unknown.
  - Used only when the unknown is available.
- MSE: $E[\|\hat{\mu}_c - \mu_c\|^2] = V(\hat{\mu}_c) + bias^2(\hat{\mu}_c)$
  - Widely used for optimizing the parameters in the estimator.
  - It requires estimation of the (unknown) MSE.

Comparisons of Estimators

- MSE: $E[\|\hat{\mu}_c - \mu_c\|^2] = V(\hat{\mu}_c) + bias^2(\hat{\mu}_c)$
  - Widely used for optimizing the parameters in the estimator.
  - It requires an estimation of the (unknown) MSE.
- Unbiased, $bias(\hat{\mu}_c) = 0$
- Biased, $bias(\hat{\mu}_c) \neq 0$
- Consistent
  - Zero MSE with an infinite sample count
- e.g.
  - path tracing: Unbiased and consistent
  - Progressive photon mapping: Biased but consistent

Path Tracing with Next Event Estimation

Problems in Naïve Path Tracing

- Issues: Hard to find a light path (very high variance)
  - i.e., the probability of hitting luminaries with small sizes is very low.

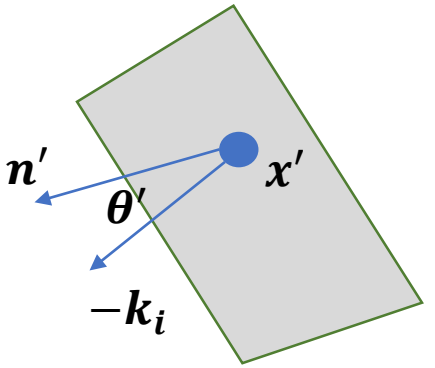
- A common practice for path tracing
  - Path tracing with direct lighting (sometimes referred to as path tracing with next event estimation)

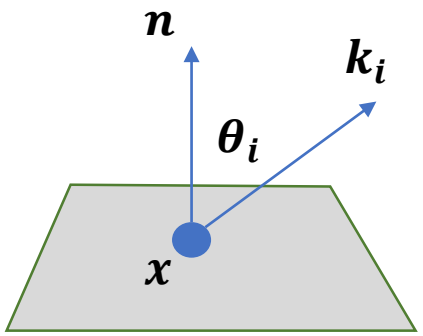
Path Tracing with Direct Lighting

- $\mathrm{L_s}(x, k_o) = \mathrm{L_s^D}(x, k_o) + \mathrm{L_s^I}(x, k_o)$
  - direct lighting $\mathrm{L_s^D}(x, k_o)$ + indirect lighting $\mathrm{L_s^I}(x, k_o)$

- $L_s^{\mathrm{D}}(k_o) = \int_{all\ x' \text{ in luminaries}} \frac{\rho(k_i,k_o)L_{\mathrm{e}}(x',x-x')v(x,x')cos\theta_i cos\theta'}{||x-x'||^2} dA'$
  - L_{e}: emitted radiance

- $L_s^{\mathrm{I}}(k_o) = \int_{all\ k_i} \rho(k_i, k_o) L_f^{\mathrm{R}}(k_i) cos\theta_i d\sigma_i$
  - $L_f^{\mathrm{R}}(k_i)$: field radiance (only reflected, i.e., not from luminaries)





Path Tracing with Direct Lighting

- $L_s^{\mathrm{D}}(x, k_o) = \int_{all\ x'} \frac{\rho(k_i,k_o) L_e(x', -k_i) v(x,x') cos\theta_i cos\theta_I}{||x-x'||^2} dA'$

- Sample a point x' on a luminaire with density function p ($x' \sim p$)

- $L_s^{\mathrm{D}}(x, k_o) \approx \frac{\rho(k_i,k_o) L_e(x', -k_i) v(x,x') cos\theta_i cos\theta_I}{p(x')\ \|\mathrm{x} - \mathrm{x}'\|^2}$

- Pick a uniform random point x' from the luminaire
  - $p = \frac{1}{A}$ (A is the area of the luminaire)
  - $L_s^{\mathrm{D}}(x, k_o) \approx \frac{\rho(k_i,k_o) L_e(x', -k_i) v(x,x') cos\theta_i cos\theta' A}{\|\mathrm{x} - \mathrm{x}'\|^2}$

Path Tracing with Direct Lighting

- Other details:
  - Only applied when the surface point is non-specular.
  - e.g., narrow BRDFs
    - Sampling a point on luminaries is not effective
  - Uses shadow rays for checking occlusions between the surface point and sampled light point Shadow rays
    - Usually faster than finding the first intersection
  - Samples a point on luminaries can be done via inverse transform sampling when the shapes of luminaries are simple (e.g., spheres, triangles)

Path Tracing with Direct Lighting

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

- Multiple light sources are given:
  - Generate one shadow ray per each light source, but this is not practical with many light sources
  - A common choice is to pick only a point x' and generate a shadow ray towards x'

Path Tracing with Direct Lighting

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

- Determining $p(x')$ requires the following:
  - $p(\mathrm{x}') = \mathrm{p}(\mathrm{l})\mathrm{p}(\mathrm{x}'|\mathrm{l})$
    - Probability of selecting a luminary l: $\mathrm{p}(\mathrm{l})$
    - Probability of sampling a point on the chosen light: $\mathrm{p}(\mathrm{x}'|\mathrm{l})$
  - How to select a light source l:
    - Uniform: the probability of selecting a light is equal.
    - Spatial: set the probability proportional to the light power (assume that all lights are visible against a point)
    - Visibility-aware selection: requires an estimation of visibility across surface points
    - Light clustering is a well-known approach for many lights, e.g., thousands of lights

Path Tracing with Direct Lighting

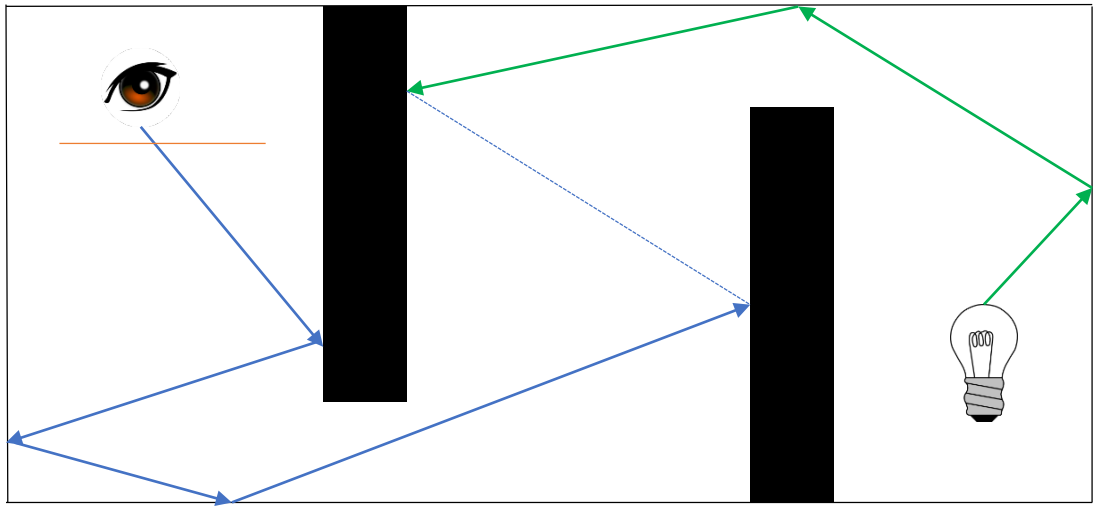
- Additional discussion:
  - Sampling a point on luminaries is often effective for Lambertian surfaces but not very effective on highly glossy surfaces
    - BRDF sampling for direct lighting can be better for such cases
    - How to combine Light sampling and BRDF sampling?
      - Multiple importance sampling (MIS) provides a solution; this will be covered later
  - When is direct lighting effective?
    - Intuitively, it is effective when lights are visible from most surfaces
    - A counter example

Image from https://benedikt-bitterli.me/resources/

Variants of Monte Carlo Rendering

Bidirectional Path Tracing

- [Veach and Guibas 95][Lafortune and Willems 93]
- Find light subpaths from the eye and light sources, and connect the subpaths



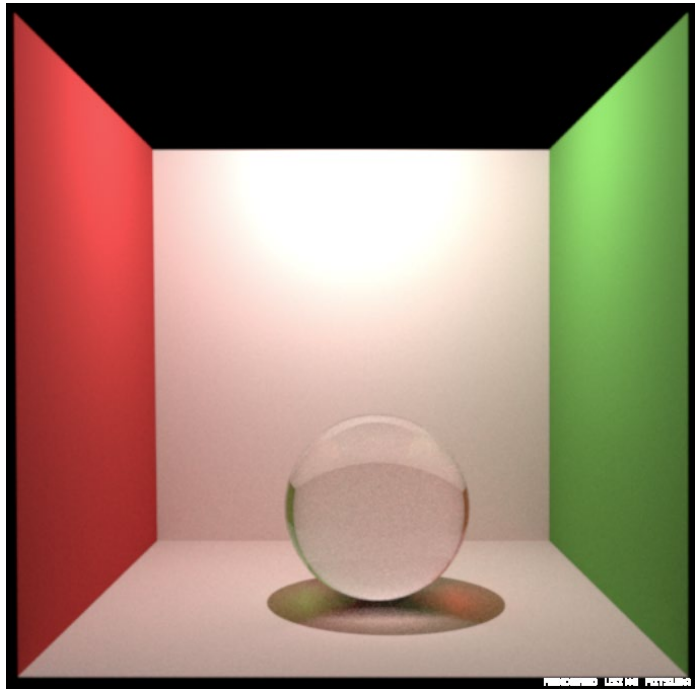
Unidirectional vs. bidirectional PT

Path tracing, 64 samples per pixel

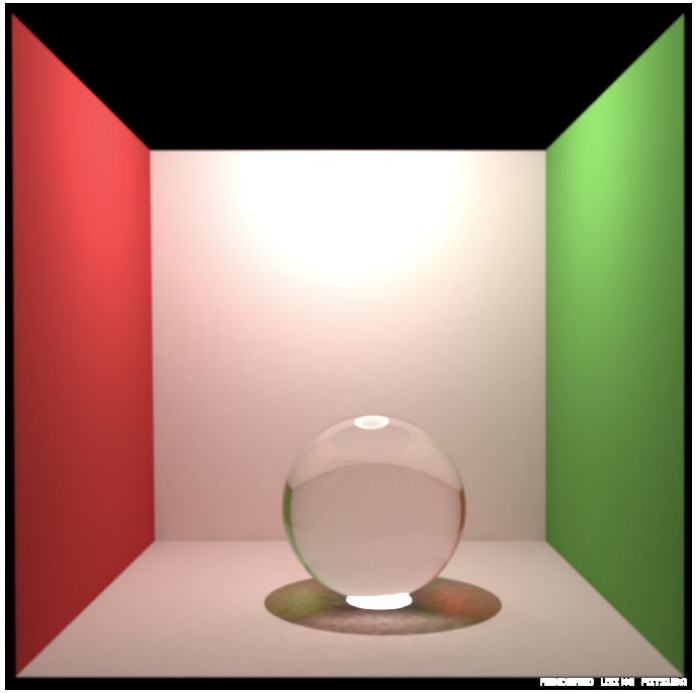
Bidirectional Path tracing, 64 samples per pixel

Photon Mapping

- [Jensen 1996]
- Motivation: it is hard to simulate some rendering effects (e.g., caustics) unless a large number of ray samples are used



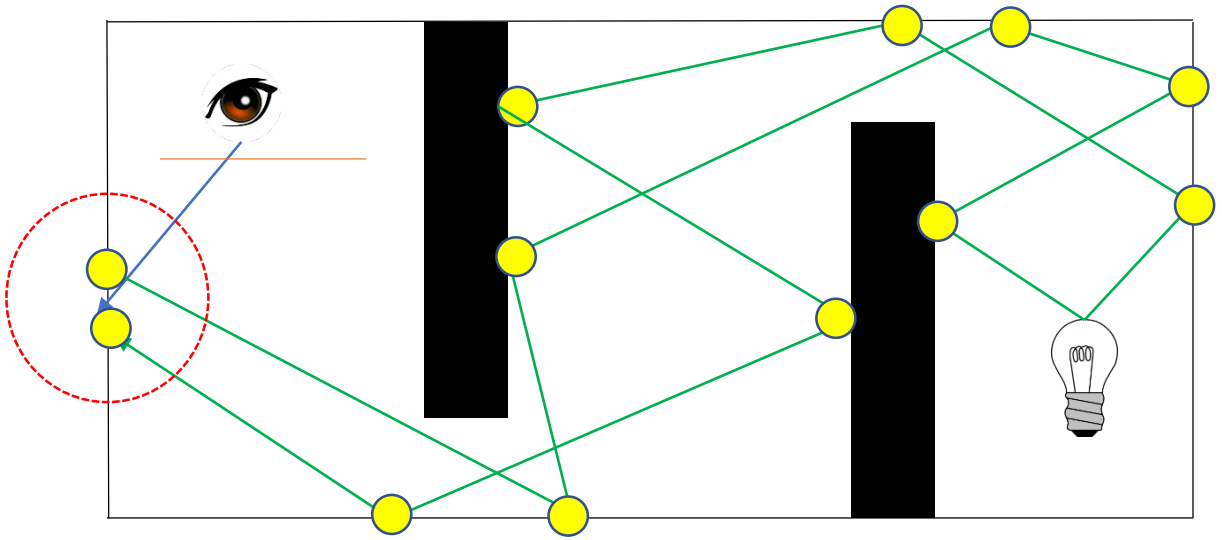
Path tracing



Photon mapping

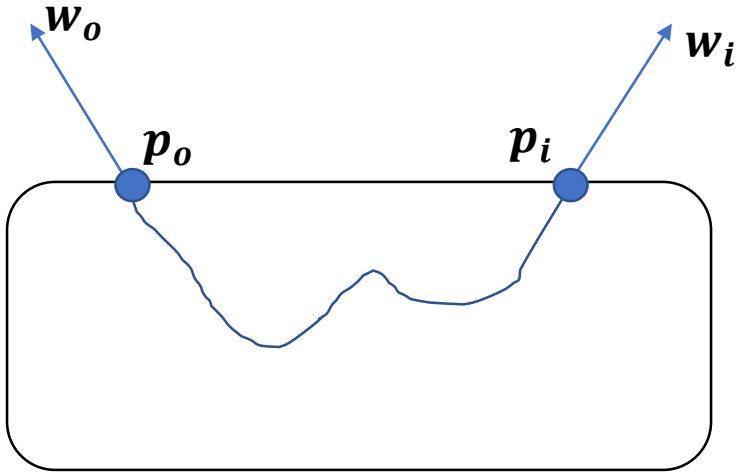
Photon Mapping

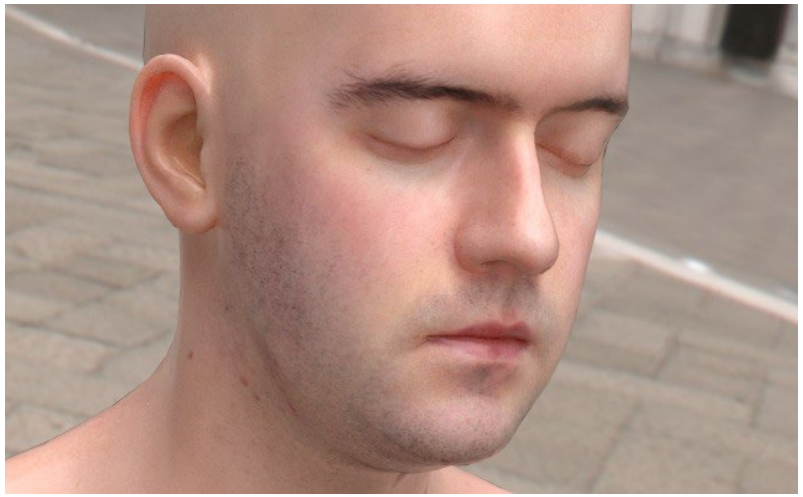
- Generate photons from each light and store them in kd-tree
- Trace primary rays and estimate a photon density at intersected points
  - Similar to the bidirectional path tracing, but the stored photons can be utilized for all eye rays (why?)



Advanced Topics

- Modeling complex materials
  - Bidirectional scattering-surface reflectance distribution function (BSSRDF)
- $\rho(\boldsymbol{w_i}, \boldsymbol{w_o})$: BRDF
- $\rho(\boldsymbol{p_i}, \boldsymbol{w_i}, \boldsymbol{p_o}, \boldsymbol{w_o})$: BSSRDF





Human head model with a BSSRDF

Model courtesy of Infinite Realities, Inc.
Image comes from pbrt.org

Advanced Topics

- Complex geometry (e.g., hair, fur) and scattering

[Jakob et al. 2010]

Model courtesy of Cem Yuksel
Image comes from pbrt.org

Advanced Topics

- Importance Sampling
  - $F_n = \frac{1}{n}\sum_{i=1}^{n}\frac{f(x_i)}{p(x_i)}$
  - Can we estimate an optimal pdf instead of the uniform distribution?

- Denoising

Path tracing,
16 samples per pixel

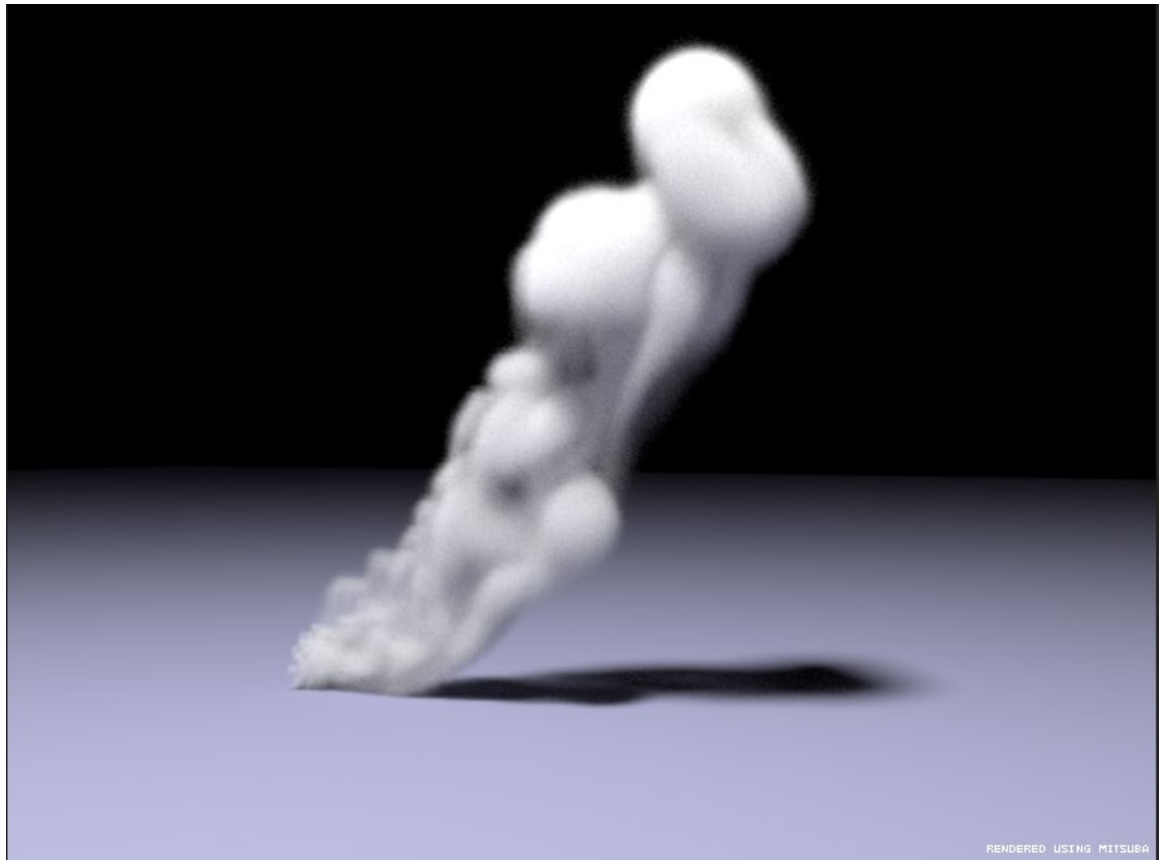
Denoised image,
16 samples per pixel

Path tracing
32K samples per pixel

Images from [Moon et al. 2016]

Advanced Topics

- Volume rendering



Graphics Topics

- Graphics Areas
  - Rendering
  - Modeling
  - Animation
  - User interaction
  - Augmented & virtual reality
  - Visualization
  - etc.

- Check out recent SIGGRAPH/SIGGRAPH Asia papers to know the active areas