GAMES Webinar: Rendering Tutorial 2

Monte Carlo Methods

Shuang Zhao

Assistant Professor

Computer Science Department

University of California, Irvine

Outline

1. Monte Carlo integration
   - A powerful numerical tool for estimating complex integrals
2. Rendering equation
   - The physical framework governing light transport
3. Path tracing
   - Basically, 1 + 2

4. Path integral formulation
5. Advanced methods

Monte Carlo Integration

GAMES Webinar: Rendering Tutorial 2

Why Monte Carlo?

- The “gold standard” approach to solve the RTE
- Advantages:
  - Provides estimates with quantifiable uncertainty
  - Adaptable to systems with complex geometries

Monte Carlo Integration

- A powerful framework for computing **integrals**

$$\int_{\Omega} f(\boldsymbol{x}) \, \mathrm{d}\mu(\boldsymbol{x}) = \, ?$$

- Numerical
- Nondeterministic (i.e., using randomness)
- Scalable to *high-dimensional* problems

Random Variables

- (Discrete) random variable X

- Possible outcomes: $x_1, x_2, \ldots, x_n$
  - with probability masses $p_1, p_2, \ldots, p_n$ such that

$$\sum_{i=1}^{n} p_i = 1$$

- E.g., "fair" coin
  - Outcomes: x_1 = "head", x_2 = "tail"
  - Probabilities: $p_1 = p_2 = ½$

Random Variables

- (Continuous) random variable *X*

- Possible outcomes: $[a, b] \subset \mathbb{R}$
  - with probability density function (PDF) *p*(*x*) satisfying

$$\int_a^b p(x)\,\mathrm{d}x = 1$$

  - Cumulative density function (CDF) *P*(*x*) given by

$$P(x) := \mathbb{P}[X \leq x] = \int_a^x p(y)\,\mathrm{d}y$$

Strong Law of Large Numbers

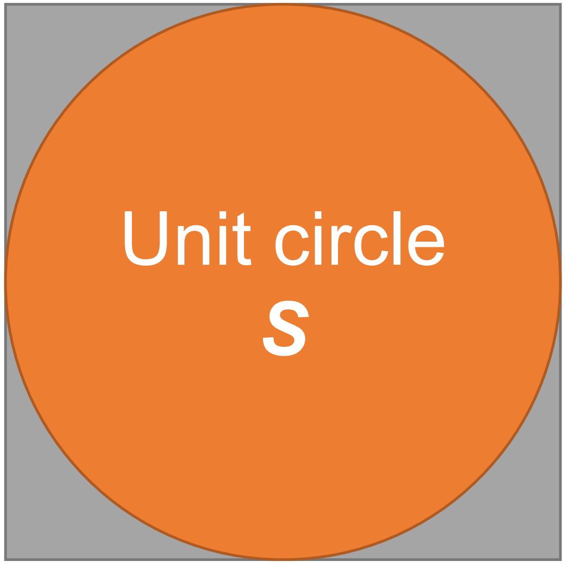
Let x_1, x_2, …, x_n be n independent observations (aka. **samples**) of X

$$\bar{x} := \frac{1}{n}\sum_{i=1}^{n} x_i \xrightarrow{n\to\infty} \mathbb{E}[X]$$

Sample mean

"Actual" mean

Example: Evaluating *π*



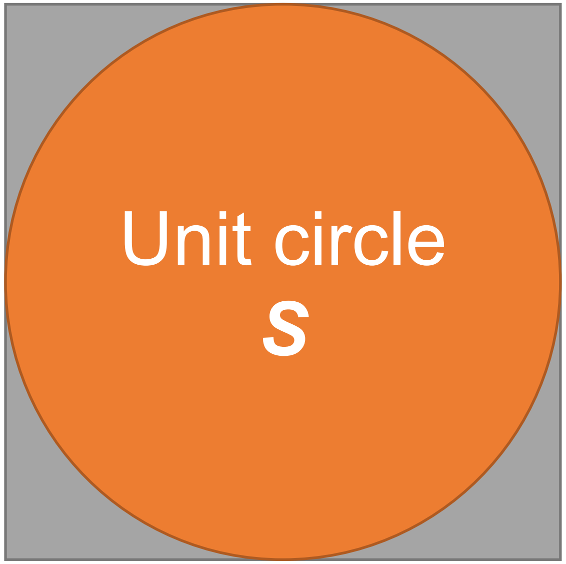
- Let $\boldsymbol{X}$ be a point uniformly distributed in the square
- $\mathbb{P}[\boldsymbol{X} \in S] = \frac{\pi}{4}$
- Let $f(\boldsymbol{X}) := \begin{cases} 4 & \boldsymbol{X} \in S \\ 0 & \boldsymbol{X} \notin S \end{cases}$, then

$$\begin{aligned} \mathbb{E}[f(\boldsymbol{X})] &= 4 \cdot \mathbb{P}[\boldsymbol{X} \in S] + 0 \cdot \mathbb{P}[\boldsymbol{X} \notin S] \\ &= 4 \cdot \frac{\pi}{4} = \pi \end{aligned}$$

Circle area = *π*

Square area = 4

Example: Evaluating π



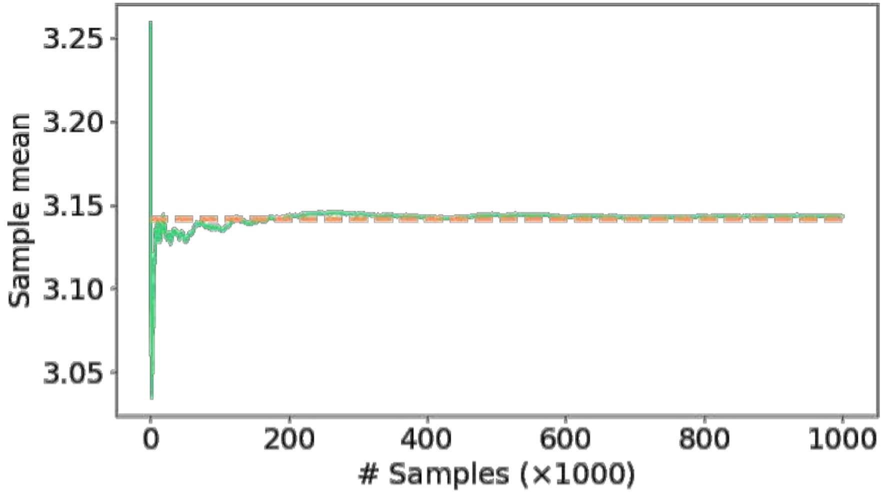
Circle area = π

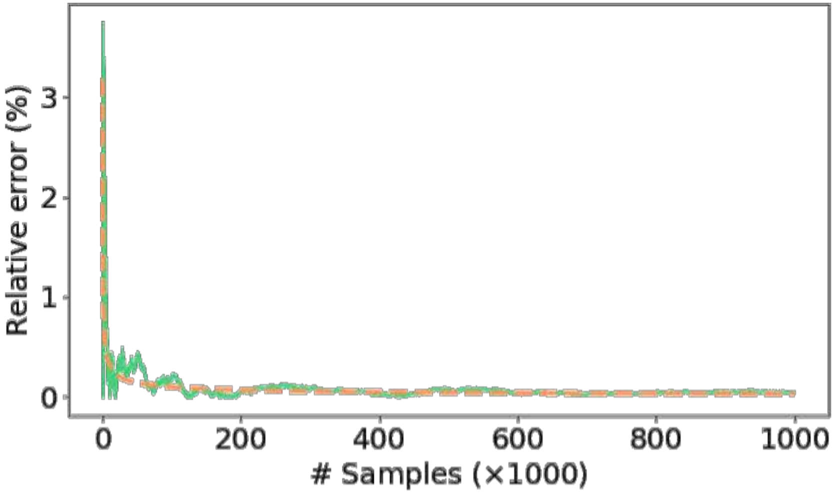
Square area = 4

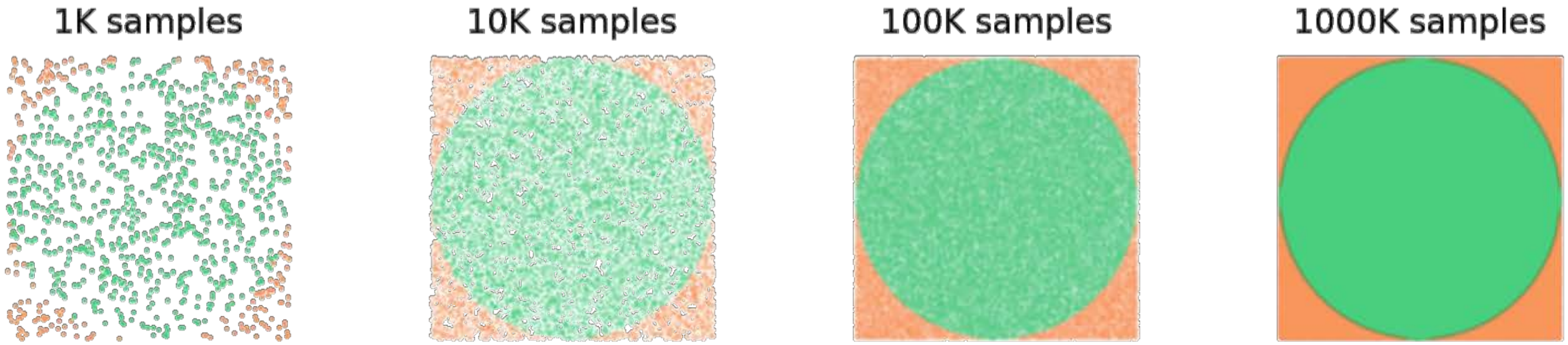
$$f(\boldsymbol{X}) := \begin{cases} 4 & \boldsymbol{X} \in S \\ 0 & \boldsymbol{X} \notin S \end{cases}$$

- Simple solution for computing π:
  - Generate n samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ independently
  - Compute $\frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{x}_i)$
- Live demo

Example: Evaluating π



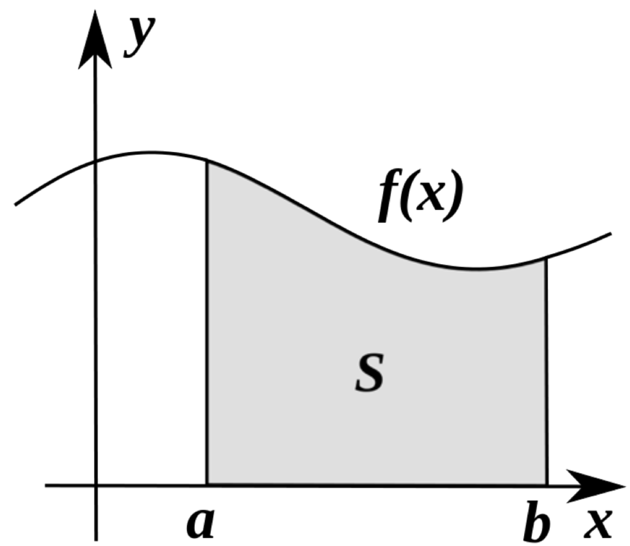




Integral

- f(x): one-dimensional scalar function

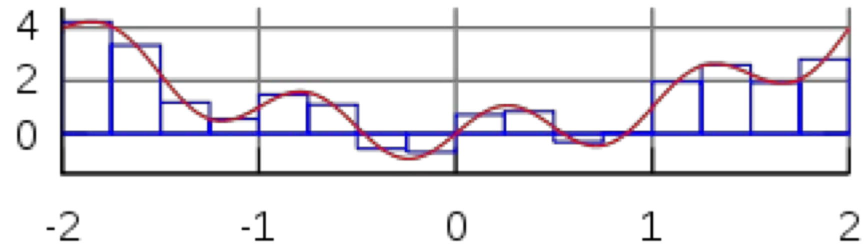
$$I = \int_a^b f(x)\,\mathrm{d}x$$



Deterministic Integration

- Quadrature rule:

$$I = \int_a^b f(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} \frac{b-a}{n} f(x_i)$$



- Scales poorly with high dimensionality:
  - Needs n^m bins for a m-dimensional problem
- We have a high-dimensional problem!

Monte Carlo Integration: Overview

- **Goal:** Estimating $I = \int_a^b f(x)\,\mathrm{d}x$
- **Idea:** Constructing random variable $\langle I \rangle$
  - Such that $\mathbb{E}[\langle I \rangle] = I$
  - $\langle I \rangle$ is called an **unbiased estimator** of I
- But how?

Monte Carlo Integration

- Let *p*() be any probability density function over [*a*, *b*] and *X* be a random variable with density *p*

- Let $\langle I \rangle := \dfrac{f(X)}{p(X)}$, then:

$$\mathbb{E}[g(X)] = \int_a^b g(x)\,p(x)\,\mathrm{d}x$$

$$\mathbb{E}[\langle I \rangle] = \mathbb{E}\left[\frac{f(X)}{p(X)}\right] = \int_a^b \frac{f(x)}{p(x)} p(x)\,\mathrm{d}x = I$$

- To estimate $\mathbb{E}[\langle I \rangle]$: strong law of large numbers

Monte Carlo Integration

- Goal: to estimate $I = \int_a^b f(x)\,\mathrm{d}x$
  - Pick a probability density function $p(x)$
  - Generate *n* independent samples:

$$x_1, x_2, \ldots, x_n \sim p$$

  - Evaluate $\hat{I}_j := \dfrac{f(x_j)}{p(x_j)}$ for *j* = 1, 2, …, *n*
  - Return sample mean: $\bar{I} := \dfrac{1}{n} \sum_{j=1}^{n} \hat{I}_j$

How to pick density function *p*()?

- **In theory**
  - (Almost) anything
- **In practice:**
  - Uniform distributions (almost) always work
    - As long as the domain is *bounded*
  - Choice of *p()* greatly affects the effectiveness (i.e., convergence rate) of the resulting estimator $\langle I \rangle$

Monte Carlo Integration "Hello, World!"

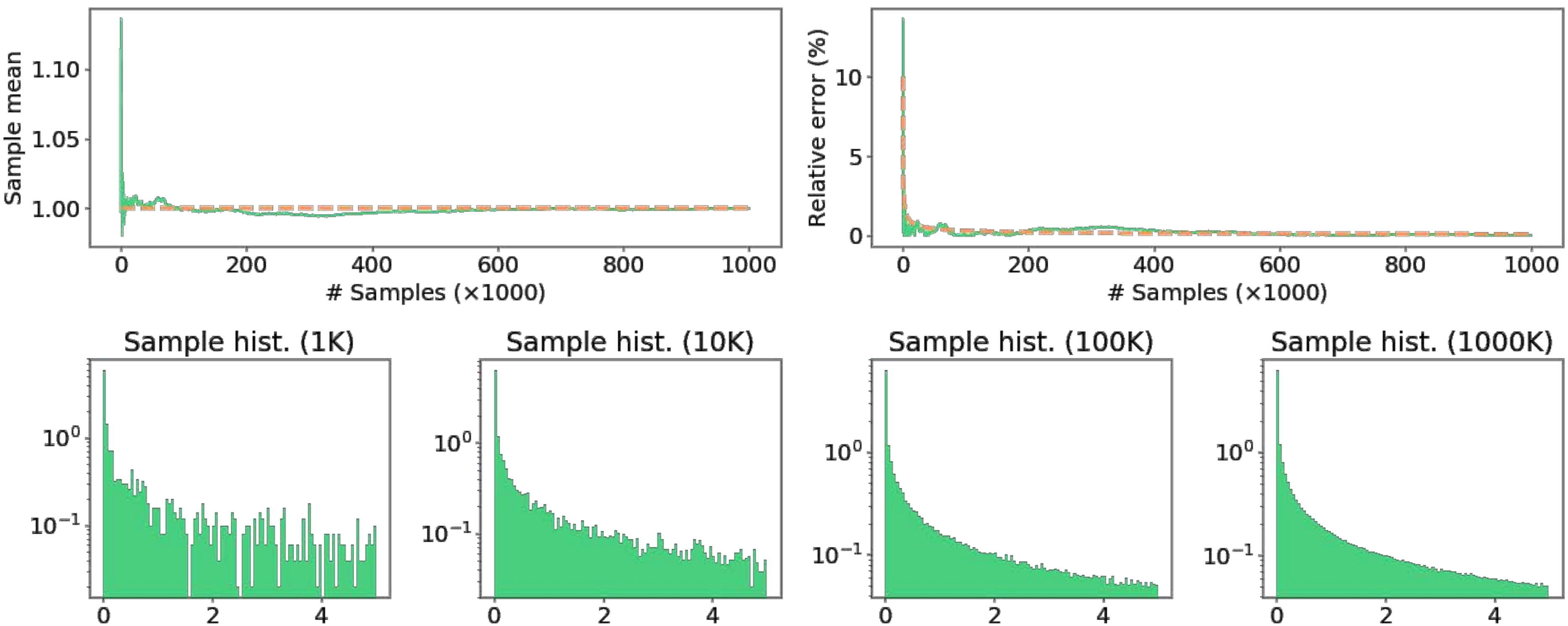
- Estimating $I = \int_0^1 5x^4 \,\mathrm{d}x \quad \left(= x^5\big|_0^1 = 1\right)$

- **Algorithm:**
  - Draw i.i.d $x_1, x_2, \ldots, x_n$ uniformly from [0, 1)
  - Return $\frac{1}{n}\sum_{j=1}^{n}\frac{f(x_j)}{p(x_j)} = \frac{1}{n}\sum_{j=1}^{n} 5x_j^4$

- Live demo

Monte Carlo Integration “Hello, World!”

- Estimating $I = \int_0^1 5x^4 \,\mathrm{d}x \quad \left(= x^5\big|_0^1 = 1\right)$



Multiple Importance Sampling

- To estimate

$$I = \int_{\Omega} f(\boldsymbol{x}) \, \mathrm{d}\mu(\boldsymbol{x})$$

- Assume there are *n* probability densities p_1, p_2, …, p_n to sample $\boldsymbol{x}$. Then,

$$\langle I \rangle_{\mathrm{MIS}} := \sum_{i=1}^{n} w_i(\boldsymbol{x}_i) \frac{f(\boldsymbol{x}_i)}{p_i(\boldsymbol{x}_i)} \quad \text{where } \boldsymbol{x}_i \sim p_i$$

  is an unbiased estimator of I as long as:
  - $\sum_{i=1}^{n} w_i(\boldsymbol{x}) = 1$ for all $\boldsymbol{x}$ with $f(\boldsymbol{x}) \neq 0$
  - $w_i(\boldsymbol{x}) = 0$ whenever $p_i(\boldsymbol{x}) = 0$

Weighting Functions

- **The balance heuristic**

$$w_i(\boldsymbol{x}) = \frac{p_i(\boldsymbol{x})}{\sum_{j=1}^{n} p_j(\boldsymbol{x})}$$

  - Then,

$$\langle I \rangle_{\text{balance}} := \sum_{i=1}^{n} w_i(\boldsymbol{x}_i) \frac{f(\boldsymbol{x}_i)}{p_i(\boldsymbol{x}_i)} = \sum_{i=1}^{n} \frac{f(\boldsymbol{x}_i)}{\sum_{j=1}^{n} p_j(\boldsymbol{x}_i)}$$

- How “good” is the new estimator?
  - As long as there exists a “good” estimator for I, $\langle I \rangle_{\text{balance}}$ will also be “good”

Example: MIS

$\mathbb{1}$ denotes the *indicator function*

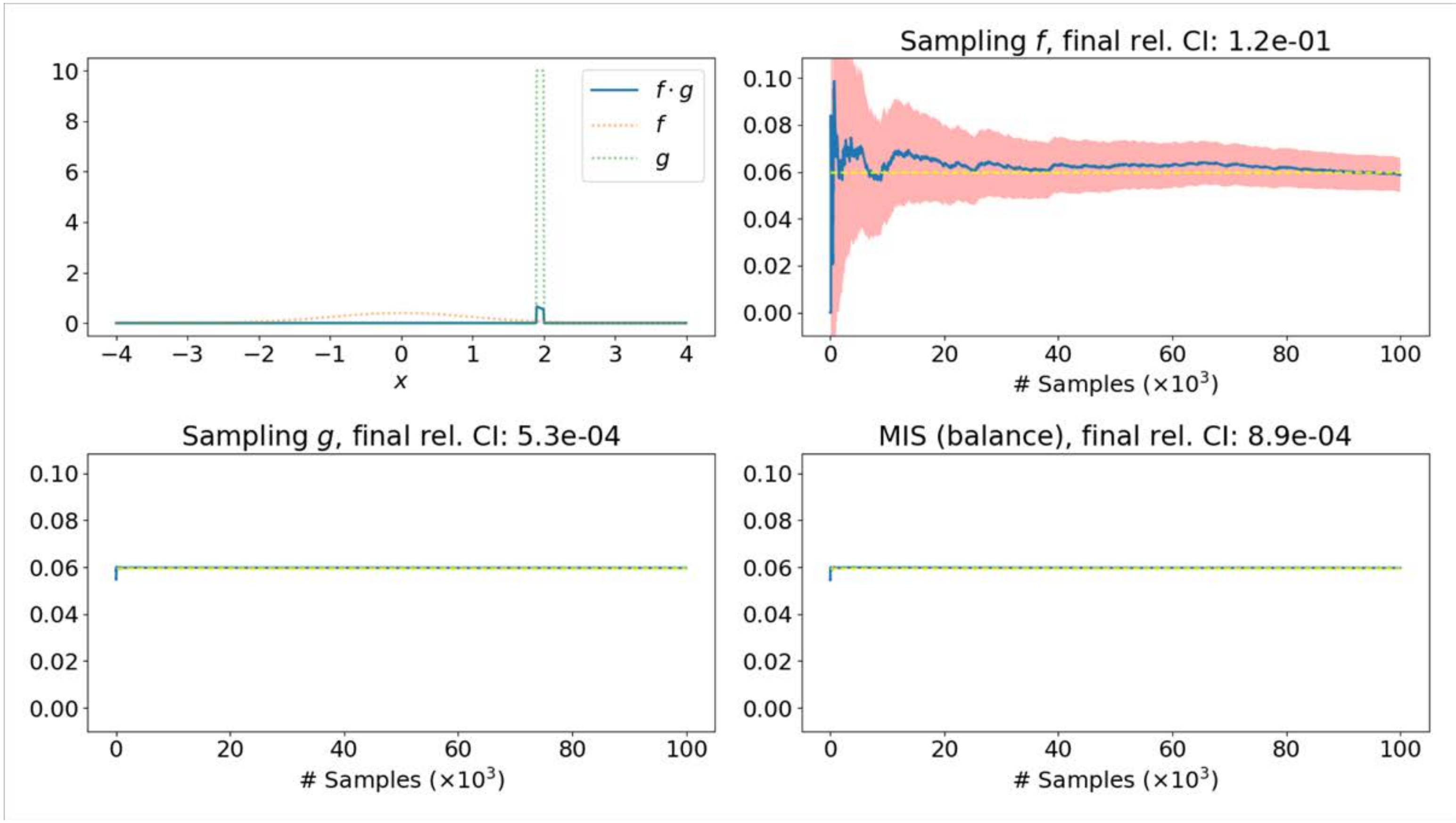
- Consider the problem of evaluating:

$$I = \int_{-\infty}^{\infty} \underbrace{\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)}_{=:f(x|\sigma)} \underbrace{\frac{\mathbb{1}\left[x \in [a,b)\right]}{b-a}}_{=:g(x|a,b)} \mathrm{d}x$$

with two probability densities $p_1 = f,\ p_2 = g$
  - *f*: normal distribution with mean 0 and variance σ^2
  - *g*: uniform distribution between *a* and *b*

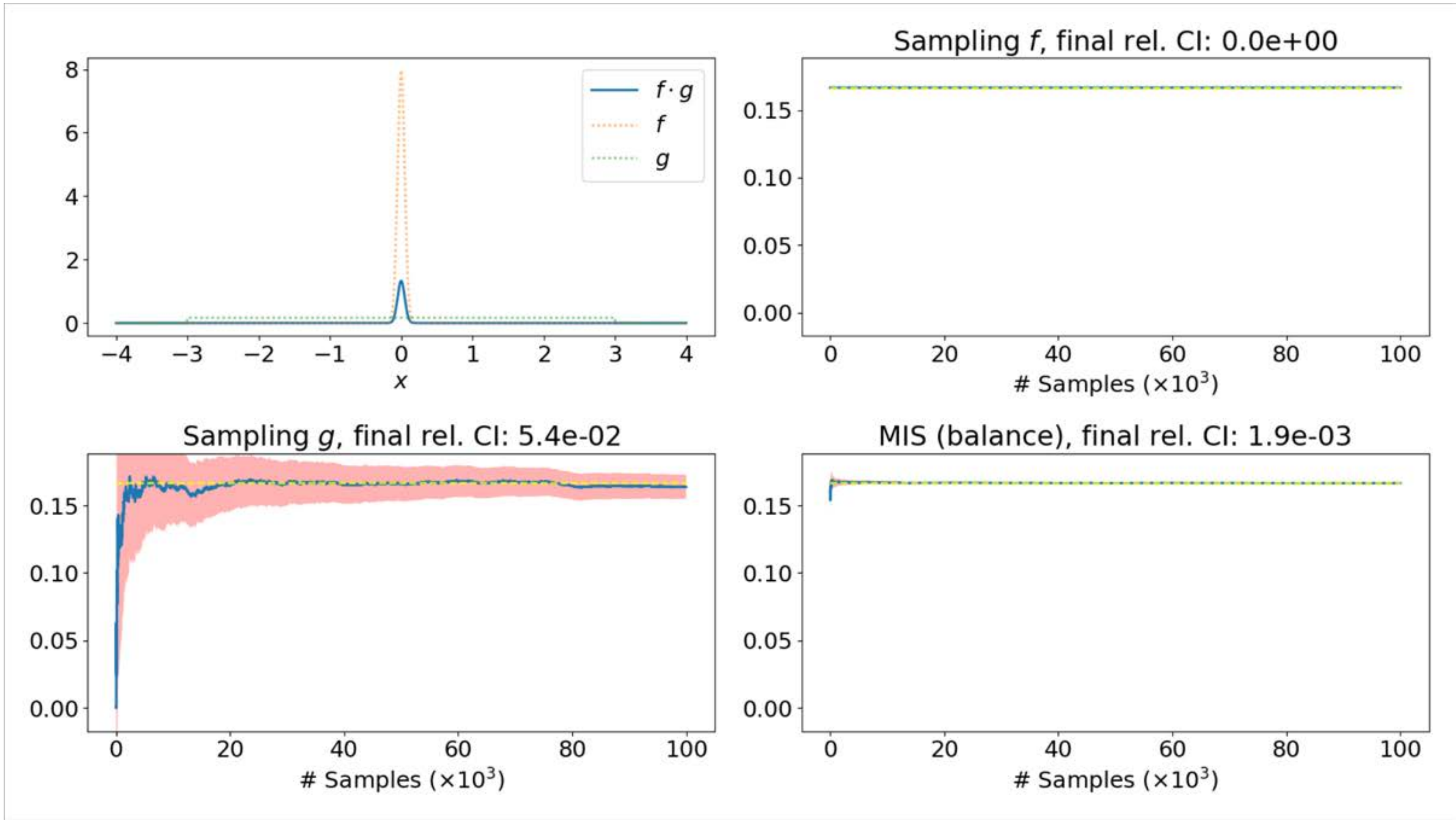
Example: MIS

- σ = 1; a = 1.9, b = 2.0



Example: MIS

- σ = 0.05; a = -3.0, b = 3.0

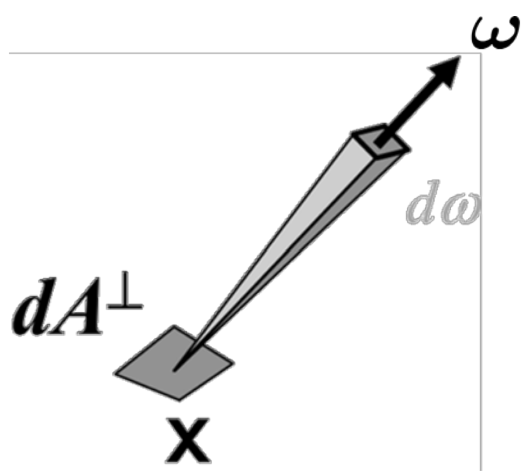


The Rendering Equation

GAMES Webinar: Rendering Tutorial 2

Radiance

- Radiant energy at ***x*** in direction ***ω***:
  - A 5D function $L(x, \omega)$: Power
    - per projected surface area
    - per solid angle



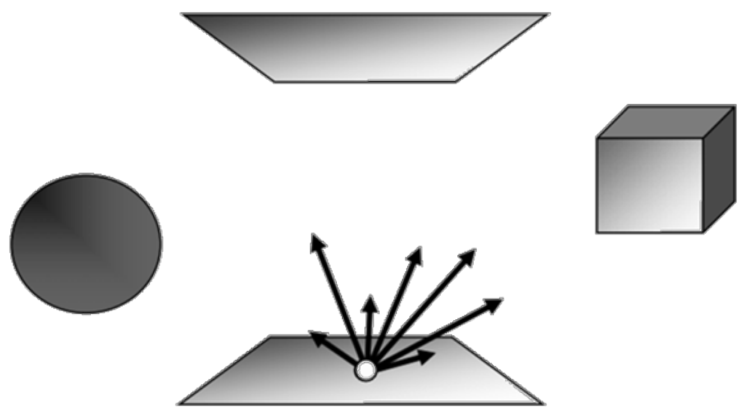
Light Transport

- Goal
  - Describe steady-state *radiance distribution* in virtual scenes

- Assumptions
  - Geometric optics
  - Achieves steady state instantaneously

Radiance at Equilibrium

- Radiance values at all points in the scene and in all directions expresses the equilibrium
  - 5D “Light-field”
- We only consider radiance on surfaces (4D)
  - Assuming no volumetric scattering or absorption

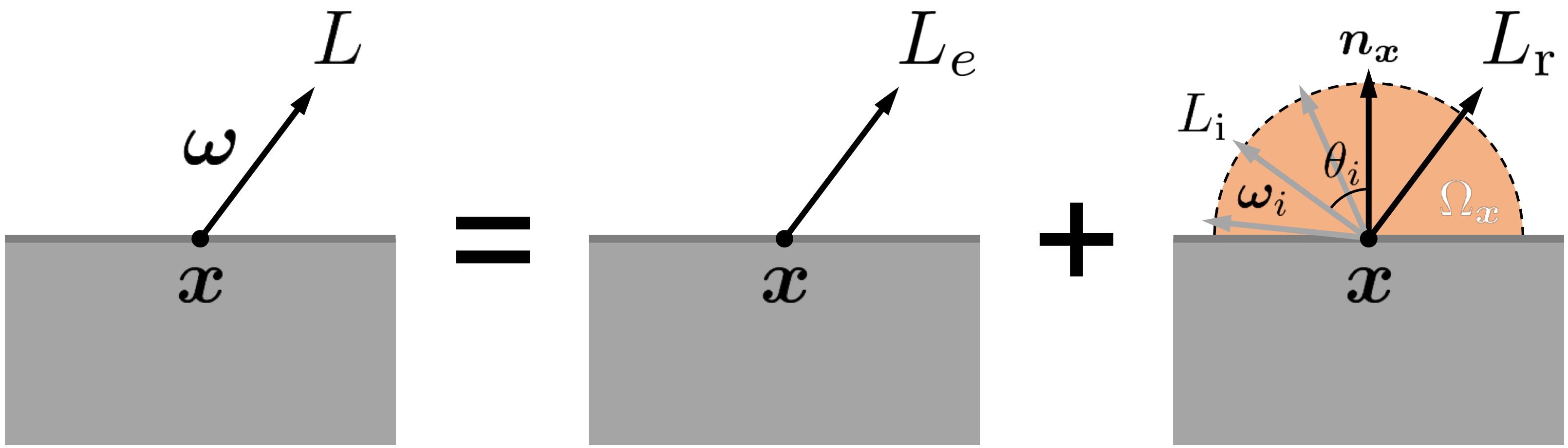


Rendering Equation (RE)

- RE describes the distribution of radiance at equilibrium

- RE involves:
  - Scene geometry
  - Light source info. **(Known)**
  - Surface reflectance info.
  - Radiance values at all surface points in all directions **(Unknown)**

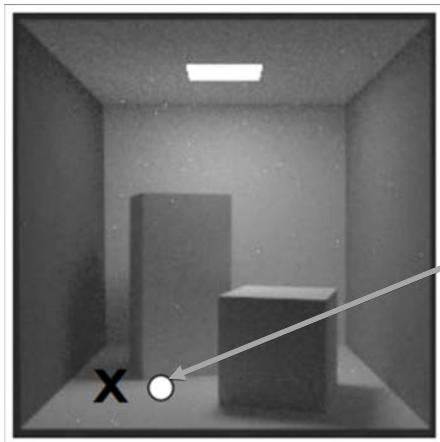
Rendering Equation (RE)



$$L(\boldsymbol{x}, \boldsymbol{\omega}) = L_e(\boldsymbol{x}, \boldsymbol{\omega}) + L_r(\boldsymbol{x}, \boldsymbol{\omega})$$

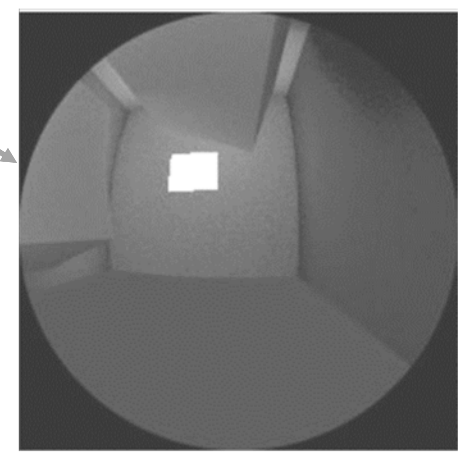
$$\int_{\Omega_{\boldsymbol{x}}} L_i(\boldsymbol{x}, \boldsymbol{\omega}_i)\, f_r(\boldsymbol{x}, \boldsymbol{\omega}_i \leftrightarrow \boldsymbol{\omega}) \underbrace{\langle \boldsymbol{n}_{\boldsymbol{x}}, \boldsymbol{\omega}_i \rangle}_{= \cos\theta_i} \mathrm{d}\boldsymbol{\omega}_i$$

Rendering Equation

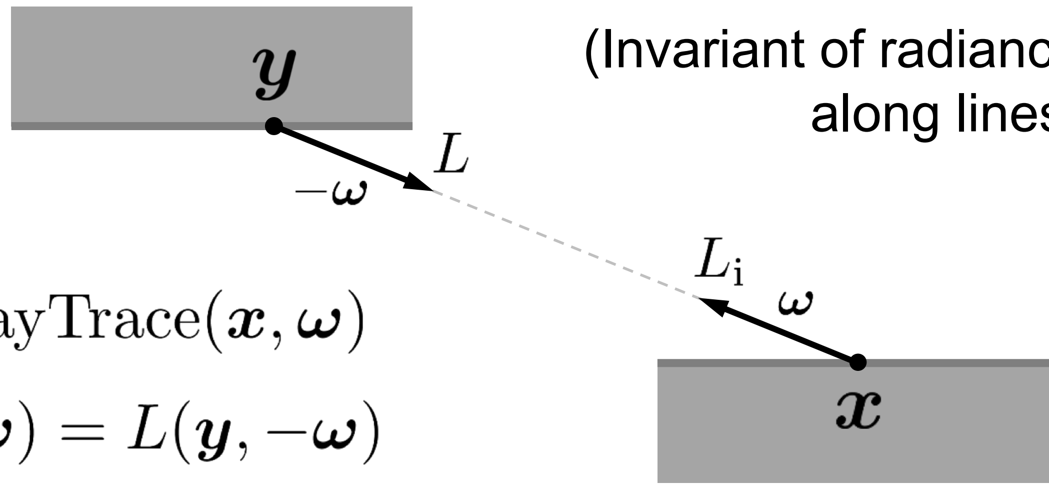


$$\boxed{L(\boldsymbol{x},\boldsymbol{\omega})} = L_e(\boldsymbol{x},\boldsymbol{\omega}) + \int_{\Omega_{\boldsymbol{x}}} \boxed{L_i(\boldsymbol{x},\boldsymbol{\omega}_i)}\, f_r(\boldsymbol{x},\boldsymbol{\omega}_i \leftrightarrow \boldsymbol{\omega})\langle \boldsymbol{n}_{\boldsymbol{x}}, \boldsymbol{\omega}_i\rangle \mathrm{d}\boldsymbol{\omega}_i$$

Incoming radiance



Rendering Equation



(Invariant of radiance along lines)

$$\boldsymbol{y} = \mathrm{RayTrace}(\boldsymbol{x}, \boldsymbol{\omega})$$

$$L_i(\boldsymbol{x}, \boldsymbol{\omega}) = L(\boldsymbol{y}, -\boldsymbol{\omega})$$

$$L(\boldsymbol{x}, \boldsymbol{\omega}) = L_e(\boldsymbol{x}, \boldsymbol{\omega}) + \int_{\Omega_{\boldsymbol{x}}} L(\boldsymbol{y}, -\boldsymbol{\omega}_i)\, f_r(\boldsymbol{x}, \boldsymbol{\omega}_i \leftrightarrow \boldsymbol{\omega}) \langle \boldsymbol{n}_{\boldsymbol{x}}, \boldsymbol{\omega}_i \rangle \mathrm{d}\boldsymbol{\omega}_i$$

Monte Carlo Path Tracing

GAMES Webinar: Rendering Tutorial 2

Path Tracing (Version 0)

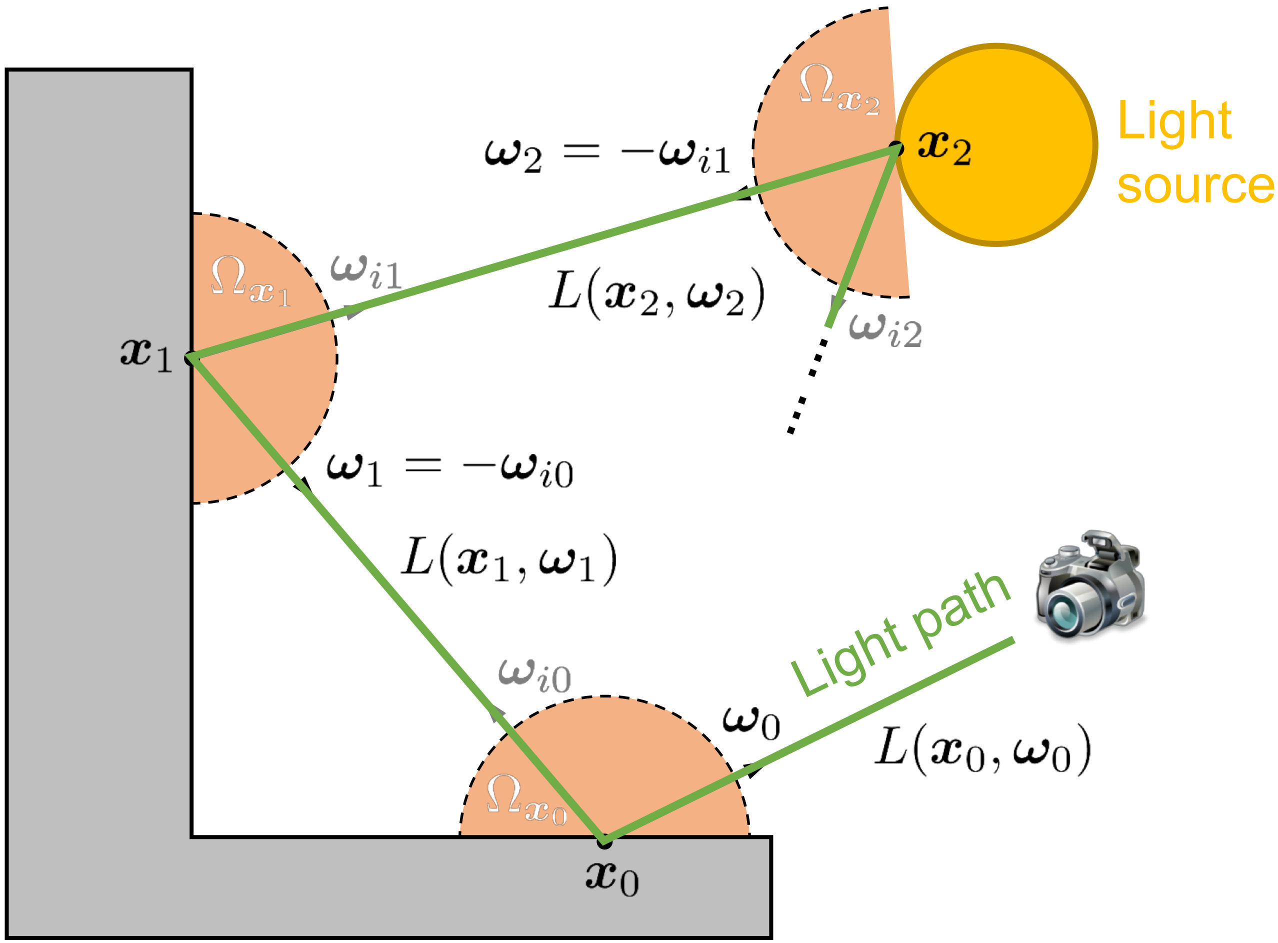
$$L(\boldsymbol{x}, \boldsymbol{\omega}) = L_e(\boldsymbol{x}, \boldsymbol{\omega}) + L_r(\boldsymbol{x}, \boldsymbol{\omega})$$

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}) = \int_{\Omega_{\boldsymbol{x}}} L(\boldsymbol{y}, -\boldsymbol{\omega}_i)\, f_r(\boldsymbol{x}, \boldsymbol{\omega}_i \leftrightarrow \boldsymbol{\omega})\, \langle \boldsymbol{n}_{\boldsymbol{x}}, \boldsymbol{\omega}_i \rangle \, \mathrm{d}\boldsymbol{\omega}_i$$

- Estimating L_r using MC integration:
  - Draw $\boldsymbol{\omega}_i$ uniformly at random
  - $p(\boldsymbol{\omega}_i) = 1/(2\pi)$

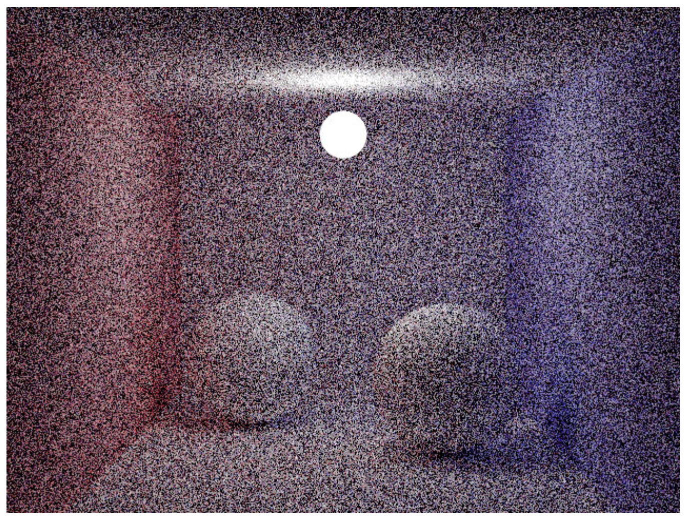
$$\begin{aligned}\langle L_r(\boldsymbol{x}, \boldsymbol{\omega}) \rangle &= \frac{L(\boldsymbol{y}, -\boldsymbol{\omega}_i)\, f_r(\boldsymbol{x}, \boldsymbol{\omega}_i \leftrightarrow \boldsymbol{\omega})\, \langle \boldsymbol{n}_{\boldsymbol{x}}, \boldsymbol{\omega}_i \rangle}{p(\boldsymbol{\omega}_i)} \\ &= 2\pi\, L(\boldsymbol{y}, -\boldsymbol{\omega}_i)\, f_r(\boldsymbol{x}, \boldsymbol{\omega}_i \leftrightarrow \boldsymbol{\omega})\, \langle \boldsymbol{n}_{\boldsymbol{x}}, \boldsymbol{\omega}_i \rangle\end{aligned}$$

Path Tracing (Version 0)



Challenge

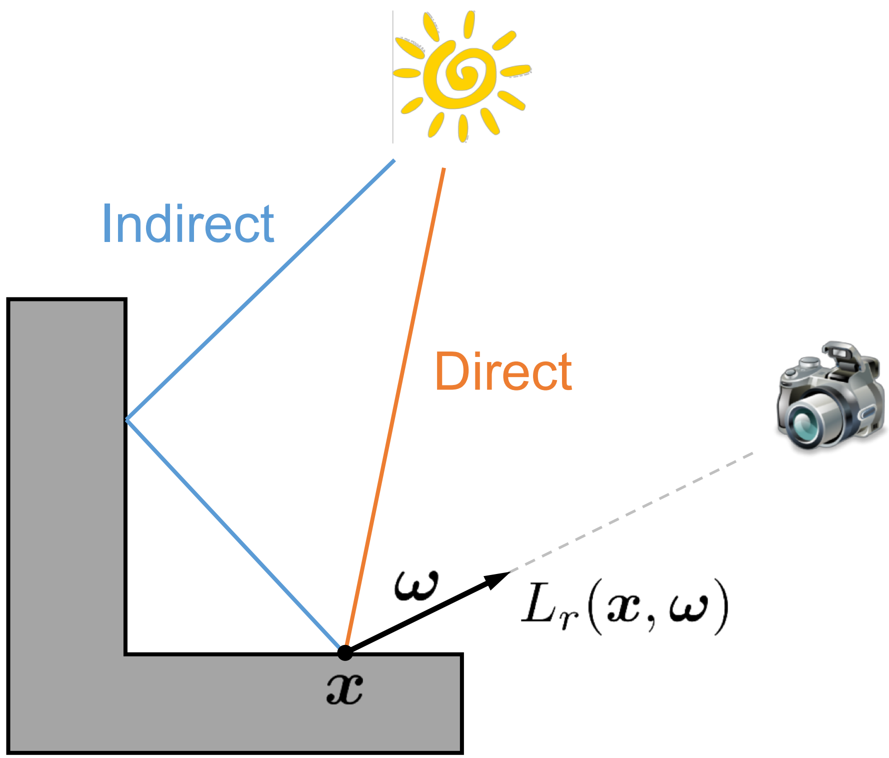
- Small light source leads to high noise
  - Because the probability for a “light path” to hit the light source is small



64 samples per pixel

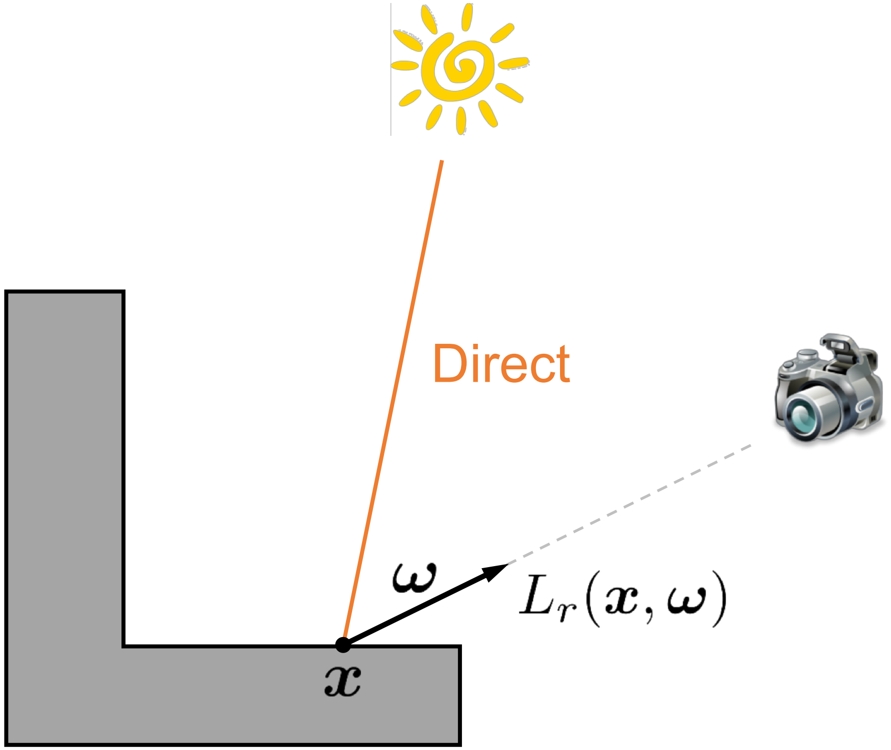
Idea: Separating Direct & Indirect

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}) = L_r^{\text{direct}}(\boldsymbol{x}, \boldsymbol{\omega}) + L_r^{\text{indirect}}(\boldsymbol{x}, \boldsymbol{\omega})$$



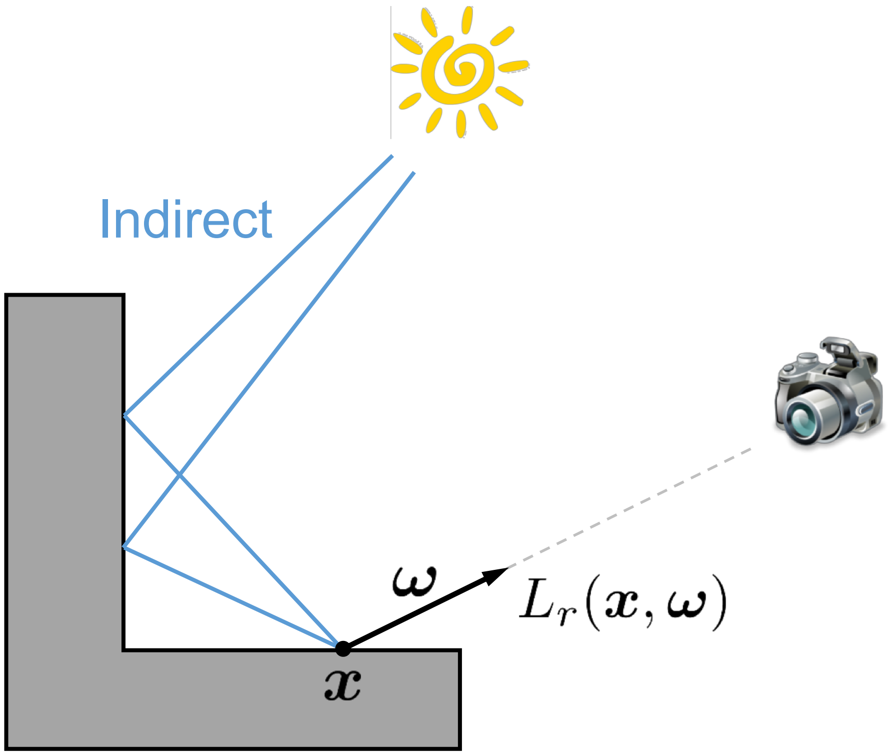
Direct Illumination

$$L_r^{\mathrm{direct}}(\boldsymbol{x}, \boldsymbol{\omega}) = \int_{\Omega_{\boldsymbol{x}}} L_e(\boldsymbol{y}, -\boldsymbol{\omega}_i)\, f_r(\boldsymbol{x}, \boldsymbol{\omega}_i \leftrightarrow \boldsymbol{\omega})\, \langle \boldsymbol{n}_{\boldsymbol{x}}, \boldsymbol{\omega}_i \rangle \, \mathrm{d}\boldsymbol{\omega}_i$$



Indirect Illumination

$$L_r^{\text{indirect}}(\boldsymbol{x}, \boldsymbol{\omega}) = \int_{\Omega_{\boldsymbol{x}}} L_r(\boldsymbol{y}, -\boldsymbol{\omega}_i)\, f_r(\boldsymbol{x}, \boldsymbol{\omega}_i \leftrightarrow \boldsymbol{\omega})\, \langle \boldsymbol{n}_{\boldsymbol{x}}, \boldsymbol{\omega}_i \rangle\, \mathrm{d}\boldsymbol{\omega}_i$$



Summary: Direct + Indirect

$L = L_e + L_r$

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}) = \int_{\Omega_{\boldsymbol{x}}} L(\boldsymbol{y}, -\boldsymbol{\omega}_i) \, f_r(\boldsymbol{x}, \boldsymbol{\omega}_i \leftrightarrow \boldsymbol{\omega}) \, \langle \boldsymbol{n}_{\boldsymbol{x}}, \boldsymbol{\omega}_i \rangle \, \mathrm{d}\boldsymbol{\omega}_i$$

$$L_r^{\text{direct}}(\boldsymbol{x}, \boldsymbol{\omega}) = \int_{\Omega_{\boldsymbol{x}}} L_e(\boldsymbol{y}, -\boldsymbol{\omega}_i) \, f_r(\boldsymbol{x}, \boldsymbol{\omega}_i \leftrightarrow \boldsymbol{\omega}) \, \langle \boldsymbol{n}_{\boldsymbol{x}}, \boldsymbol{\omega}_i \rangle \, \mathrm{d}\boldsymbol{\omega}_i$$

Non-recursive

$$L_r^{\text{indirect}}(\boldsymbol{x}, \boldsymbol{\omega}) = \int_{\Omega_{\boldsymbol{x}}} L_r(\boldsymbol{y}, -\boldsymbol{\omega}_i) \, f_r(\boldsymbol{x}, \boldsymbol{\omega}_i \leftrightarrow \boldsymbol{\omega}) \, \langle \boldsymbol{n}_{\boldsymbol{x}}, \boldsymbol{\omega}_i \rangle \, \mathrm{d}\boldsymbol{\omega}_i$$

Recursion

- This idea is usually called "*next-event estimation*"

Estimating Direct Illumination

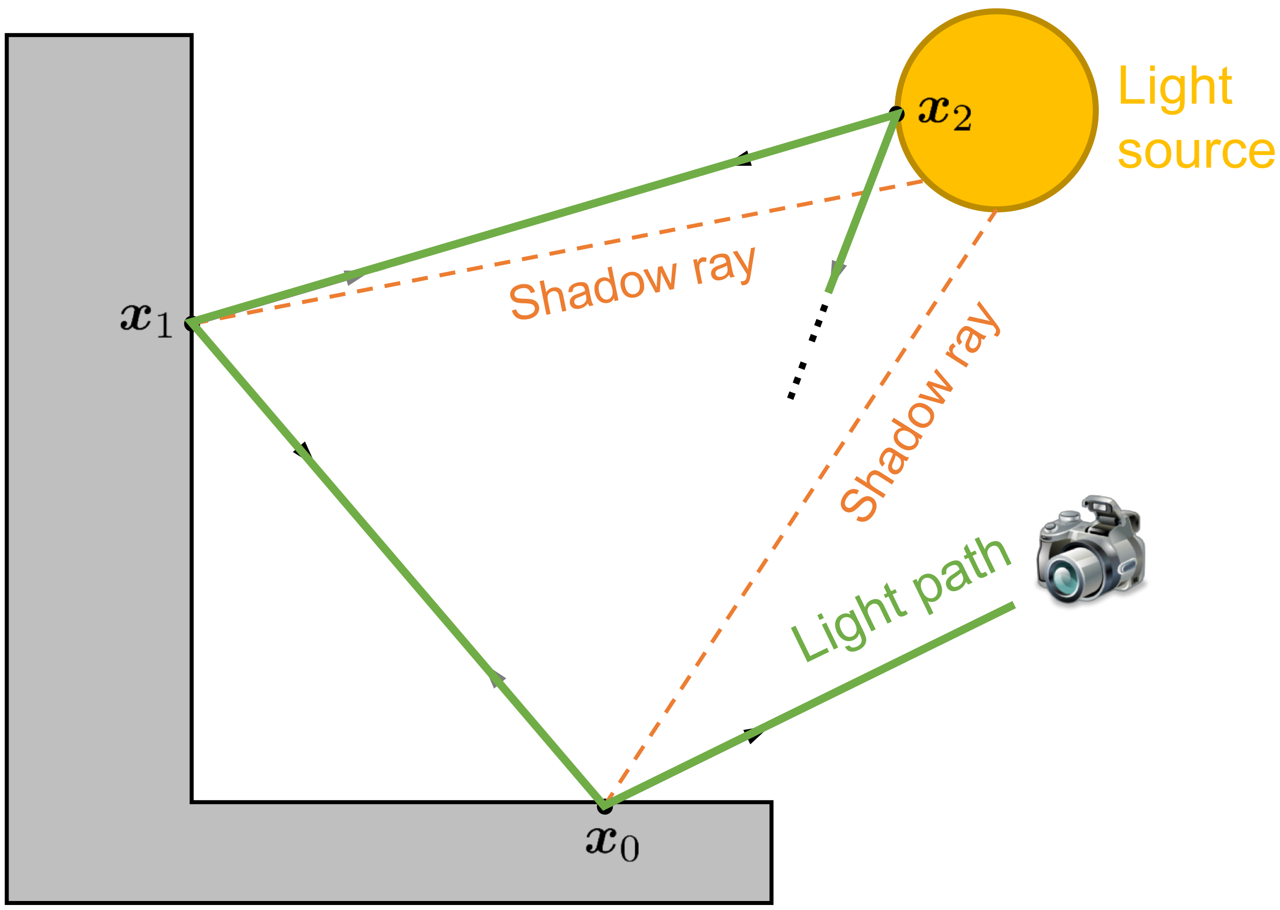
- Idea: sampling the light source

- Solid angle integral → area integral

$$
\begin{aligned}
L_r^{\text{direct}}(\boldsymbol{x}, \boldsymbol{\omega}) &= \int_{\Omega_{\boldsymbol{x}}} L_e(\boldsymbol{y}, -\boldsymbol{\omega}_i)\, f_r(\boldsymbol{x}, \boldsymbol{\omega}_i \leftrightarrow \boldsymbol{\omega})\, \langle \boldsymbol{n}_{\boldsymbol{x}}, \boldsymbol{\omega}_i \rangle \, \mathrm{d}\boldsymbol{\omega}_i \\
&= \int_{A_e} L_e(\boldsymbol{y}, -\boldsymbol{\omega}_i)\, f_r(\boldsymbol{x}, \boldsymbol{\omega}_i \leftrightarrow \boldsymbol{\omega})\, V(\boldsymbol{x}, \boldsymbol{y}) \, \frac{\langle \boldsymbol{n}_{\boldsymbol{x}}, \boldsymbol{\omega}_i \rangle \langle \boldsymbol{n}_{\boldsymbol{y}}, -\boldsymbol{\omega}_i \rangle}{\|\boldsymbol{x} - \boldsymbol{y}\|_2^2} \, \mathrm{d}\boldsymbol{y}
\end{aligned}
$$

Change of measure

Path Tracing (with NEE)



Path Integral Formulation

GAMES Webinar: Rendering Tutorial 2

Path Integral Formulation (Preview)

- **Goal:** rewriting the measurement equation

$$I = \int_{\mathcal{M}} \int_{\Omega_{\boldsymbol{x}}} W_e(\boldsymbol{x}, \boldsymbol{\omega})\, L_i(\boldsymbol{x}, \boldsymbol{\omega})\, \langle \boldsymbol{n}_{\boldsymbol{x}}, \boldsymbol{\omega} \rangle \, \mathrm{d}\boldsymbol{\omega} \, \mathrm{d}\boldsymbol{x},$$

  as an integral of the form

$$I = \int_{\Omega} f(\bar{x}) \, \mathrm{d}\mu(\bar{x}),$$

  where $\bar{x}$ denotes individual light paths for the form $(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k)$

Benefits

- Express the measurement *I* as an integral instead of an integral equation
- Provide a unified framework for deriving advanced estimators of *I*

Recap: Change of Measure

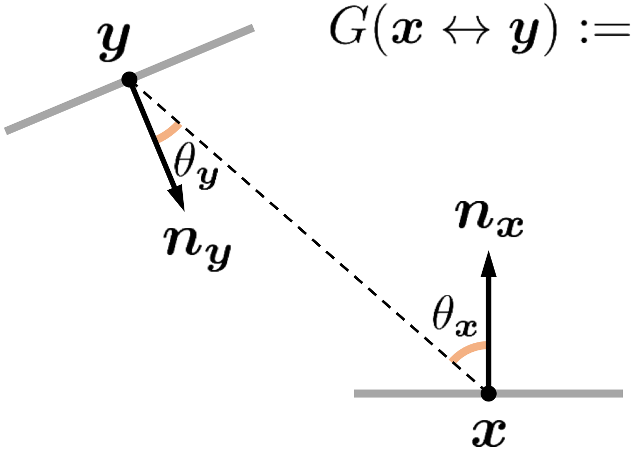
- From *solid angle* to *area*:

$$\text{Solid angle:} \quad L_r^{\text{direct}}(\boldsymbol{x}, \boldsymbol{\omega}) = \int_{\Omega_{\boldsymbol{x}}} L_i^{\text{direct}}(\boldsymbol{x}, \boldsymbol{\omega}_i)\, f_r(\boldsymbol{x}, \boldsymbol{\omega}_i \to \boldsymbol{\omega})\, \langle \boldsymbol{n}_{\boldsymbol{x}}, \boldsymbol{\omega}_i \rangle \, \mathrm{d}\boldsymbol{\omega}_i$$

$$\boxed{\text{Area:} \quad L_r^{\text{direct}}(\boldsymbol{x}, \boldsymbol{\omega}) = \int_{A} L_i^{\text{direct}}(\boldsymbol{x}, \boldsymbol{\omega}_i)\, f_r(\boldsymbol{x}, \boldsymbol{\omega}_i \to \boldsymbol{\omega})\, G(\boldsymbol{x} \leftrightarrow \boldsymbol{y}) \, \mathrm{d}\boldsymbol{y}}$$

We now apply this idea to the *RE* as well as the *measurement equation*

$$G(\boldsymbol{x} \leftrightarrow \boldsymbol{y}) := V(\boldsymbol{x}, \boldsymbol{y}) \frac{\cos\theta_{\boldsymbol{x}} \cos\theta_{\boldsymbol{y}}}{\|\boldsymbol{x} - \boldsymbol{y}\|^2}$$



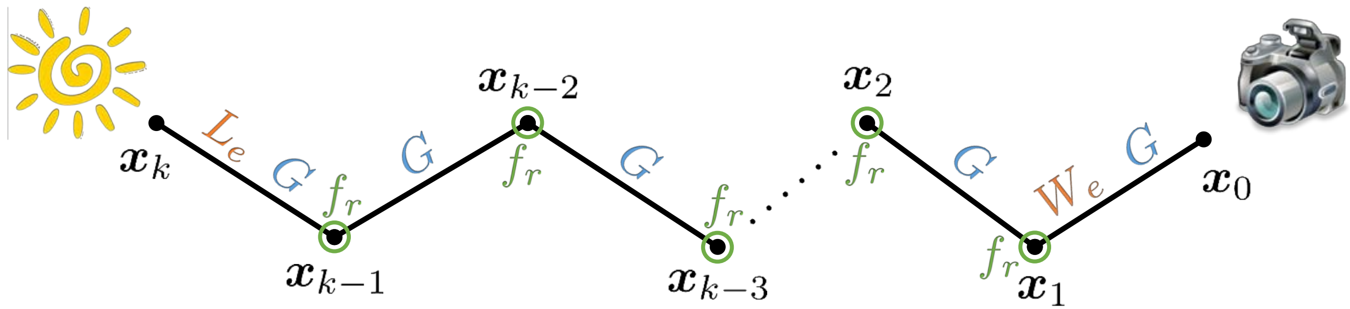
Path Integral Formulation of the RE

- By repeatedly expanding *L*, we get:

$$I = \int_\Omega f(\bar{x}) \, \mathrm{d}\mu(\bar{x}),$$

where for any $\bar{x} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k)$:

$$f(\bar{x}) := L_e(\boldsymbol{x}_k \to \boldsymbol{x}_{k-1}) \left[\prod_{j=0}^{k-1} G(\boldsymbol{x}_{j+1} \leftrightarrow \boldsymbol{x}_j) \right] \left[\prod_{j=1}^{k-1} f_r(\boldsymbol{x}_{j+1} \to \boldsymbol{x}_j \to \boldsymbol{x}_{j-1}) \right] W_e(\boldsymbol{x}_1 \to \boldsymbol{x}_0)$$



Applying the Path Integral Formulation

- The path integral

$$I = \int_\Omega f(\bar{x})\,\mathrm{d}\mu(\bar{x})$$

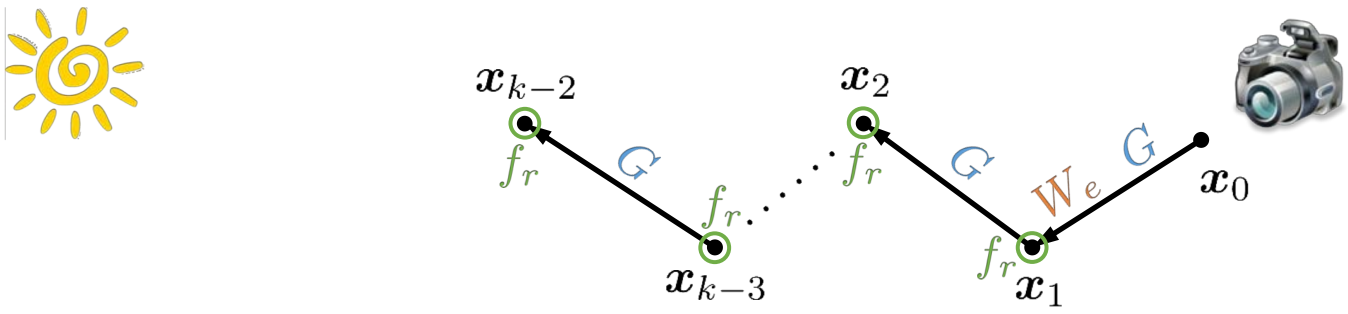
can be estimated via Monte Carlo integration framework:

$$\langle I \rangle = \frac{f(\bar{x})}{p(\bar{x})}$$

  - What density *p* to use?
  - How to draw a sample path $\bar{x}$ from *p*?

Recap: Local Path Sampling

- Path tracing



- Without next-event estimation
  - One transport path at a time
- With next-event estimation
  - Multiple transport paths at a time

Advanced Methods

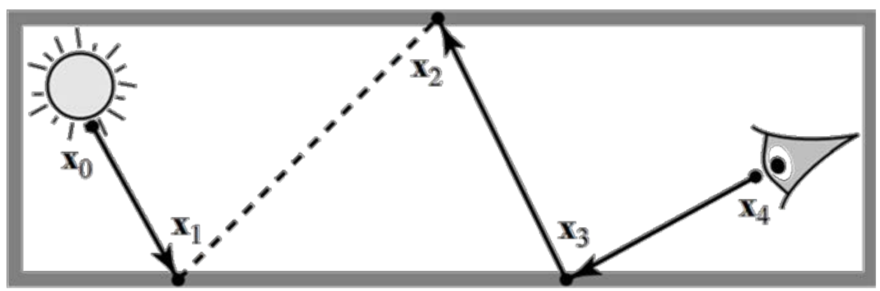
GAMES Webinar: Rendering Tutorial 2

Advanced Methods

- Bidirectional path tracing (BDPT)
- Metropolis light transport (MLT)

Bidirectional Path Tracing

- Build light transport paths by connecting two sub-paths starting from the light source and the sensor, respectively

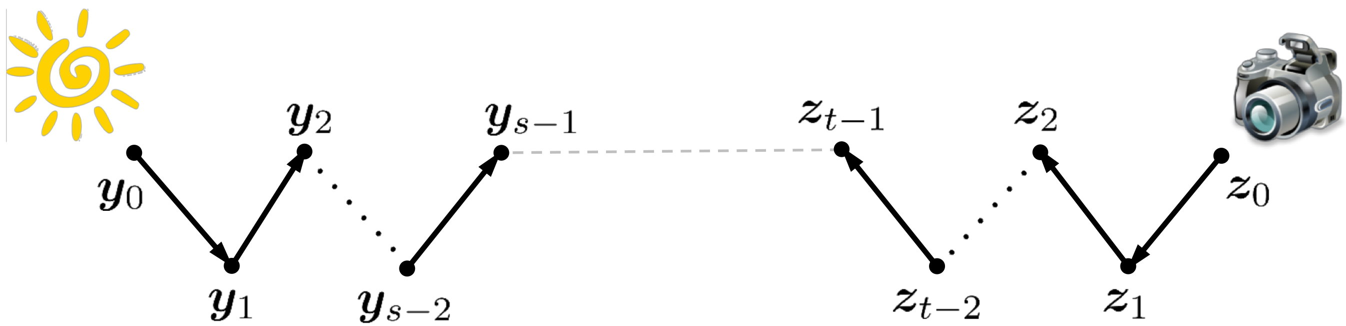


[Veach 1997]

- Use multiple importance sampling (MIS) to properly weight each path

Building Transport Paths

- For any $s, t \geq 0$, create light sub-path $(\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{s-1})$ and sensor sub-path $(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_{t-1})$ (using local path sampling)

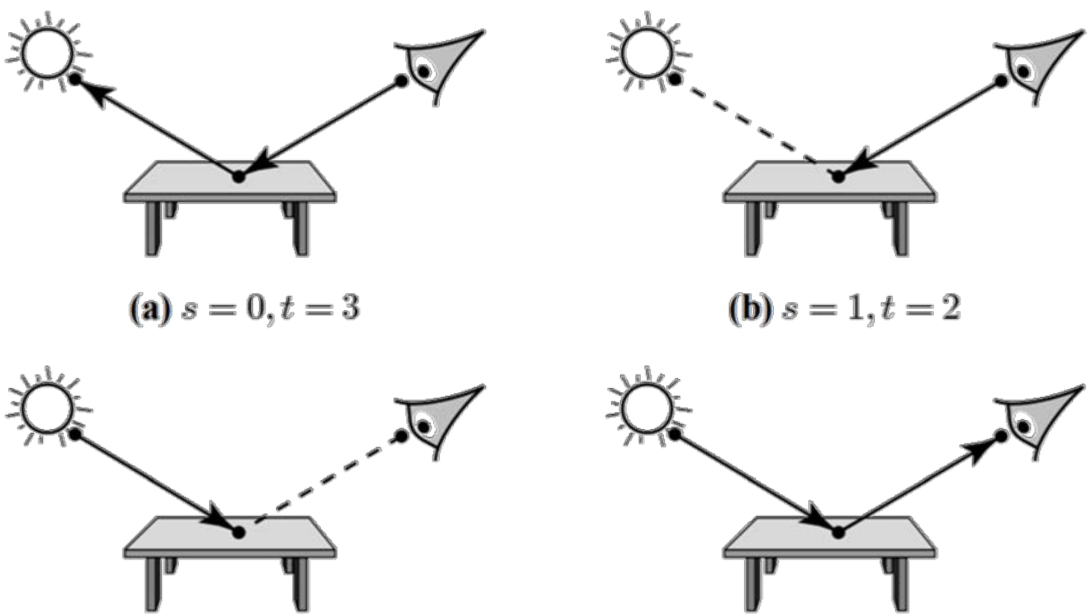


- The full path $\bar{x}_{s,t}$ is obtained by *concatenating* these two pieces:

$$\bar{x}_{s,t} := (\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{t-1}, \boldsymbol{y}_{s-1}, \ldots, \boldsymbol{y}_1, \boldsymbol{y}_0)$$

Building Transport Paths

- **Remark:** there is more than one sampling technique for each path length
  - (k + 1) techniques for paths with k vertices



(a) $s = 0, t = 3$ (b) $s = 1, t = 2$

(c) $s = 2, t = 1$ (d) $s = 3, t = 0$

[Veach 1997]

Building Transport Paths

- For each *s* and *t*, the construction of $\bar{x}_{s,t}$ gives a probability density $p_{s,t}(\bar{x}_{s,t})$
- Similar to the unidirectional case, $p_{s,t}(\bar{x}_{s,t})$ equals the product of densities of sampling both sub-paths

$$p_{s,t}(\bar{x}_{s,t}) = p((\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{s-1}))\; p((\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{t-1}))$$

Building Transport Paths

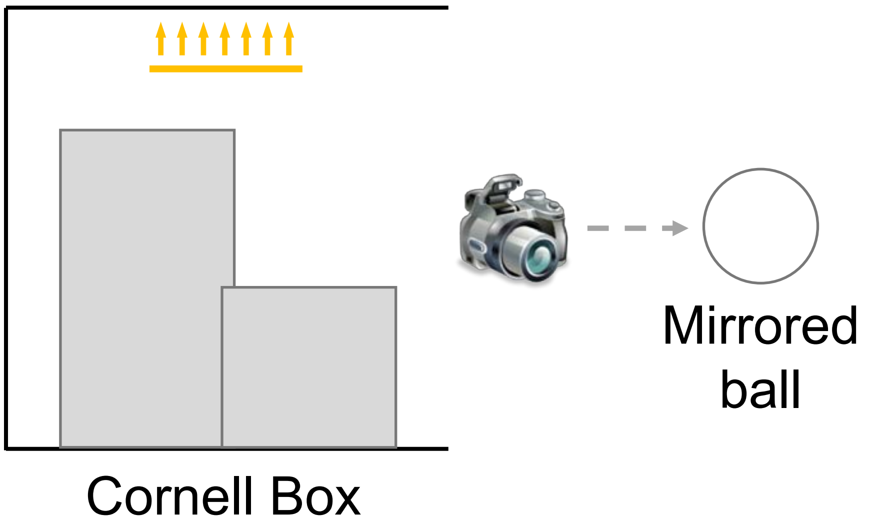
- Using multiple importance sampling, we can combine all these path sampling schemes, resulting in:

$$\langle I \rangle_{\mathrm{MIS}} = \sum_{s \geq 0} \sum_{t \geq 0} w_{s,t}(\bar{x}_{s,t}) \frac{f(\bar{x}_{s,t})}{p_{s,t}(\bar{x}_{s,t})}$$

where $w_{s,t}$ is the weighting function

- Balance heuristic

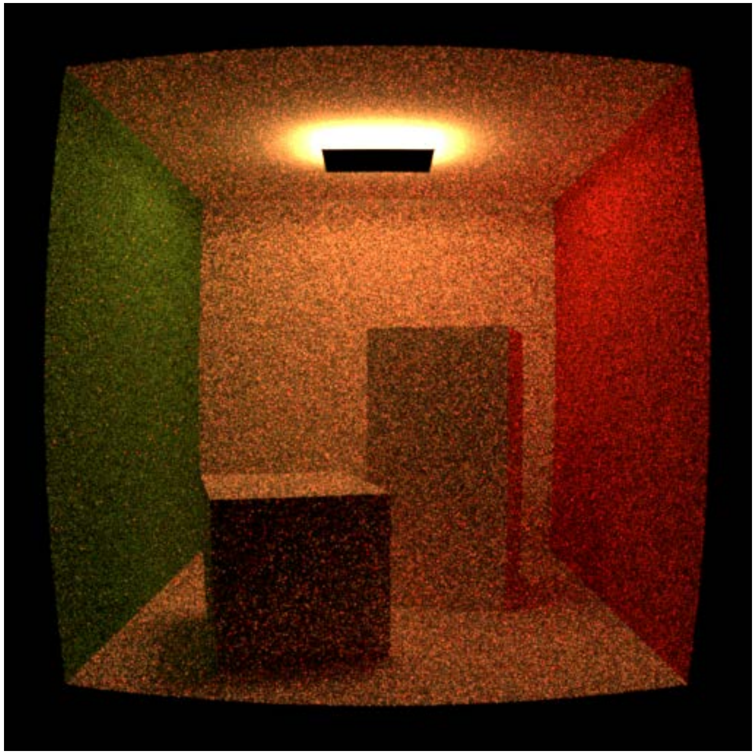
Example: Modified Cornell Box



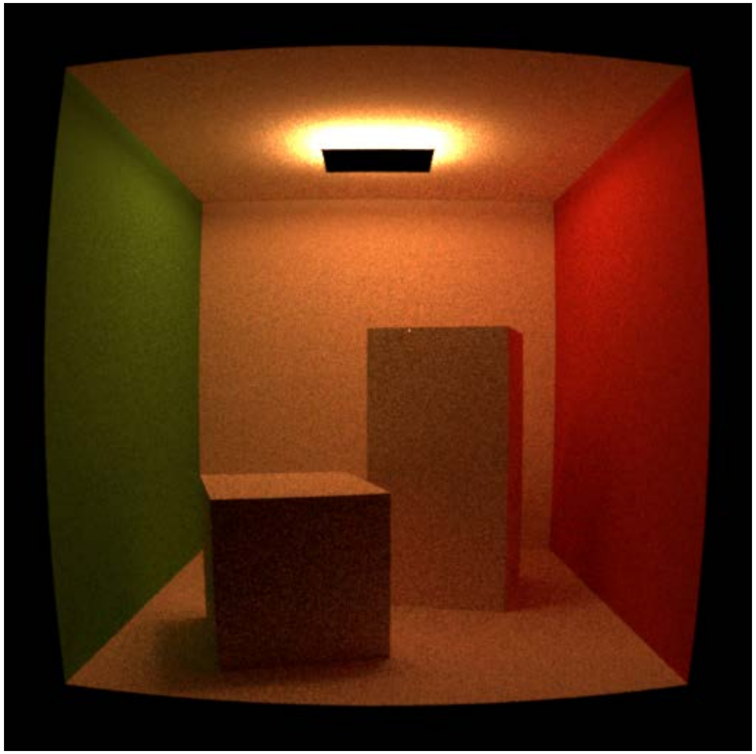
- Area light facing up
  - The scene is lit largely *indirectly*
- Camera facing a mirrored ball
  - The scene is observed indirectly
- Difficult to render with unidirectional path tracing

Example: Modified Cornell Box

(Rendered in equal-time)



Path tracing



Bidirectional path tracing

Metropolis light transport (MLT)

- A Markov Chain Monte Carlo (MCMC) framework implementing the Metropolis-Hasting method first proposed by Veach and Guibas in 1997
- Capable of efficiently constructing “difficult” transport paths
- Lots of ongoing research along this direction

Metropolis-Hasting Method

- A Markov-Chain Monte Carlo technique

- Given a non-negative function f, generate a chain of (correlated) samples X_1, X_2, X_3, … that follow a probability density proportional to f

- **Main advantage:** f does not have to be a PDF (i.e., unnormalized)

Metropolis-Hasting Method

- Input
  - Non-negative function f
  - Probability density $g(y \to x)$ suggesting a candidate for the next sample value x, given the previous sample value y

- The algorithm: given current sample X_i
  1. Sample X' from $g(X_i \to X')$
  2. Let $a = \frac{f(X')}{f(X_i)} \frac{g(X' \to X_i)}{g(X_i \to X')}$
  3. Set X_{i+1} to X' with prob. a; otherwise, set X_{i+1} to X_i

- Start with arbitrary initial state X_0

Path Mutations

- The key step of the MLT
- Given a transport path $\bar{x}$, we need to define a transition probability $g(\bar{x} \to \bar{y})$ to allow sampling mutated paths $\bar{y}$ based on $\bar{x}$
  - Given this transition density, the acceptance probability is then given by

$$a(\bar{x} \to \bar{y}) = \min \left\{ 1, \ \frac{f(\bar{y})}{f(\bar{x})} \frac{g(\bar{y} \to \bar{x})}{g(\bar{x} \to \bar{y})} \right\}$$

Desirable Mutation Properties

- High acceptance probability
  - $a(\bar{x} \to \bar{y})$ should be large with high probability
- Large changes to the path
- Ergodicity (never stuck in some-region of the path space)
  - $g(\bar{x} \to \bar{y})$ should be non-zero for all $\bar{x},\ \bar{y}$ with $f(\bar{x}) > 0,\ f(\bar{y}) > 0$
- Low cost

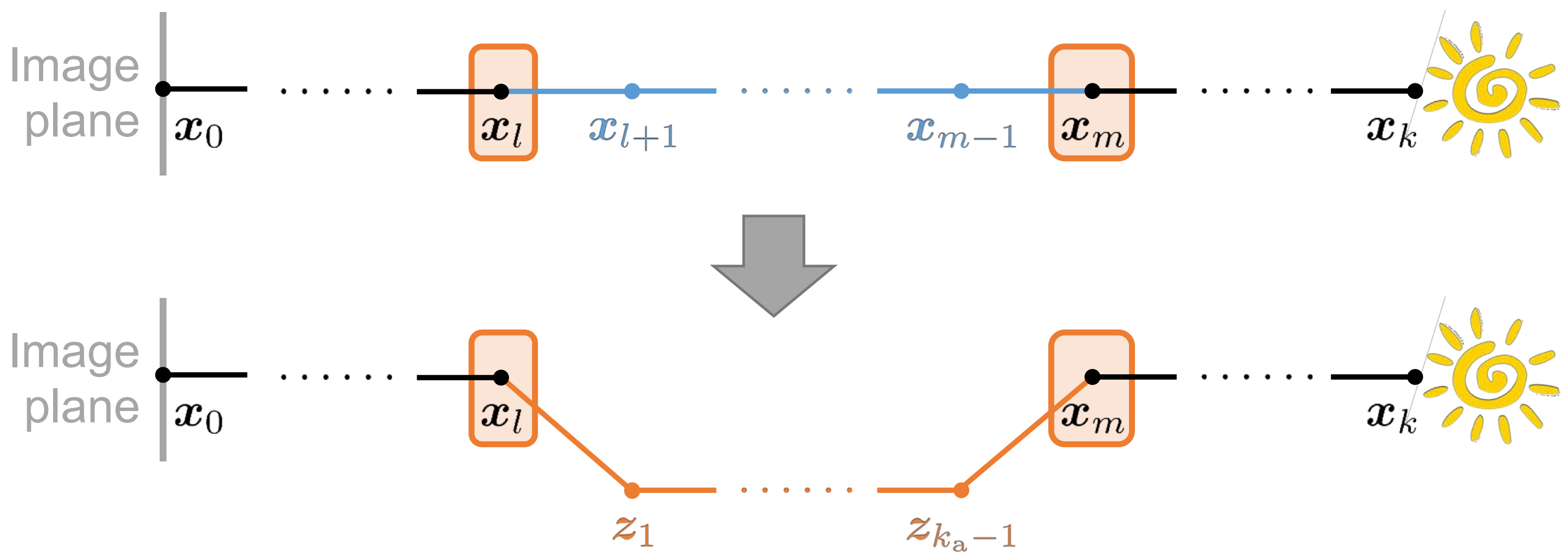
Path Mutation Strategies

- [Veach & Guibas 1997]
  - Bidirectional mutation
  - Path perturbations
  - Lens sub-path mutation
- [Jakob & Marschner 2012]
  - Manifold exploration
- [Li et al. 2015]
  - Hamiltonian Monte Carlo

…

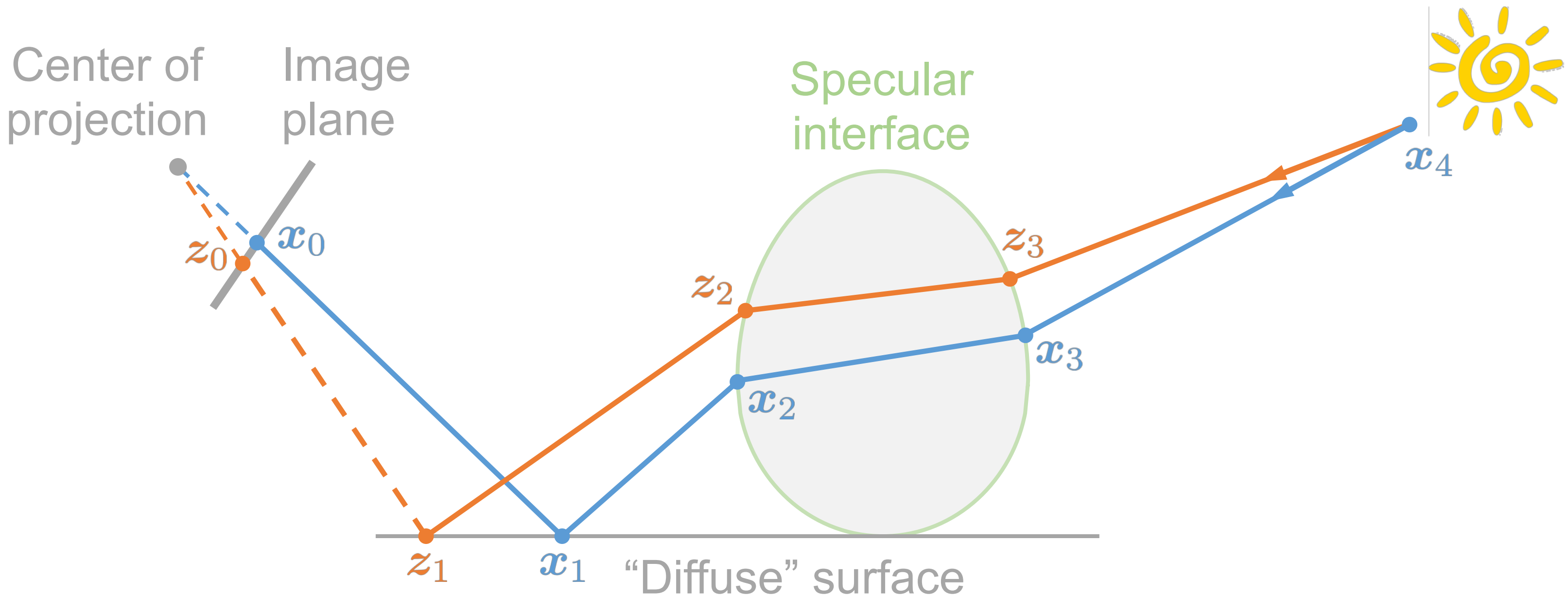
Bidirectional Path Mutations

- Basic idea
  - Given a path $\bar{x} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_k)$, pick *l*, *m* and replace the vertices $\boldsymbol{x}_{l+1}, \ldots, \boldsymbol{x}_{m-1}$ with $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{k_a-1}$
    - *l* and *m* satisfies $-1 \leq l < m \leq k+1$



Path Perturbation

- Slightly modify the direction $\boldsymbol{x}_m \rightarrow \boldsymbol{x}_{m\text{-}1}$ (at random)
- Trace a ray from $\boldsymbol{x}_m$ with this new direction to form the new sub-path



Results

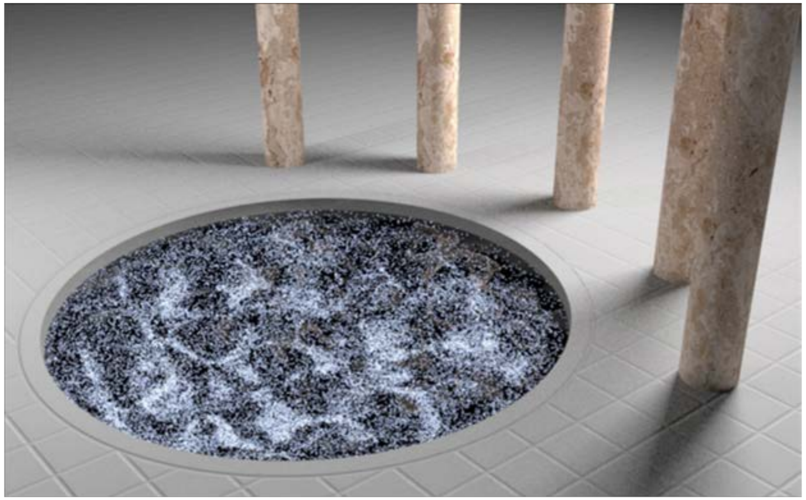
BDPT

MLT
(equal-time)

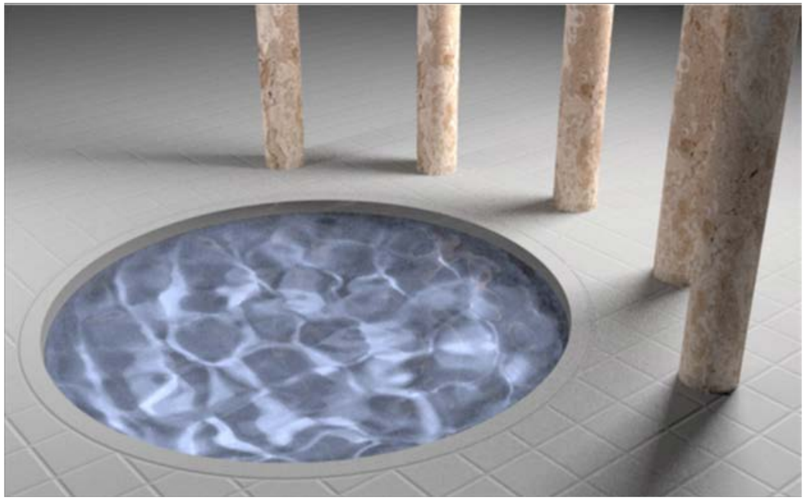
[Veach 1997]

Results

[Veach 1997]



BDPT



MLT
(equal-time)

Summary

- Physics-based rendering is a rich field

- Things we have covered
  - Light transport model
    - Rendering equation (reflection and refraction)

  - Monte Carlo solutions
    - Path tracing
    - Bidirectional path tracing
    - Metropolis light transport

Topics we have not covered

- Radiative transfer (sub-surface scattering)
- Unbiased methods
  - Primary sample space & multiplexed MLT
  - Gradient-domain PT/MLT
- Biased methods
  - Photon mapping
  - Lightcuts
  - Monte Carlo denoising
  - Diffusion approximations
- ……