LIGHT TRANSPORT

Advanced Computer Graphics 2017

Erik Sintorn

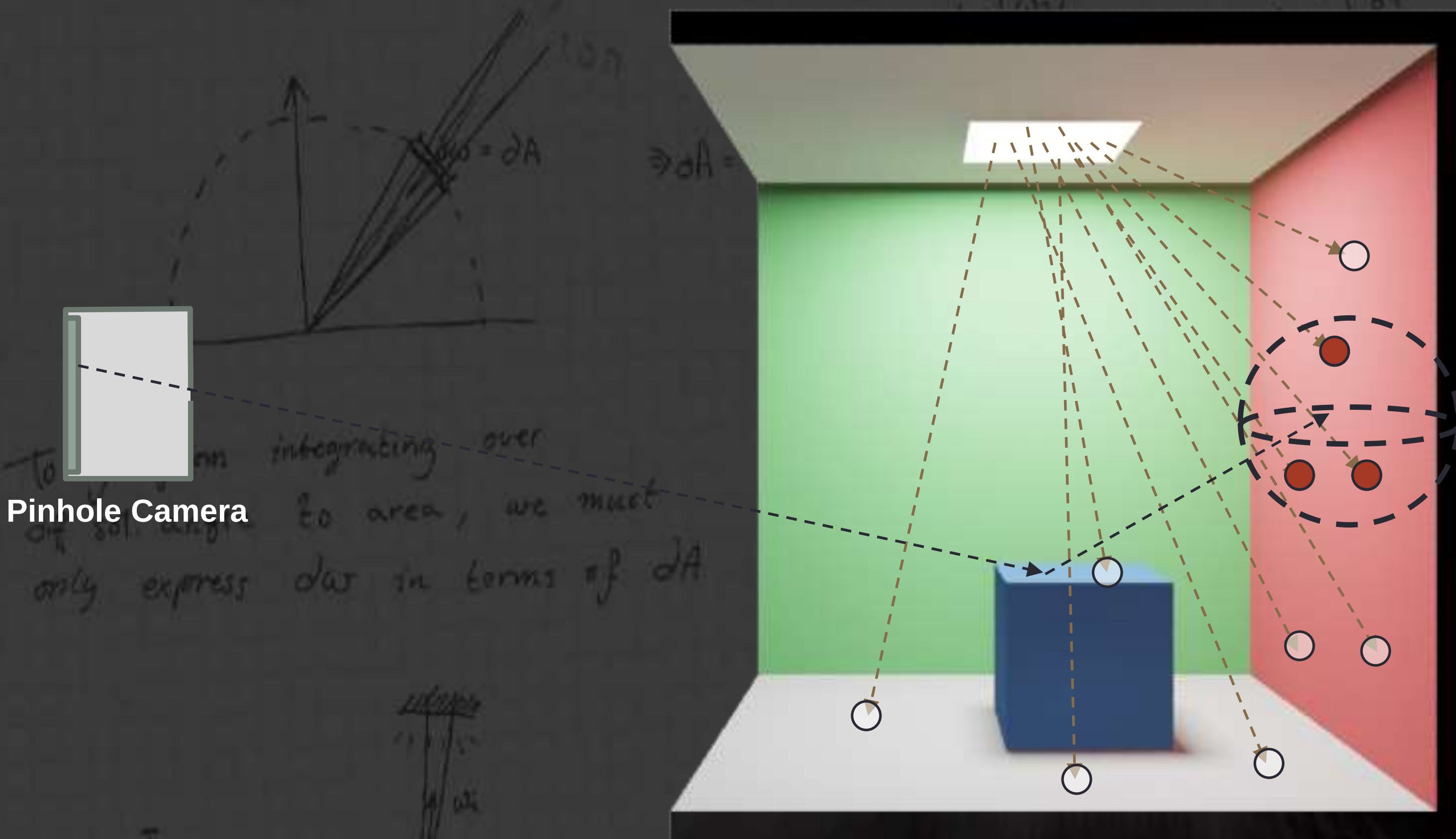
Before we start:

- Remember to choose a subject for your presentation soon.
- And your project.
- Student representatives:
  - JOHAN BACKMAN johback@student.chalmers.se
  - KEVIN BJÖRKLUND kevinb@student.chalmers.se
  - JONAS HULTÉN jonashu@student.chalmers.se
  - HAMPUS LIDIN lidin@student.chalmers.se
  - VICTOR OLAUSSON vicola@student.chalmers.se
  - Come for a quick talk with me during recess.
- Muddy Cards!

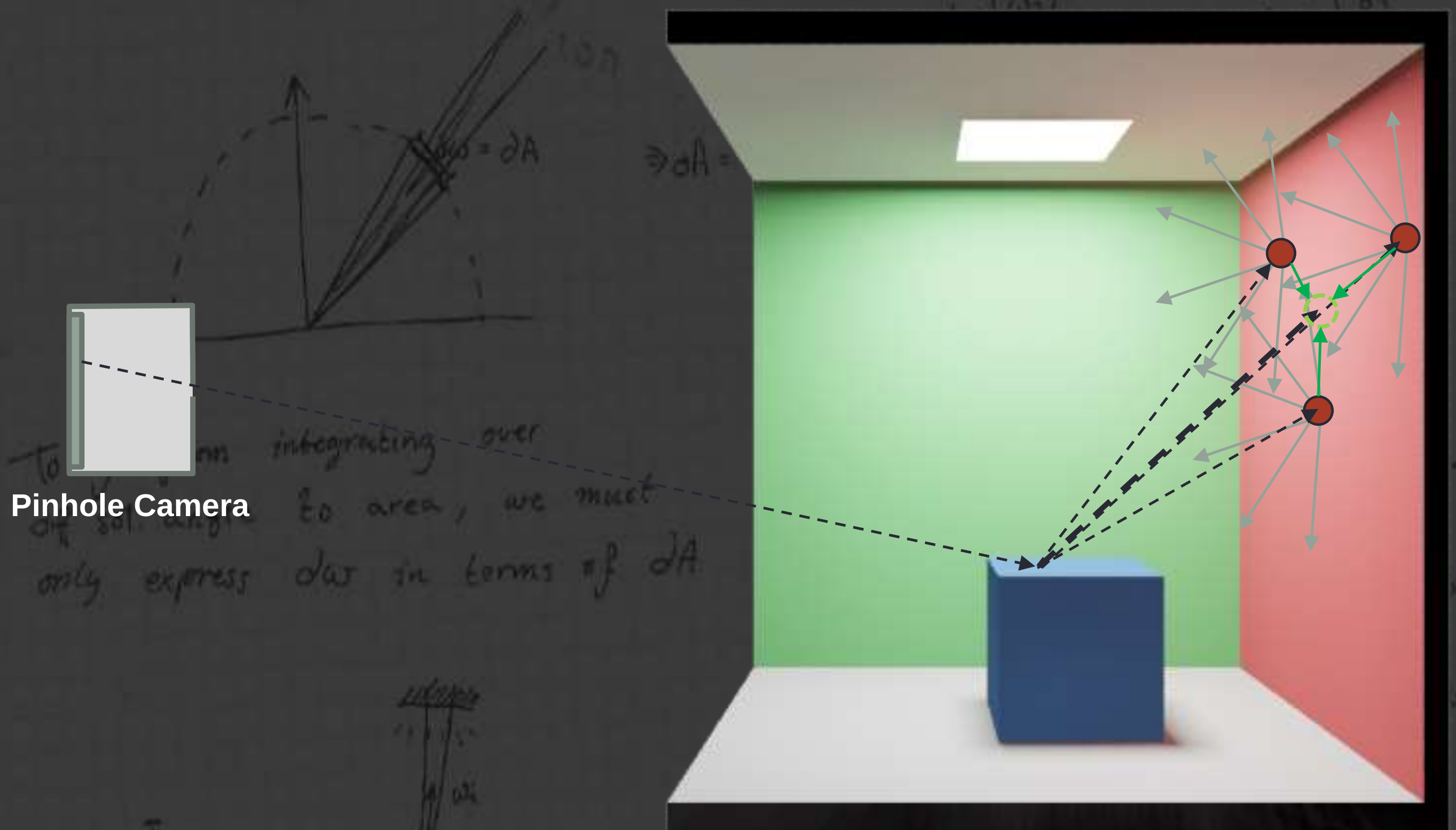
Light Transport Simulation

- Rendering an image is a matter of ”simulating” how light propagates through a virtual scene and lands on a virtual camera film.
- Many algorithms exist, and the *best* one depends on many factors.
- For a long time, *Photon Mapping* and *Irradiance Caching* were extremely popular.
  - Trade correctness for speed.
  - Will cover these only very briefly.

Photon Mapping



Irradiance Caching

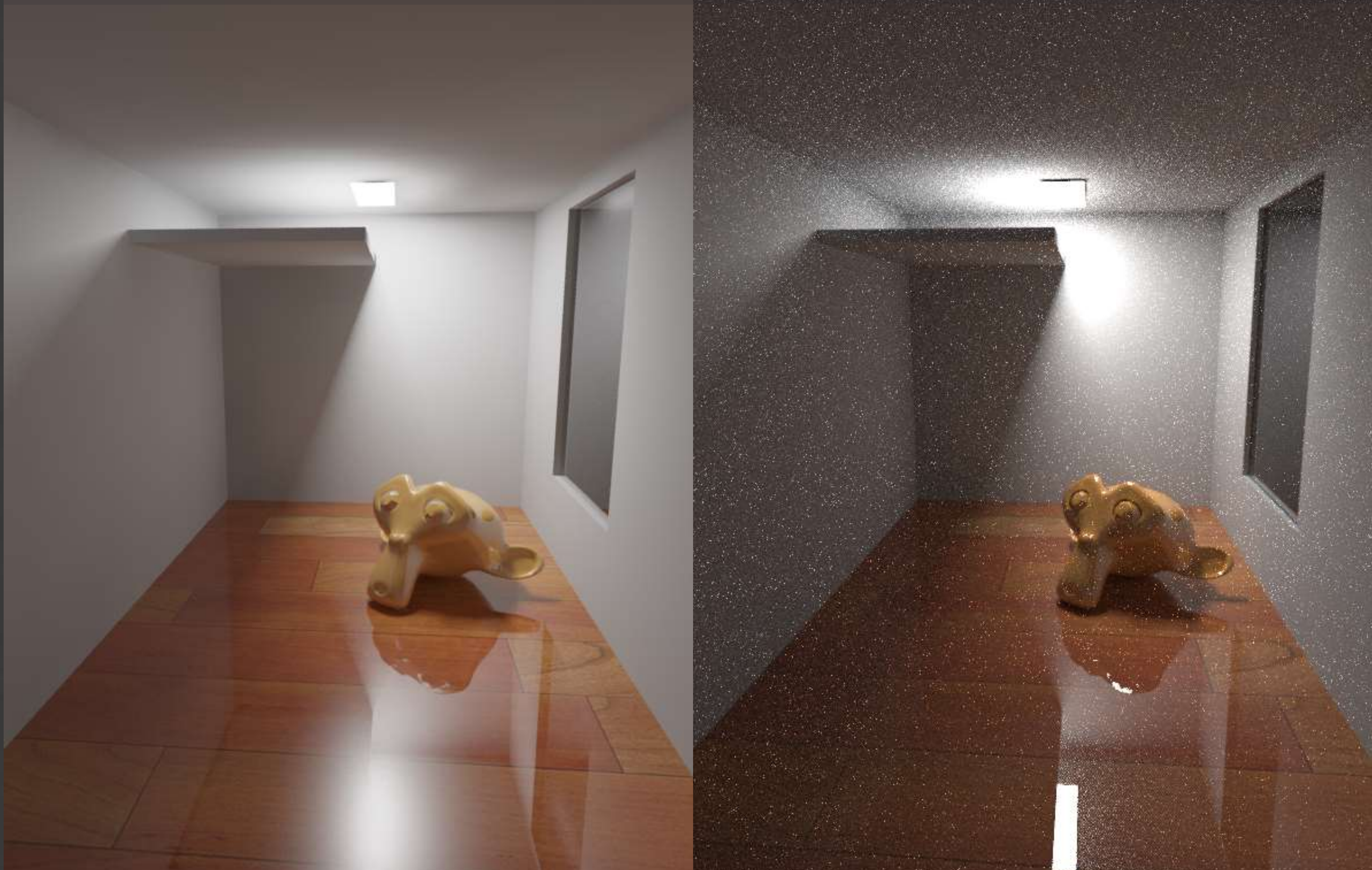


Path Tracing

- Path tracing is an *algorithm* for rendering images.
  - Introduced by James Kajiya in 1986 as a numerical solution to the Rendering Equation.
  - The algorithm is *convergent* and *unbiased*.
  - Has long been considered too noisy/slow to be used in industry.
- Today, almost all commercial renderers use some form of unbiased pathtracing (at least optionally).
  - Pixar (for example) only switched completely very recently.
- Why the sudden popularity?

Mental Ray (photon mapping) 32s

iRay (path tracing) 32s



Mental Ray (photon mapping) 2m8s

iRay (path tracing) 2m8s

Mental Ray (photon mapping) 2m8s

iRay (path tracing) 2m8s

iRay (path tracing) ~1h

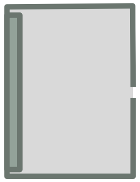
Mental Ray 15m, 100M Photons, FG 1.0

- Artifacts never go away
- Scene specific tuning choices

iRay (path tracing) 15m

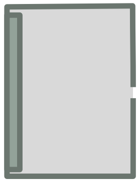
- Immediate response
- Much easier to parallelize

Where does an image come from?

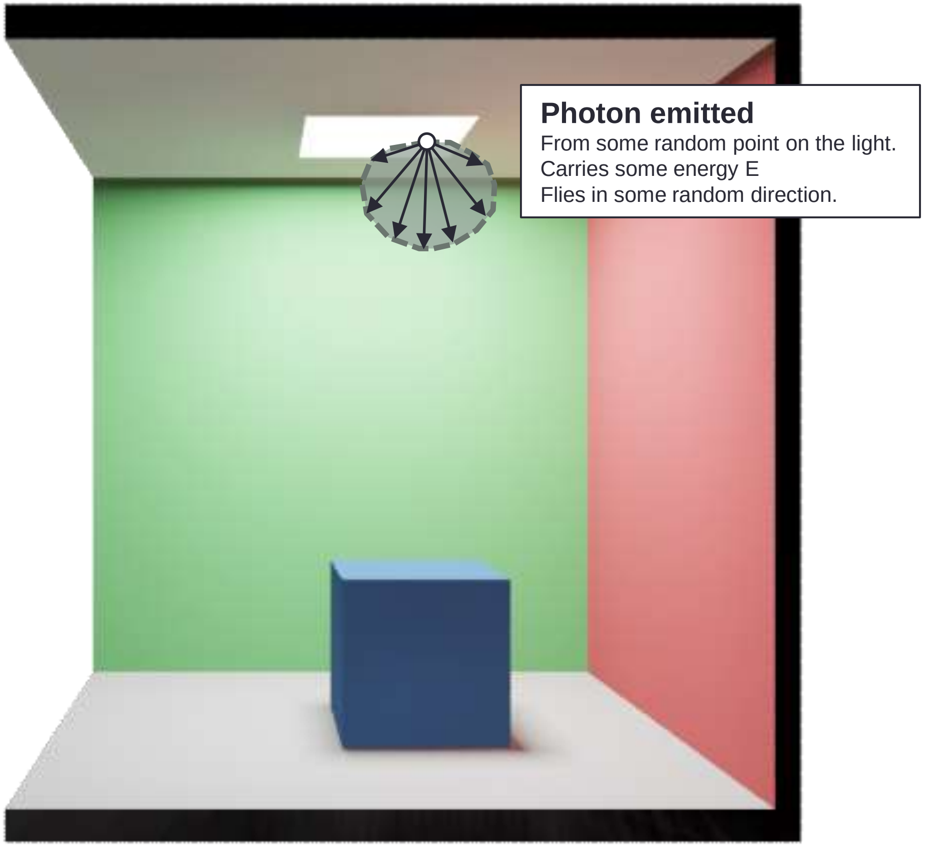


Pinhole Camera

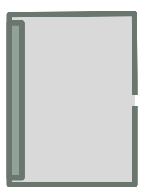
Where does an image come from?



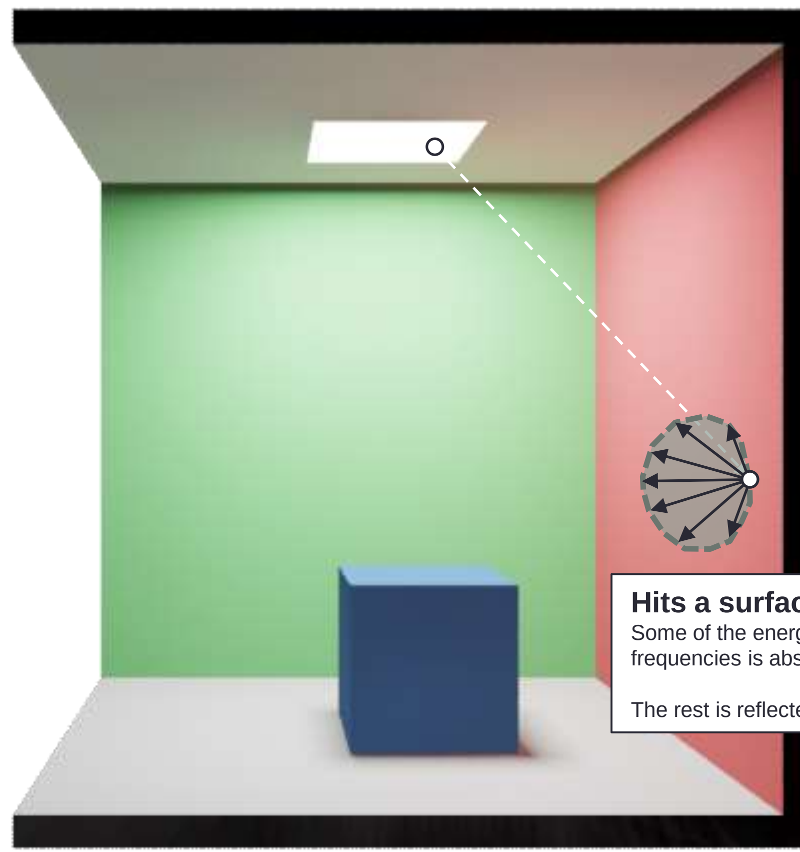
Pinhole Camera



Where does an image come from?



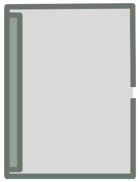
Pinhole Camera



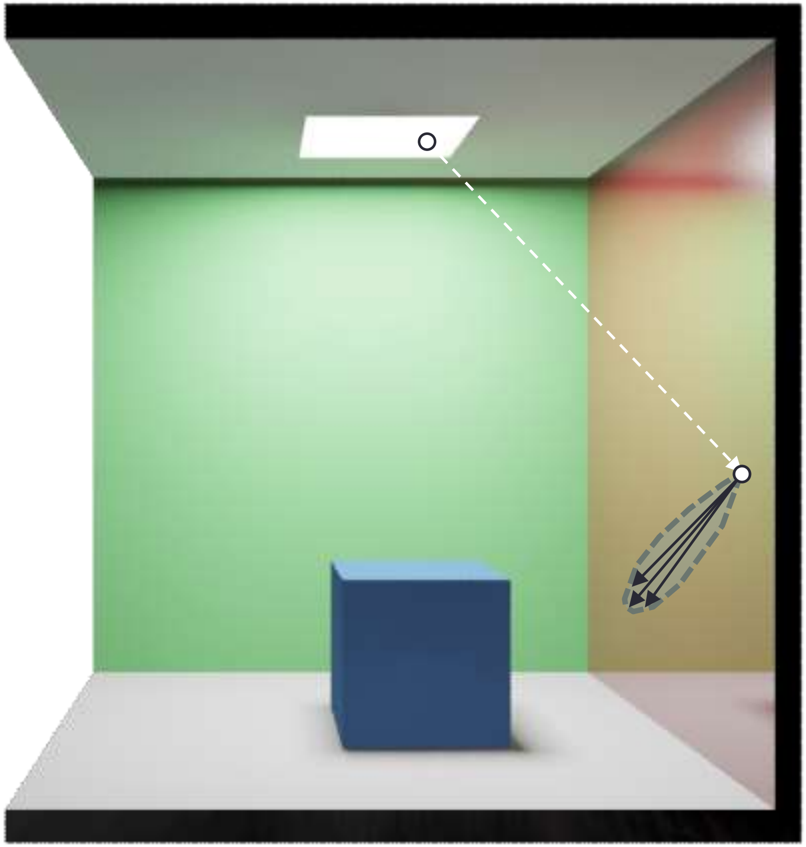
Hits a surface
Some of the energy of some frequencies is absorbed.

The rest is reflected in some direction

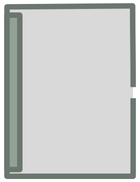
Where does an image come from?



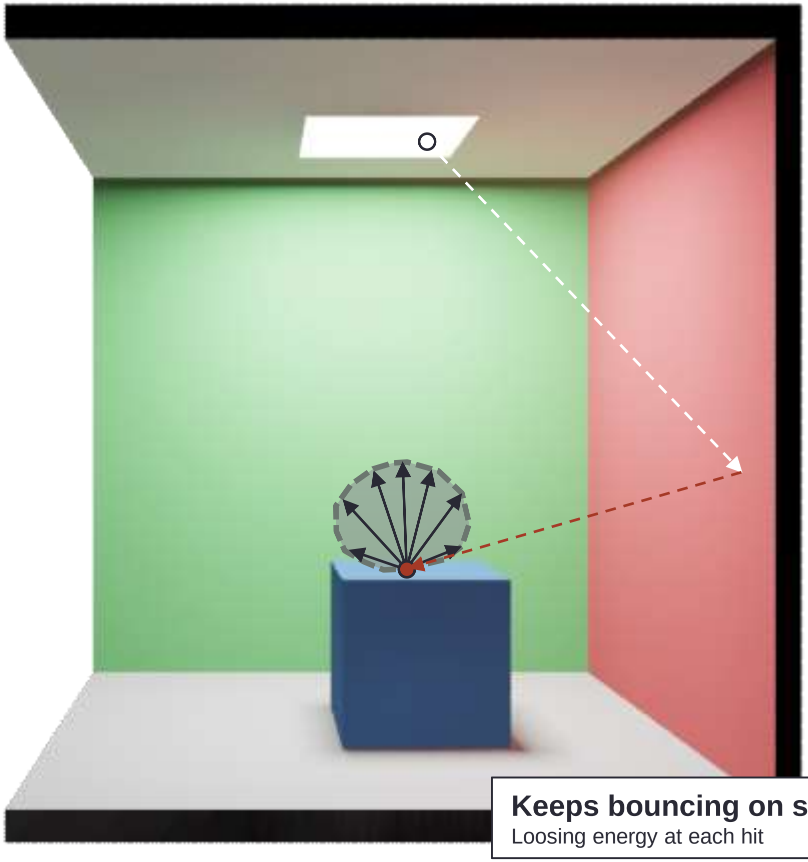
Pinhole Camera



Where does an image come from?

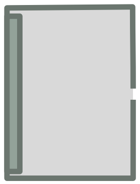


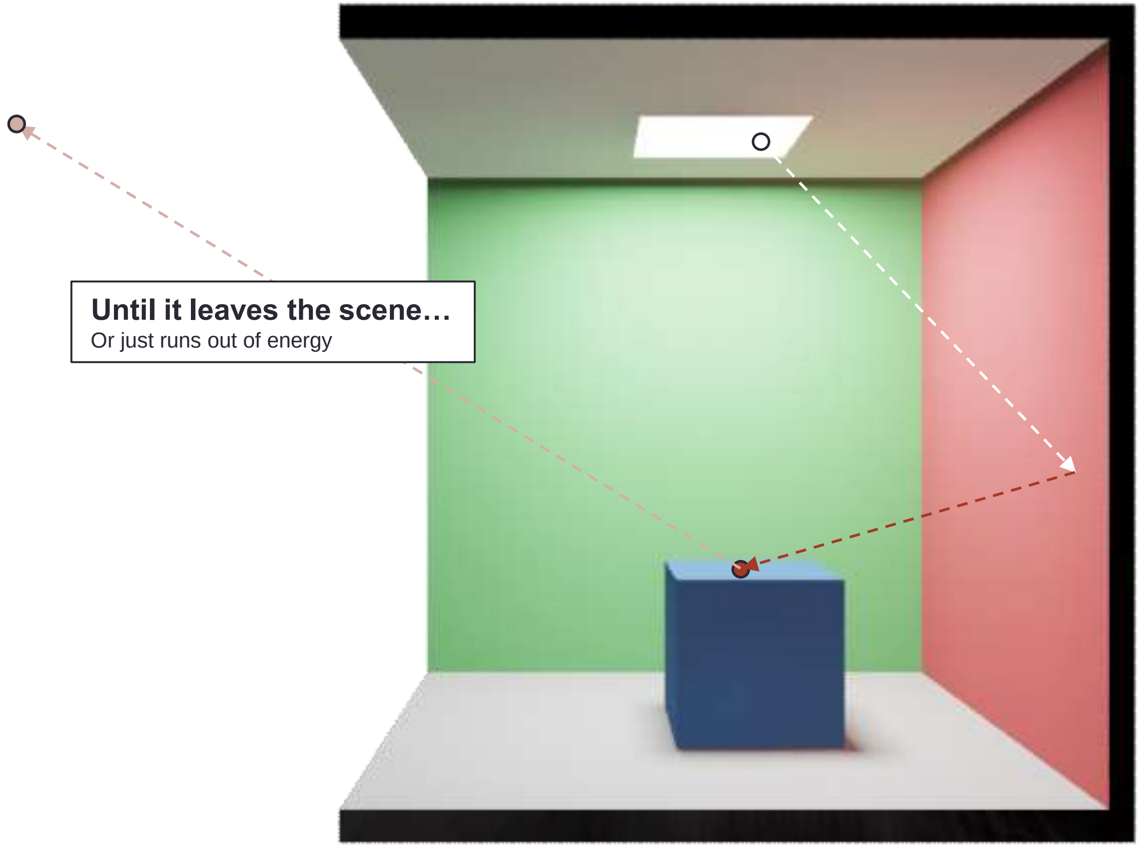
Pinhole Camera



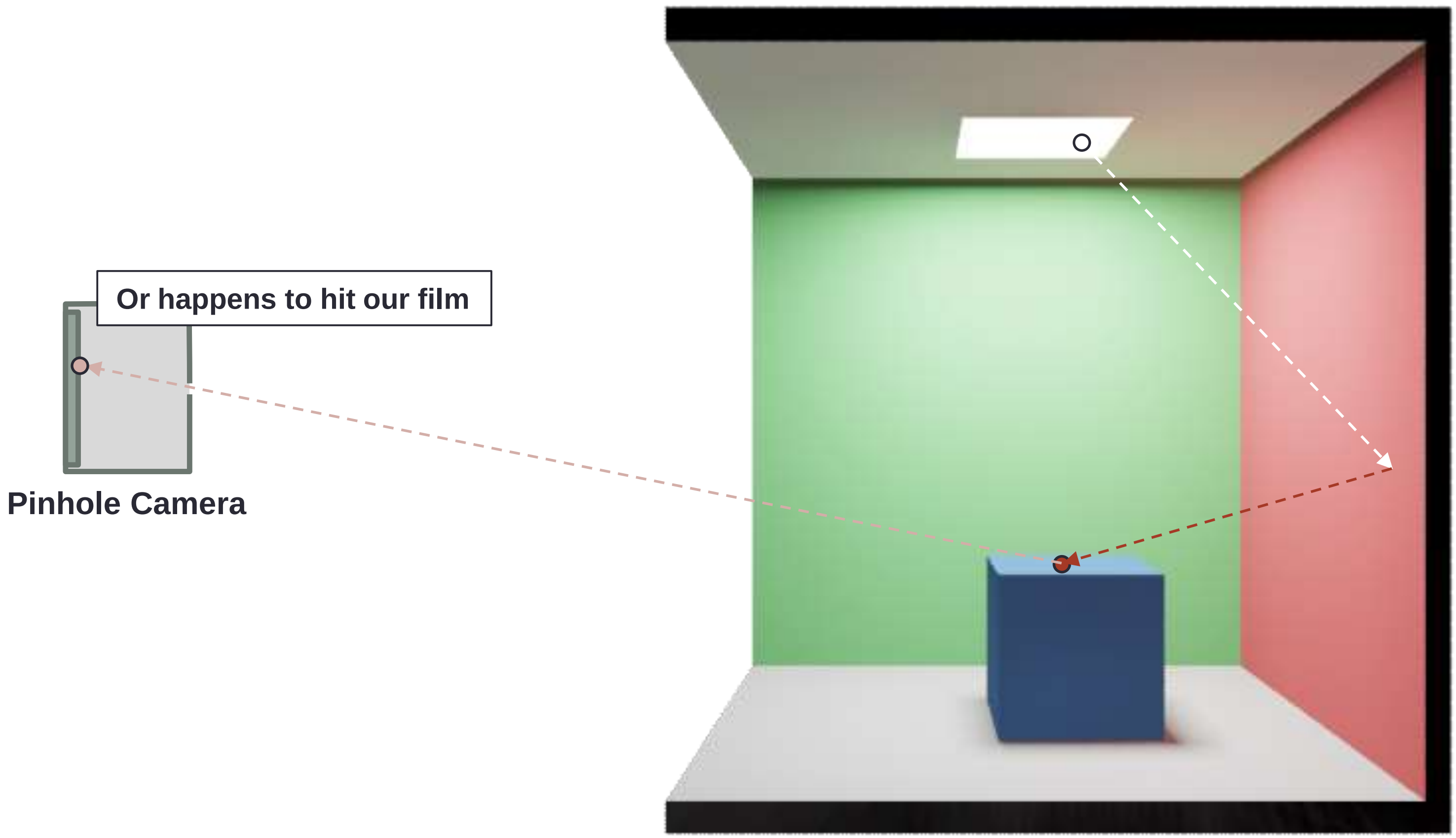
Keeps bouncing on surfaces
Loosing energy at each hit

Where does an image come from?

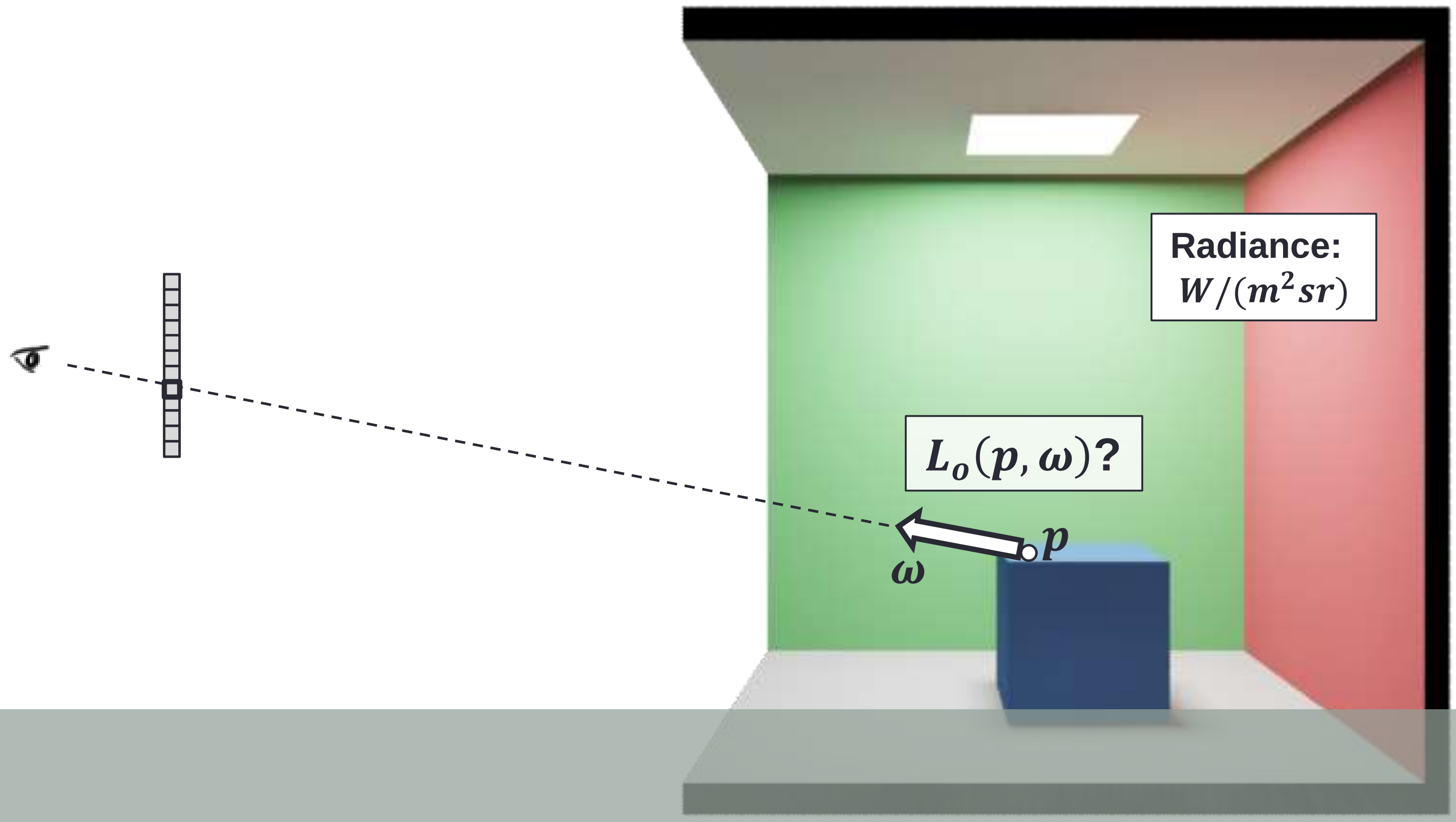




Where does an image come from?

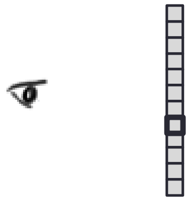


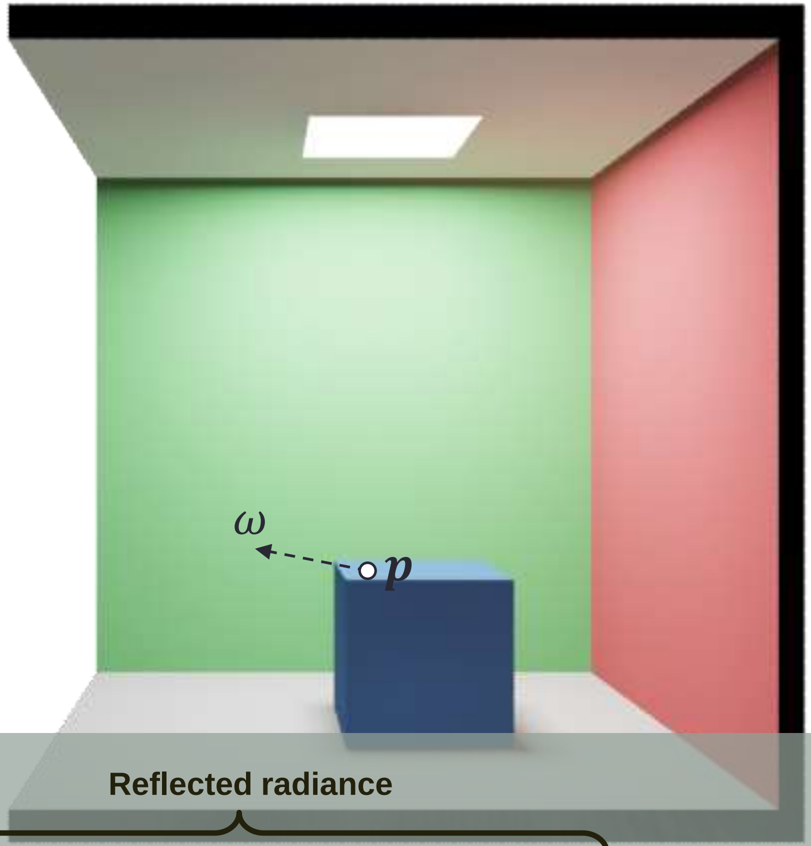
Light Transport Equation



$$L_o(\mathbf{p},\omega) = L_e(\mathbf{p},\omega) + \int_{\Omega} f(\mathbf{p},\omega,\omega')L_i(\mathbf{p},\omega')\cos(\mathbf{n},\omega')\,d\omega'$$

Light Transport Equation





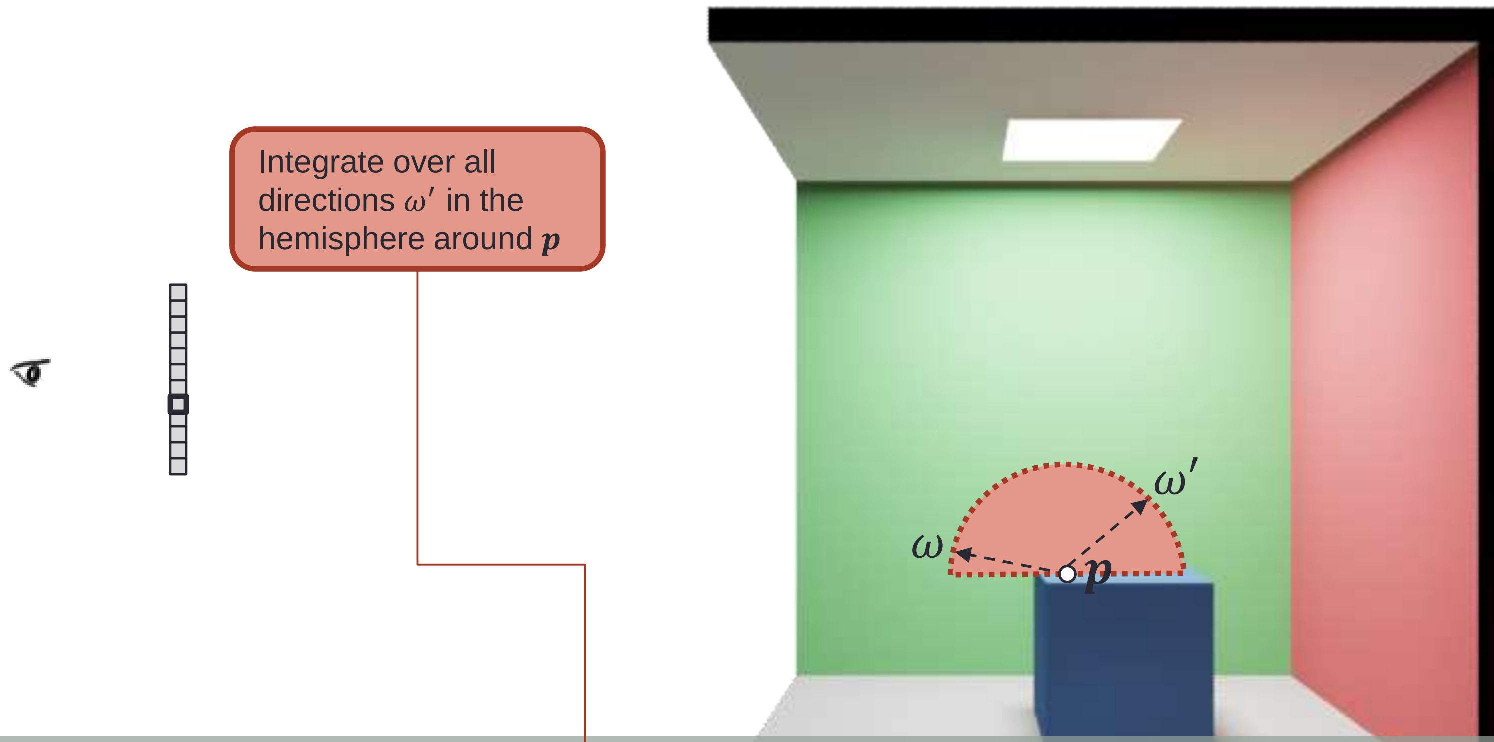
Emitted radiance

Reflected radiance

$$L_o(\mathbf{p},\omega) = \underbrace{L_e(\mathbf{p},\omega)}_{\text{Emitted radiance}} + \underbrace{\int_\Omega f(\mathbf{p},\omega,\omega')L_i(\mathbf{p},\omega')\cos(\mathbf{n},\omega')\,d\omega'}_{\text{Reflected radiance}}$$

Light Transport Equation

Integrate over all directions ω' in the hemisphere around $\boldsymbol{p}$

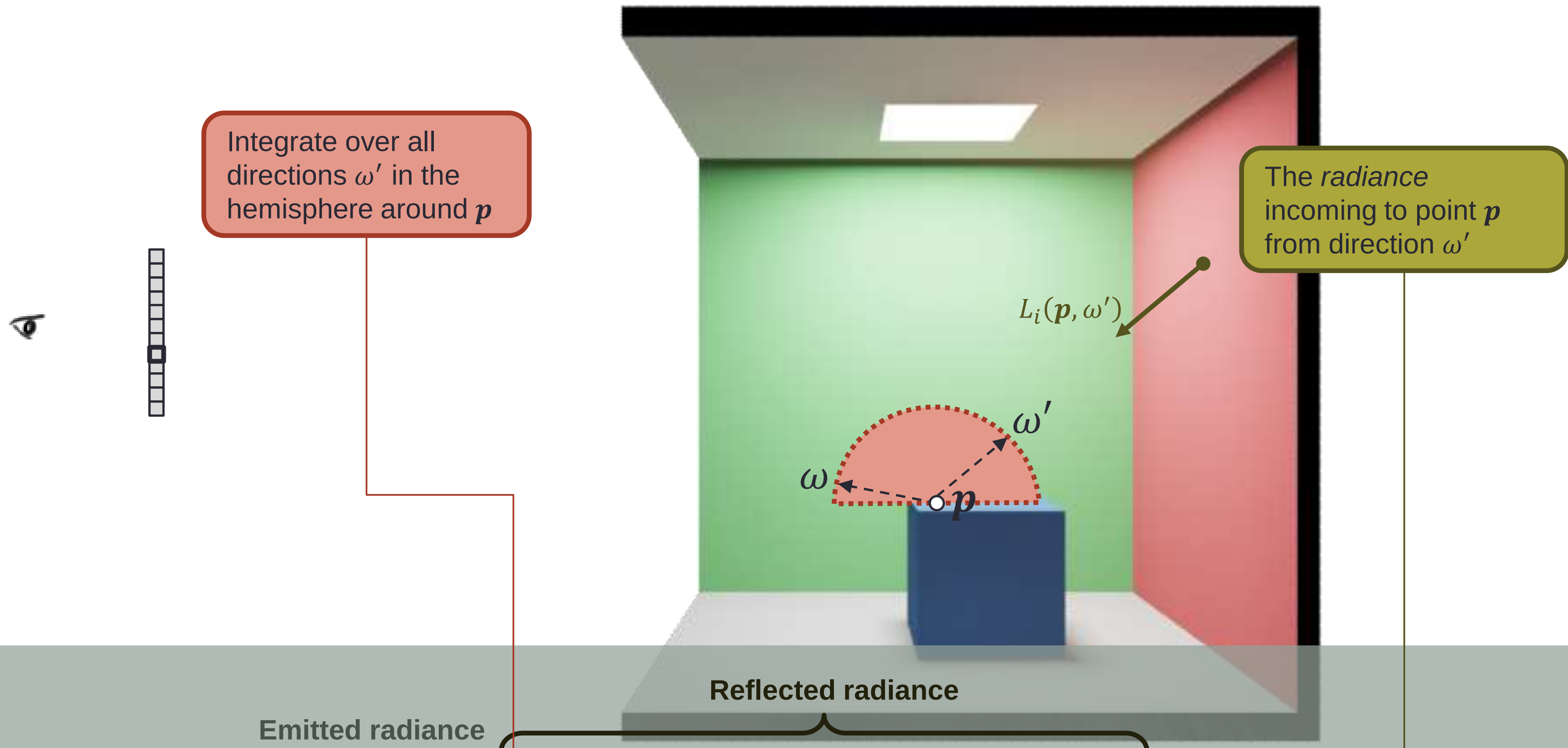


Emitted radiance

Reflected radiance

$$L_o(\mathbf{p},\omega) = L_e(\mathbf{p},\omega) + \int_{\Omega} f(\mathbf{p},\omega,\omega')L_i(\mathbf{p},\omega')\cos(\mathbf{n},\omega')\,d\omega'$$

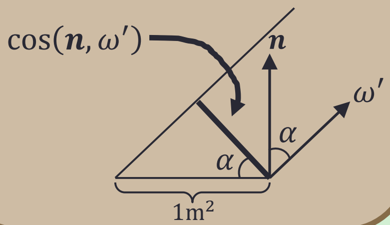
Light Transport Equation



$$L_o(\boldsymbol{p}, \omega) = L_e(\boldsymbol{p}, \omega) + \int_\Omega f(\boldsymbol{p}, \omega, \omega') L_i(\boldsymbol{p}, \omega') \cos(\boldsymbol{n}, \omega')\, d\omega'$$

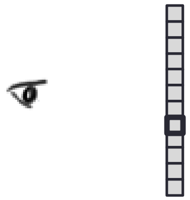
Light Transport Equation

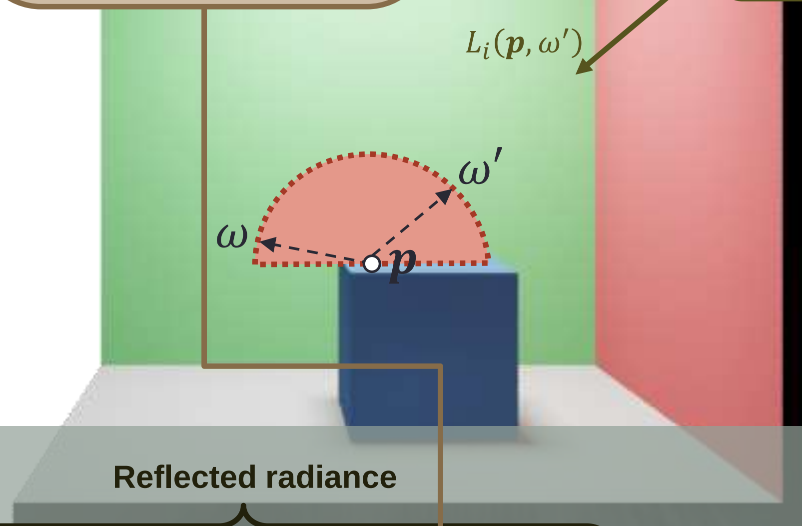
Convert between irradiance parallel to ω and irradiance on surface.



Integrate over all directions ω' in the hemisphere around $\mathbf{p}$

The *radiance* incoming to point $\mathbf{p}$ from direction ω'





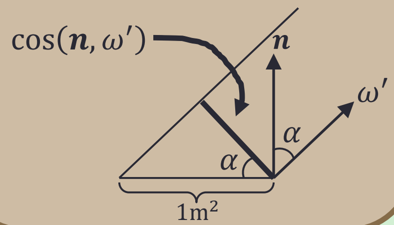
Reflected radiance

Emitted radiance

$$L_o(\mathbf{p},\omega) = L_e(\mathbf{p},\omega) + \int_\Omega f(\mathbf{p},\omega,\omega') L_i(\mathbf{p},\omega') \cos(\mathbf{n},\omega') \, d\omega'$$

Light Transport Equation

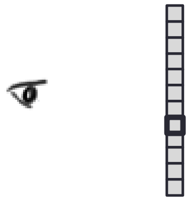
Convert between irradiance parallel to ω' and irradiance on surface.

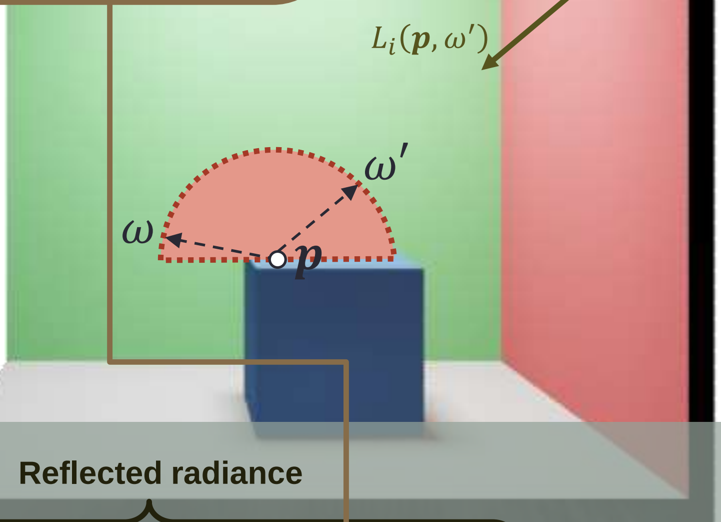


Integrate over all directions ω' in the hemisphere around $\boldsymbol{p}$

The BRDF tells us the differential outgoing *radiance* in direction ω due to irradiance on surface.

The *radiance* incoming to point $\boldsymbol{p}$ from direction ω'

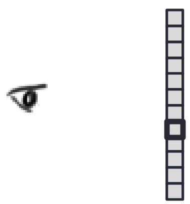


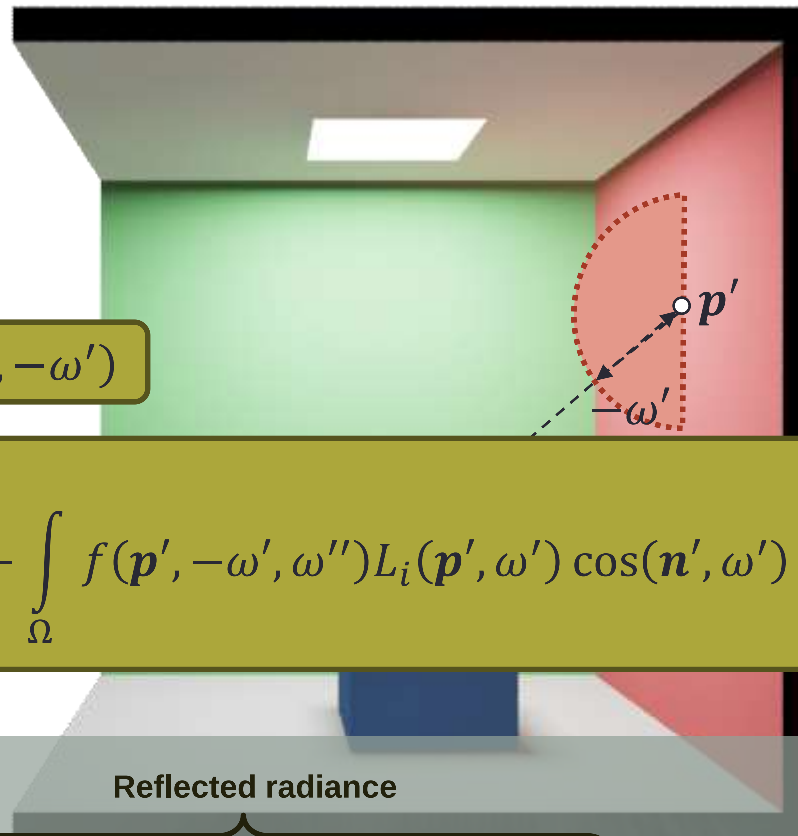


Emitted radiance **Reflected radiance**

$$L_o(\boldsymbol{p},\omega) = \underbrace{L_e(\boldsymbol{p},\omega)}_{\text{Emitted radiance}} + \underbrace{\int_\Omega f(\boldsymbol{p},\omega,\omega') L_i(\boldsymbol{p},\omega') \cos(\boldsymbol{n},\omega')\, d\omega'}_{\text{Reflected radiance}}$$

Light Transport Equation





$$L_i(\mathbf{p},\omega') = L_o(\mathbf{p}',-\omega')$$

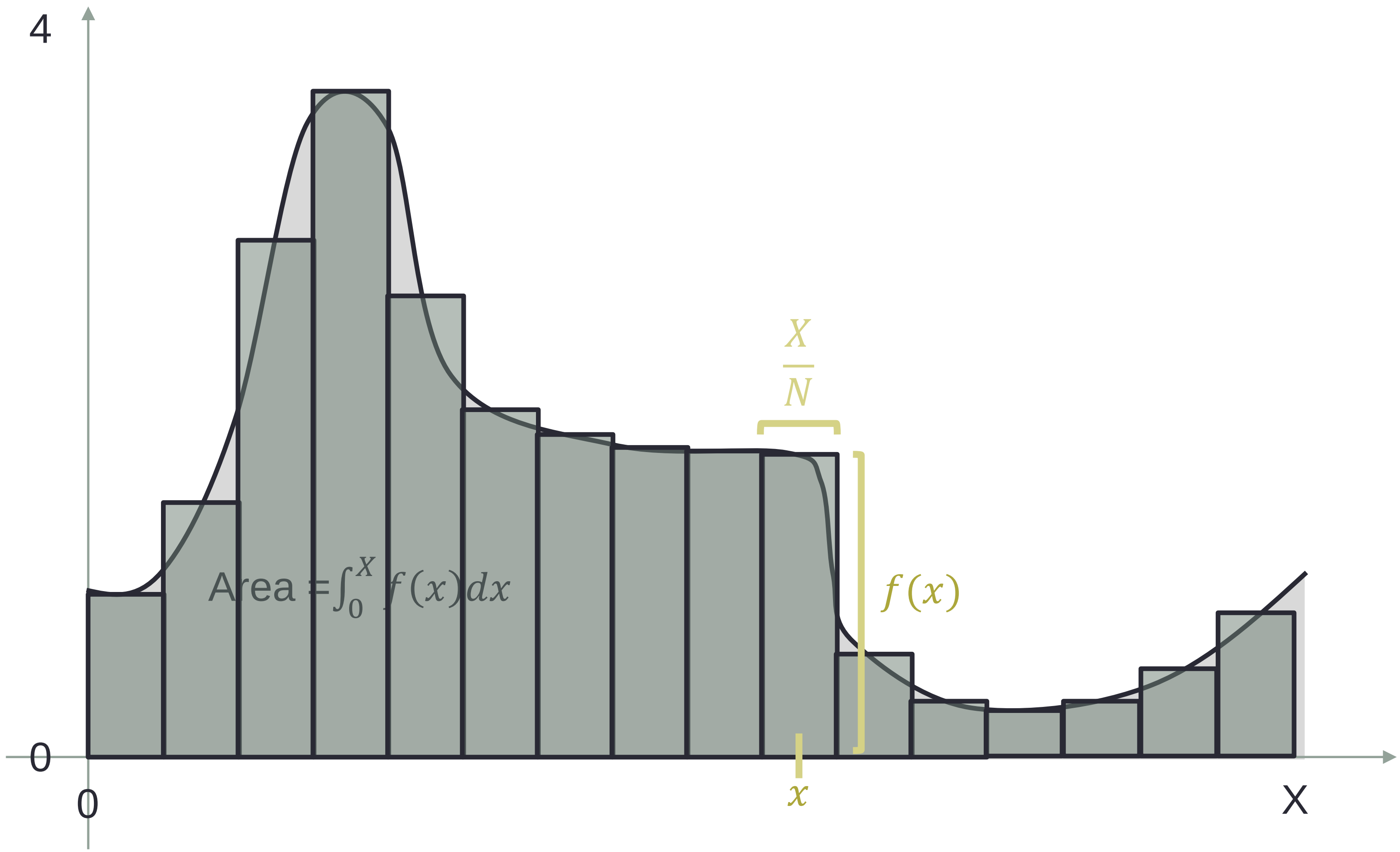
$$= L_e(\mathbf{p}',-\omega') + \int_{\Omega} f(\mathbf{p}',-\omega',\omega'')L_i(\mathbf{p}',\omega')\cos(\mathbf{n}',\omega')\,d\omega'$$

Emitted radiance

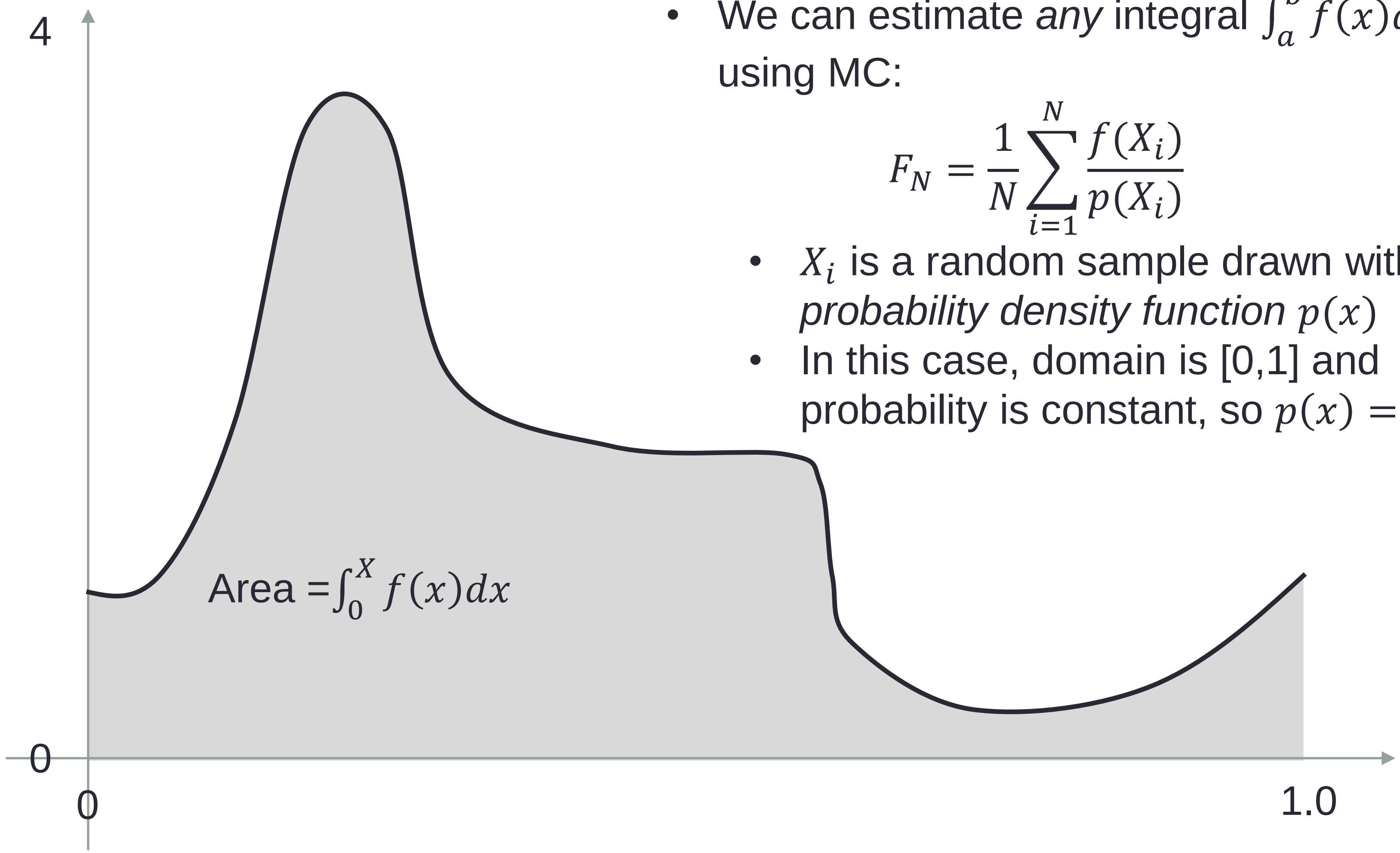
Reflected radiance

$$L_o(\mathbf{p},\omega) = \underbrace{L_e(\mathbf{p},\omega)}_{\text{Emitted radiance}} + \underbrace{\int_{\Omega} f(\mathbf{p},\omega,\omega')L_i(\mathbf{p},\omega')\cos(\mathbf{n},\omega')\,d\omega'}_{\text{Reflected radiance}}$$

Numerical Integration



Monte Carlo Integration



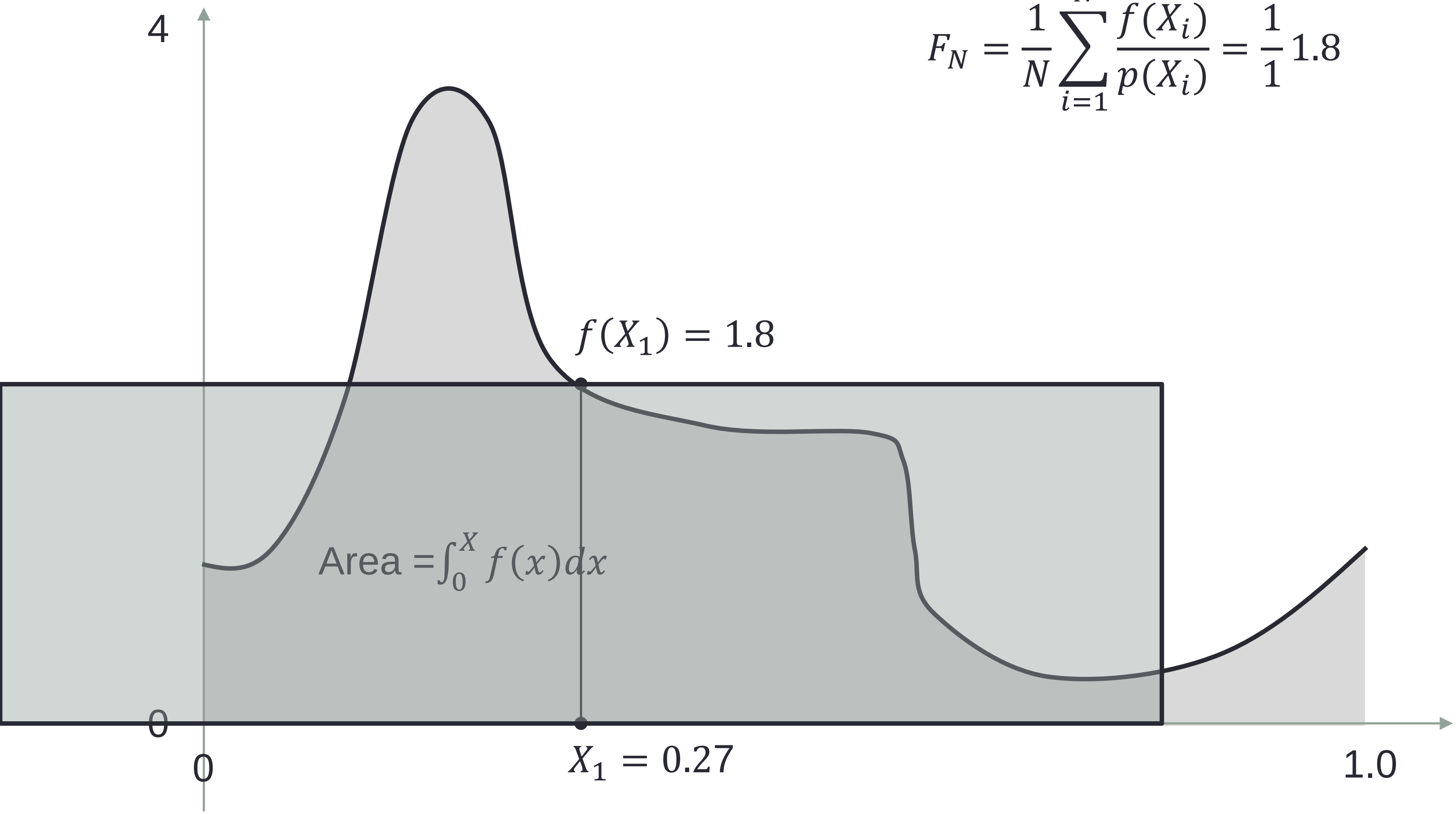
- We can estimate *any* integral $\int_a^b f(x)dx$ using MC:

$$F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}$$

  - X_i is a random sample drawn with *probability density function* $p(x)$
  - In this case, domain is [0,1] and probability is constant, so $p(x) = 1$

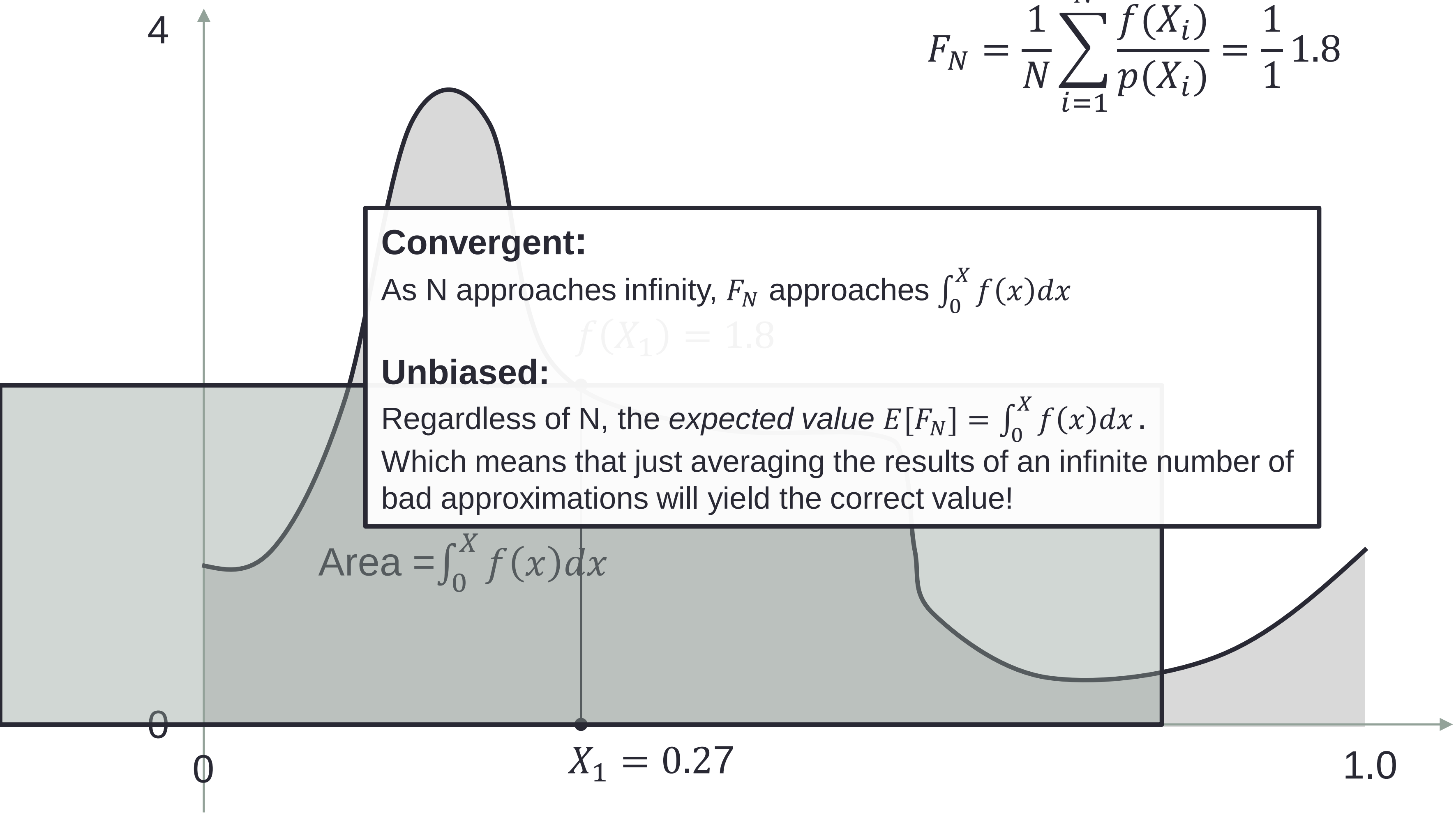
Monte Carlo Integration

$$F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)} = \frac{1}{1}1.8$$



Monte Carlo Integration

$$F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)} = \frac{1}{1}1.8$$



Convergent:

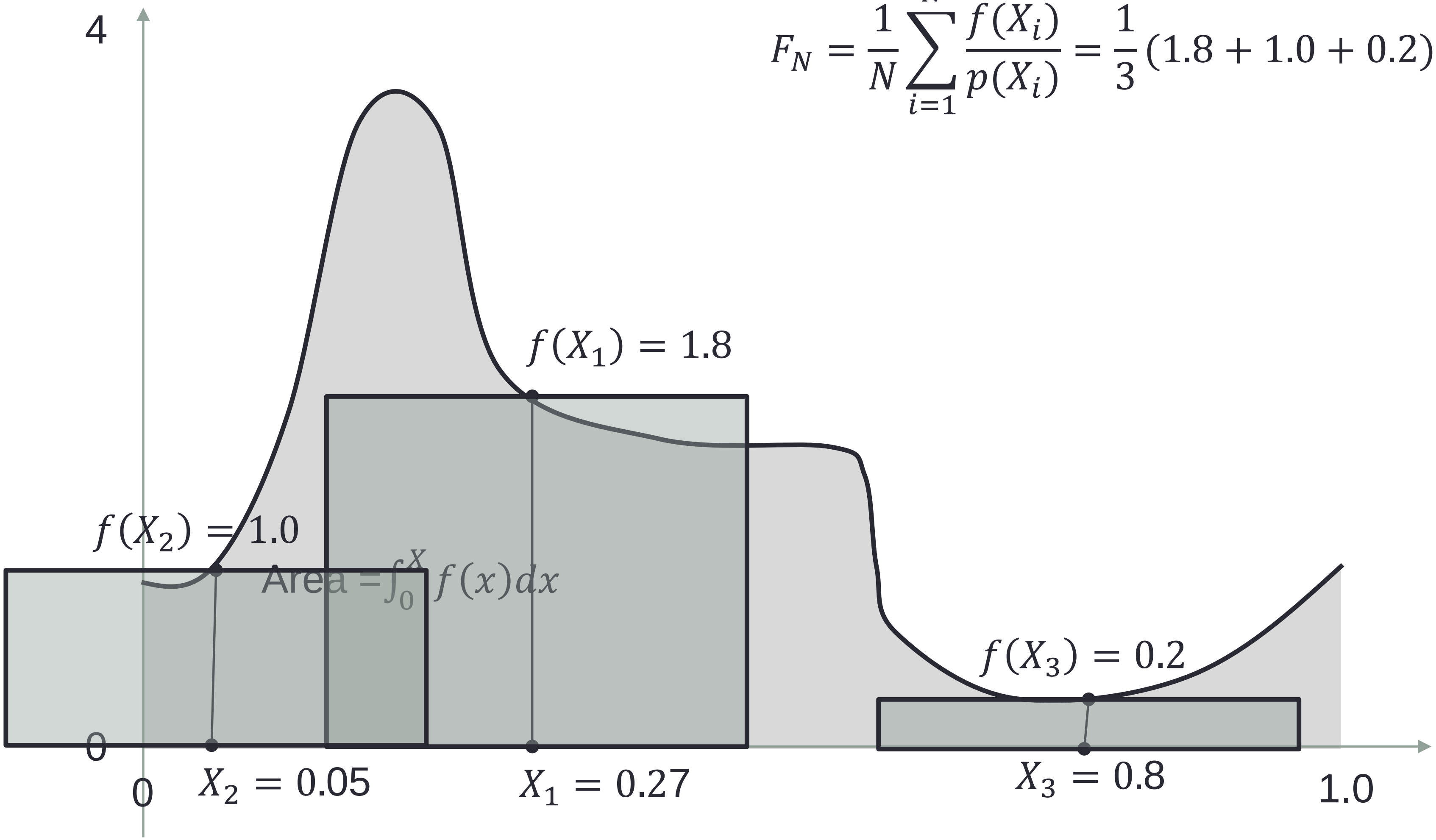
As N approaches infinity, F_N approaches $\int_0^X f(x)dx$

Unbiased:

Regardless of N, the *expected value* $E[F_N] = \int_0^X f(x)dx$.
Which means that just averaging the results of an infinite number of bad approximations will yield the correct value!

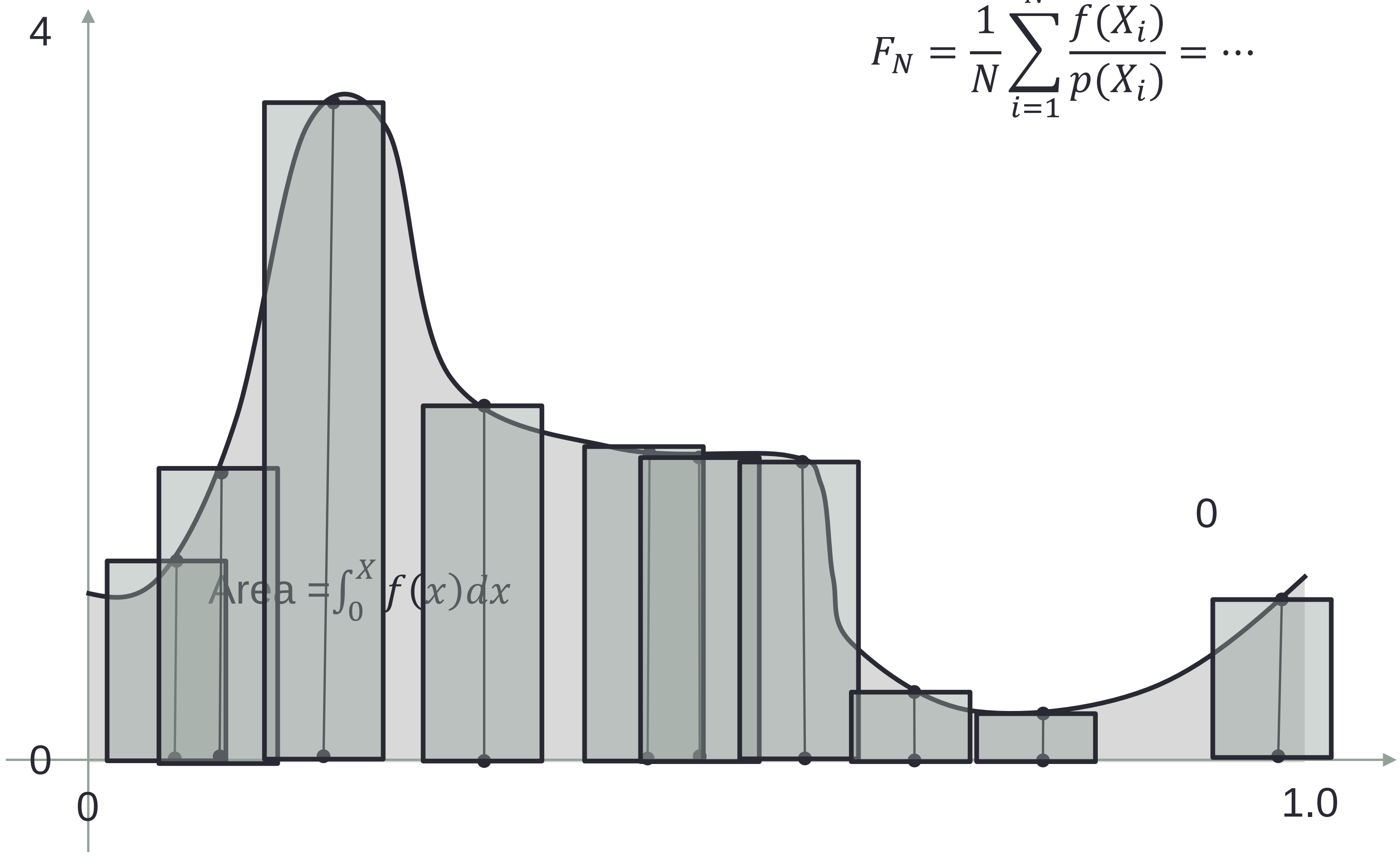
Monte Carlo Integration

$$F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)} = \frac{1}{3}(1.8+1.0+0.2)$$



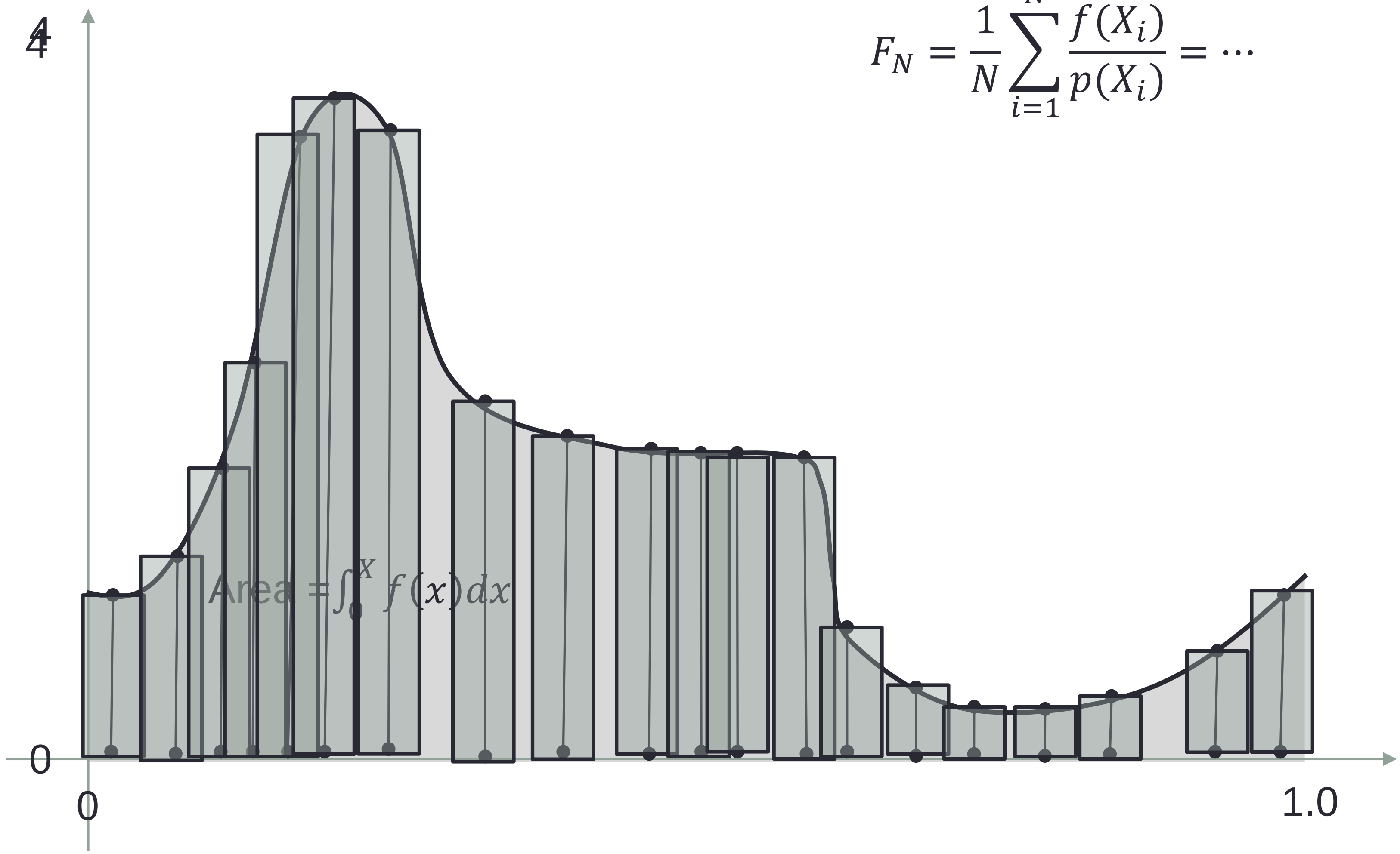
Monte Carlo Integration

$$F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)} = \cdots$$

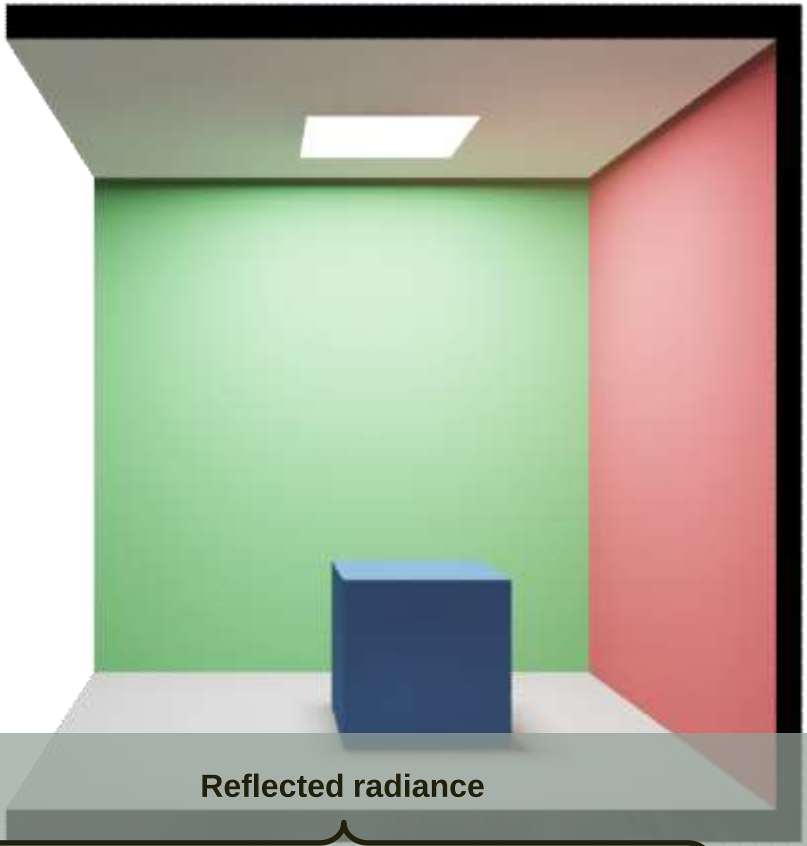


Monte Carlo Integration

$$F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)} = \cdots$$



Monte Carlo Integration

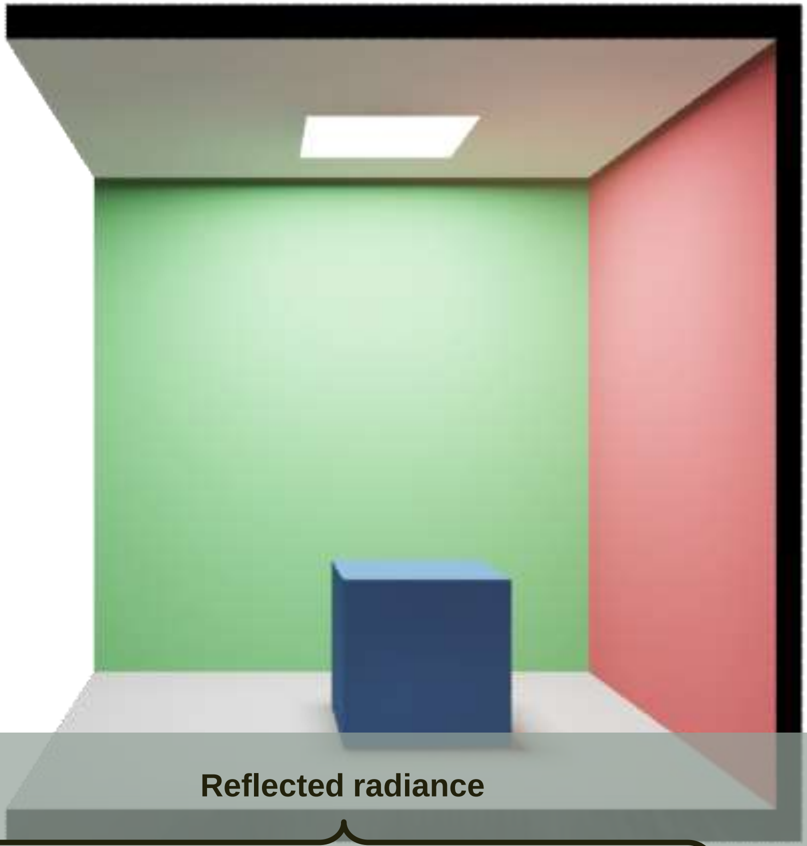


Emitted radiance

Reflected radiance

$$L_o(\mathbf{p},\omega) = L_e(\mathbf{p},\omega) + \int_{\Omega} f(\mathbf{p},\omega,\omega')L_i(\mathbf{p},\omega')\cos(\mathbf{n},\omega')\,d\omega'$$

Monte Carlo Integration

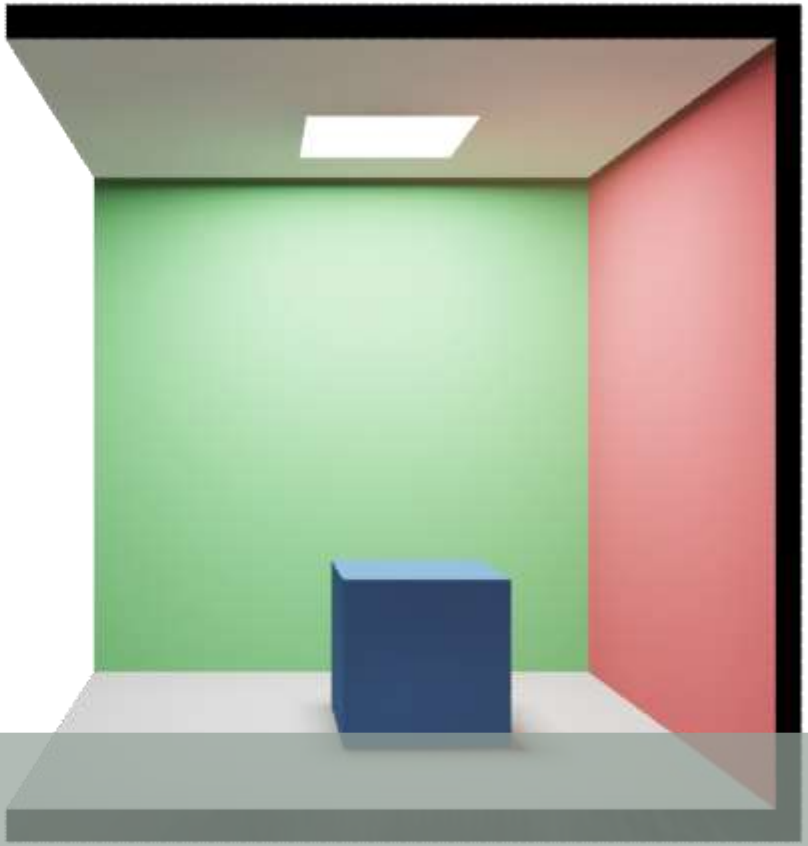


Emitted radiance

Reflected radiance

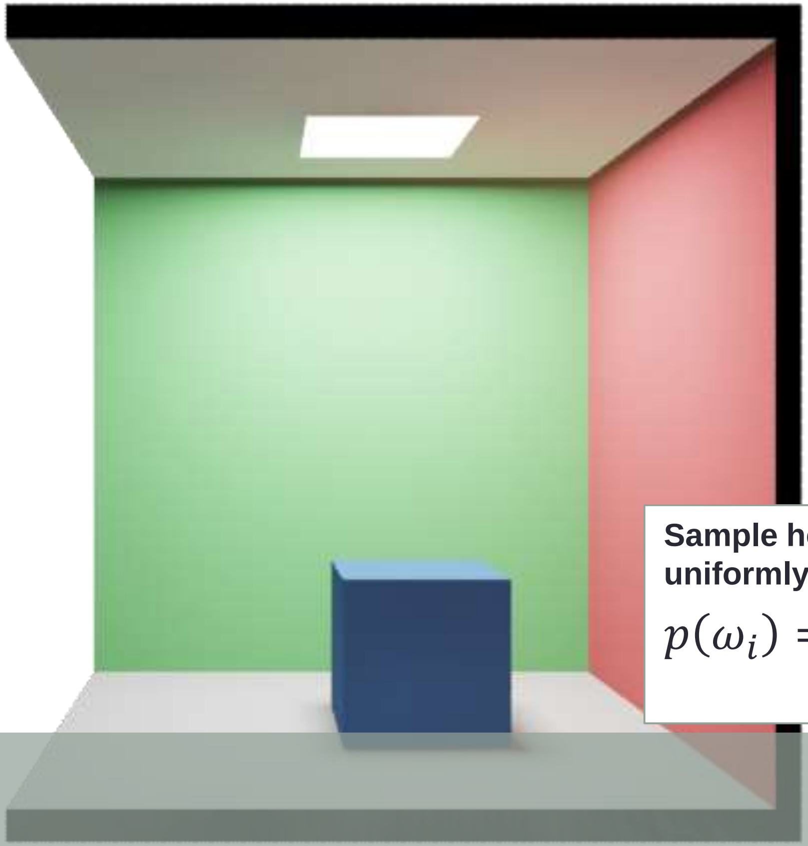
$$L_o(\mathbf{p},\omega) \approx L_e(\mathbf{p},\omega) + \frac{1}{N}\sum_{i=0}^{N} \frac{f(\mathbf{p},\omega,\omega_i)L_i(\mathbf{p},\omega_i)\cos(\mathbf{n},\omega_i)}{p(\omega_i)}$$

Monte Carlo Integration



$$L_o(\mathbf{p},\omega) = E\left[L_e(\mathbf{p},\omega) + \frac{1}{N}\sum_{i=0}^{N}\frac{f(\mathbf{p},\omega,\omega_i)L_i(\mathbf{p},\omega_i)\cos(\mathbf{n},\omega_i)}{p(\omega_i)}\right]$$

Monte Carlo Integration

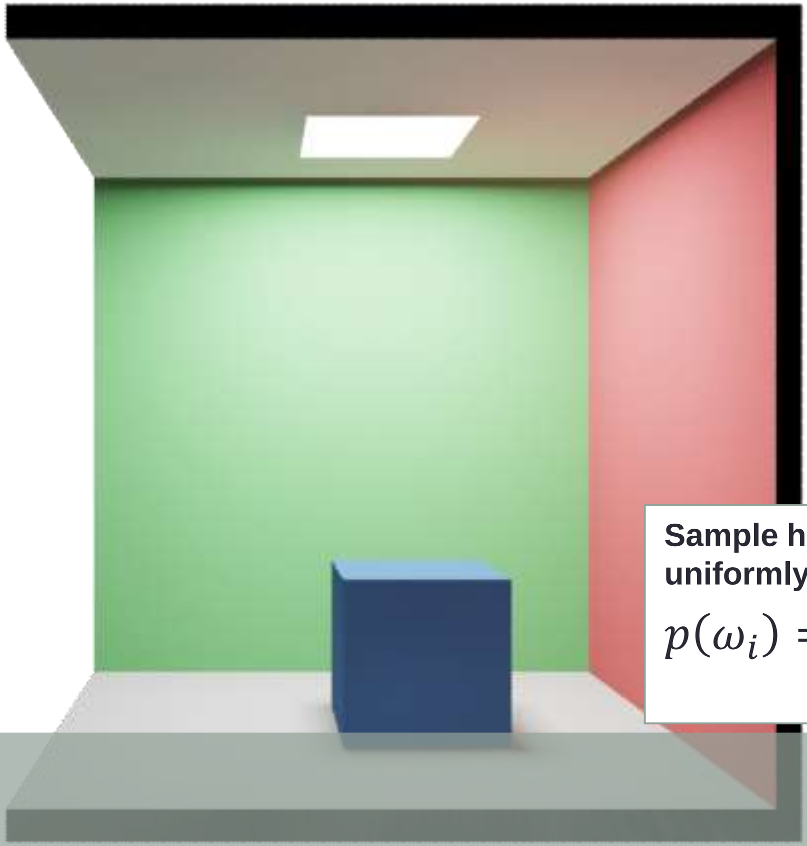


Sample hemisphere uniformly :

$$p(\omega_i) = \frac{1}{2\pi}$$

$$L_o(\boldsymbol{p},\omega) = E\left[L_e(\boldsymbol{p},\omega) + \frac{1}{1}\sum_{i=0}^{1}\frac{f(\boldsymbol{p},\omega,\omega_i)L_i(\boldsymbol{p},\omega_i)\cos(\boldsymbol{n},\omega_i)}{p(\omega_i)}\right]$$

Monte Carlo Integration

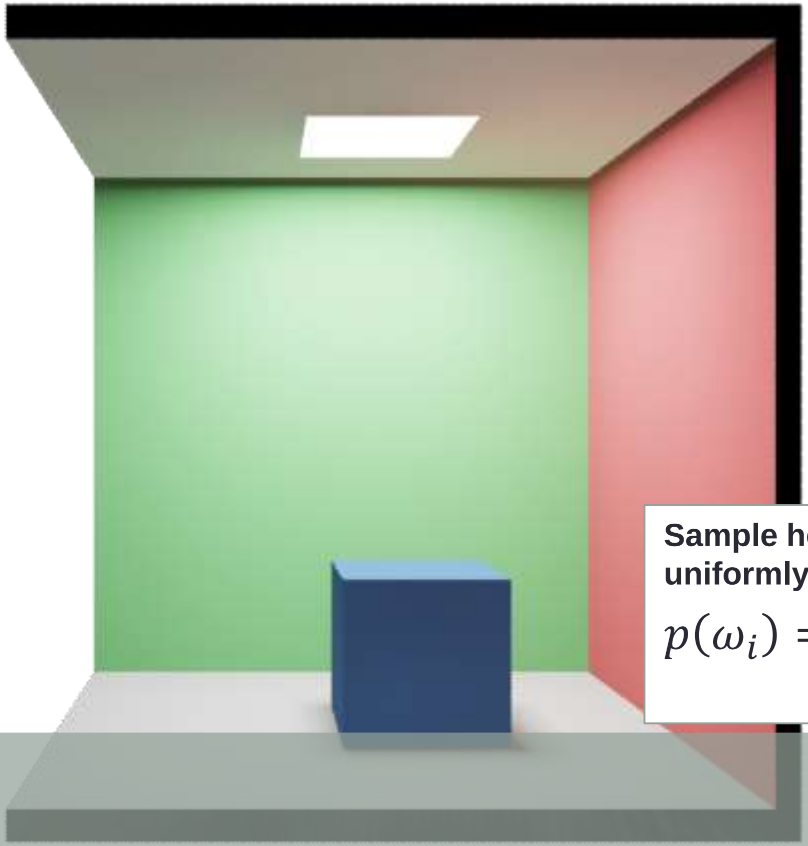


Sample hemisphere uniformly :

$$p(\omega_i) = \frac{1}{2\pi}$$

$$L_o(\mathbf{p},\omega) = E\left[L_e(\mathbf{p},\omega) + \frac{f(\mathbf{p},\omega,\omega_i)L_i(\mathbf{p},\omega_i)\cos(\mathbf{n},\omega_i)}{p(\omega_i)}\right]$$

Monte Carlo Integration

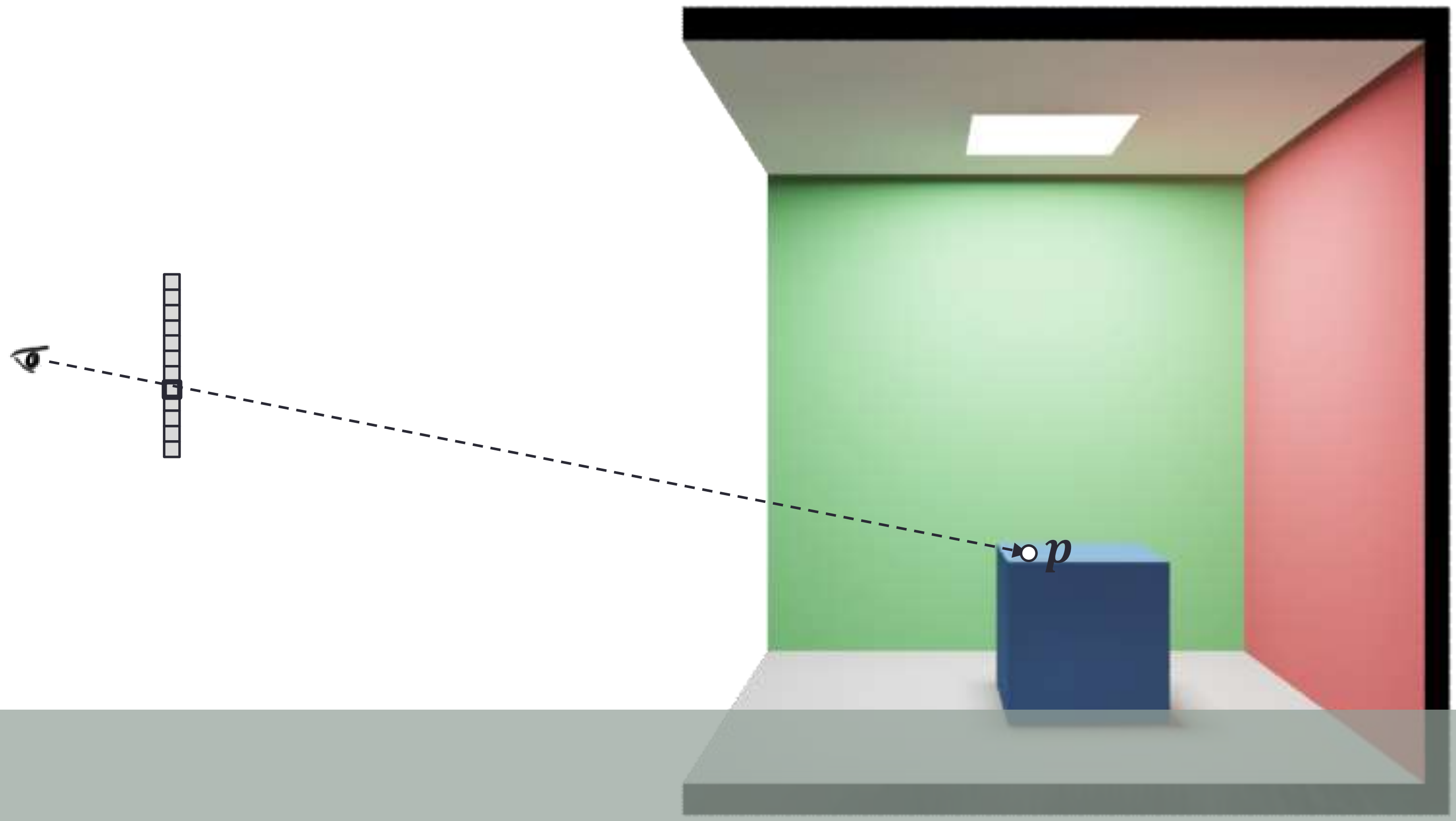


Sample hemisphere uniformly :

$$p(\omega_i) = \frac{1}{2\pi}$$

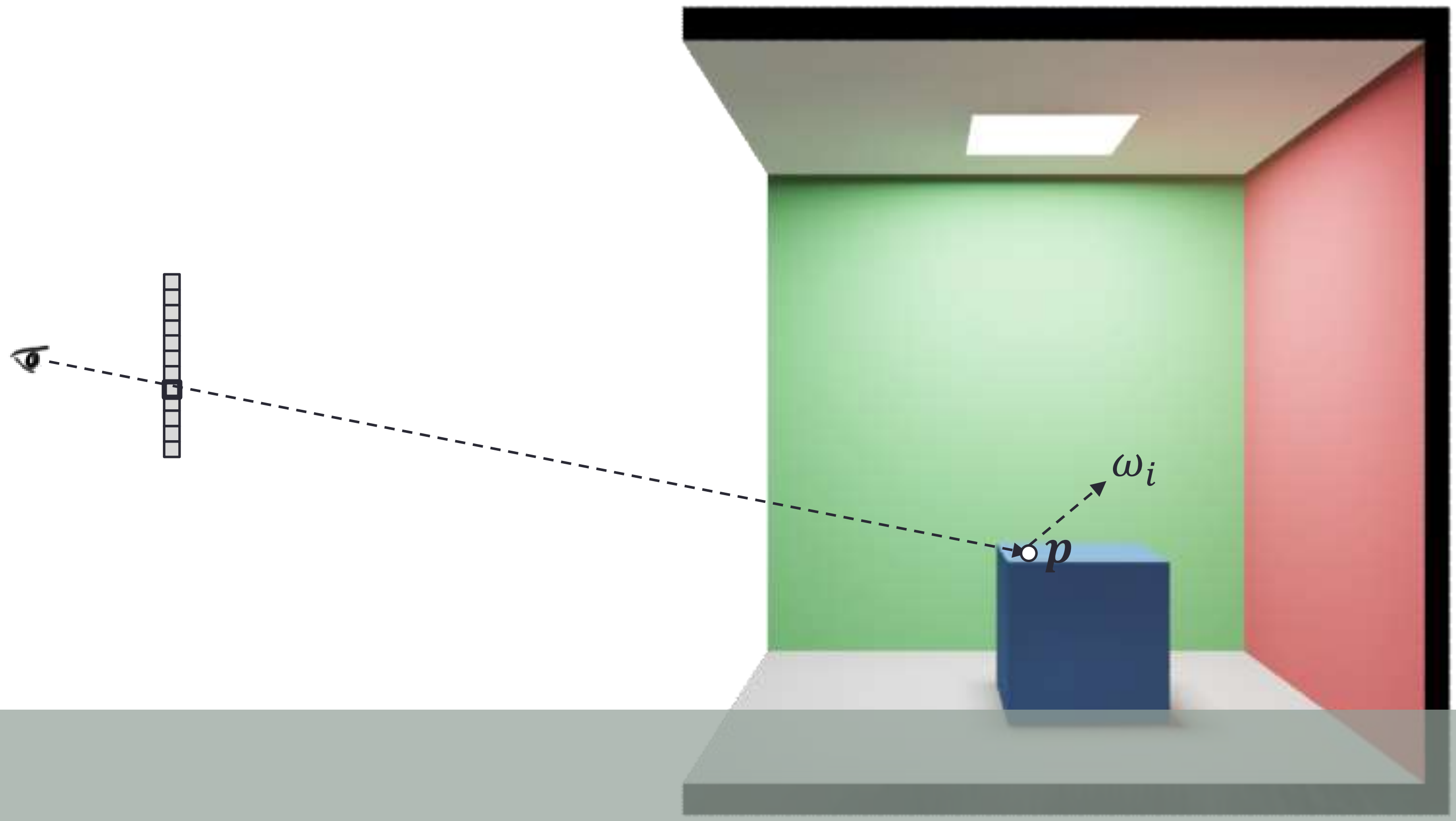
$$L_o(\boldsymbol{p}, \omega) = E[L_e(\boldsymbol{p}, \omega) + 2\pi f(\boldsymbol{p}, \omega, \omega_i) L_i(\boldsymbol{p}, \omega_i) \cos(\boldsymbol{n}, \omega_i)]$$

Naive Pathtracing



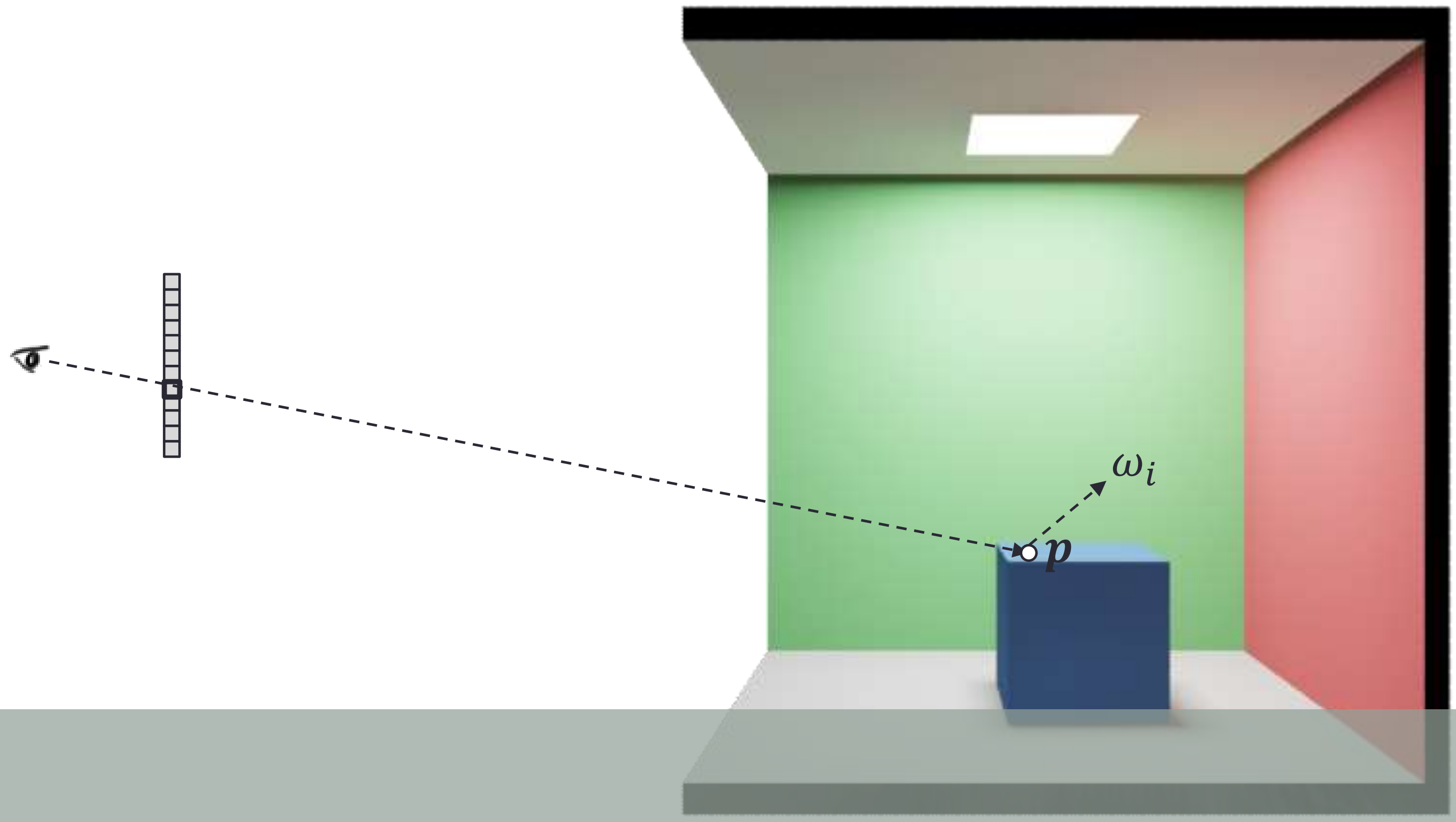
$$L_o(\mathbf{p},\omega) \approx L_e(\mathbf{p},\omega) + 2\pi f(\mathbf{p},\omega,\omega_i) L_i(\mathbf{p},\omega_i)\cos(\mathbf{n},\omega_i)$$

Naive Pathtracing



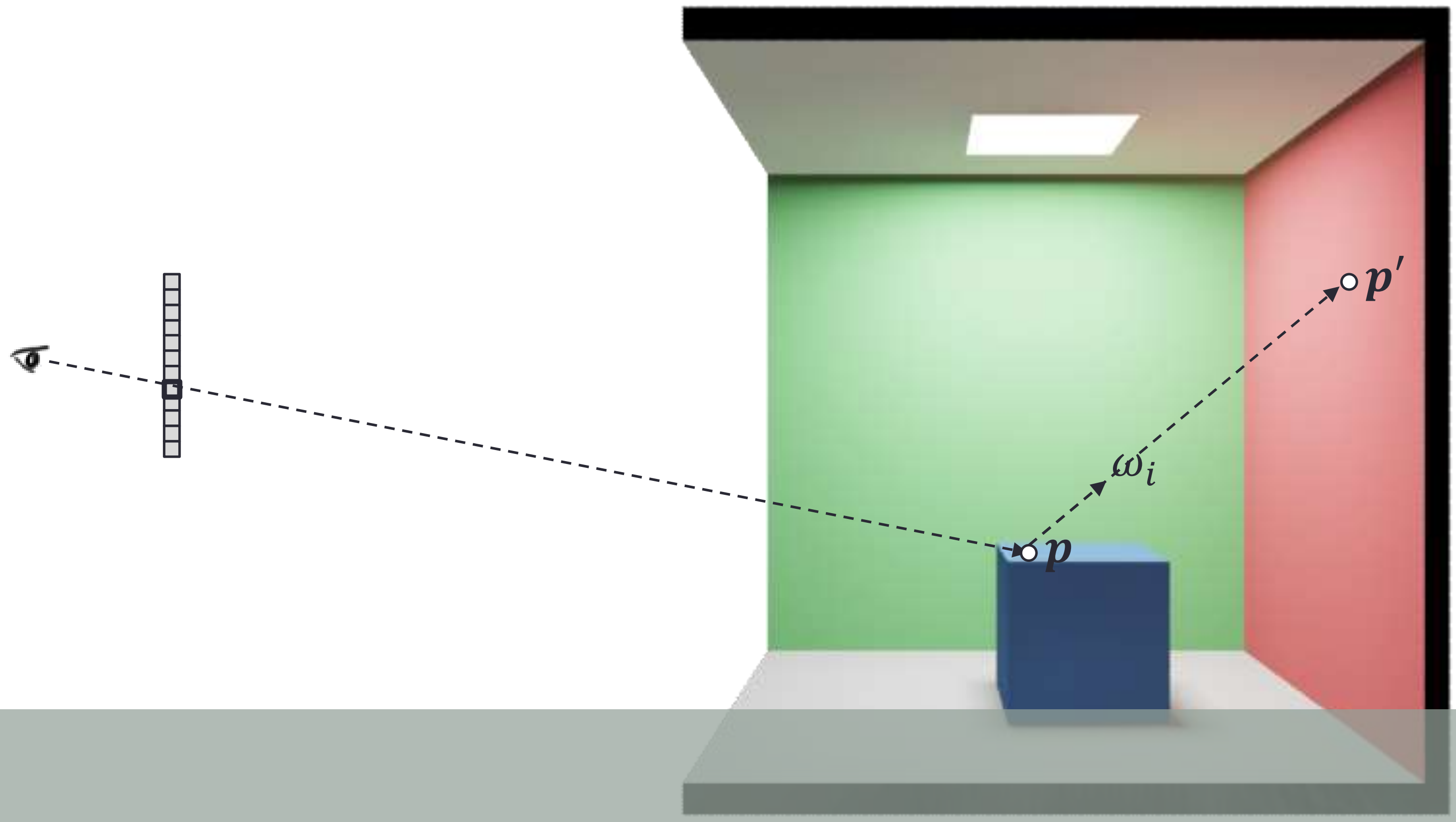
$$L_o(\boldsymbol{p},\omega) \approx L_e(\boldsymbol{p},\omega) + 2\pi f(\boldsymbol{p},\omega,\omega_i) L_i(\boldsymbol{p},\omega_i)\cos(\boldsymbol{n},\omega_i)$$

Naive Pathtracing



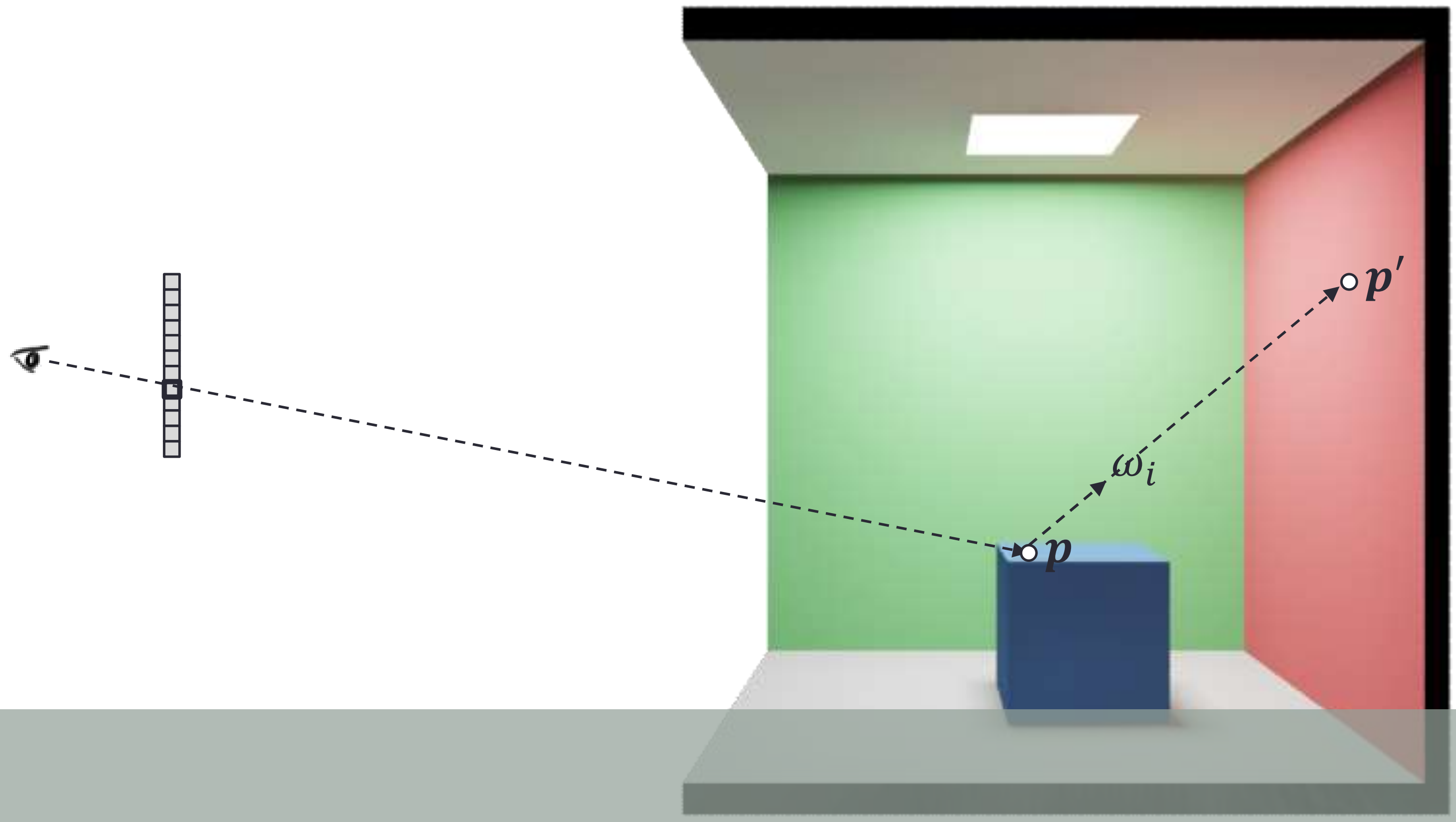
$$L_o(\boldsymbol{p}, \omega) \approx 0 + 2\pi f(\boldsymbol{p}, \omega, \omega_i) L_i(\boldsymbol{p}, \omega_i) \cos(\boldsymbol{n}, \omega_i)$$

Naive Pathtracing



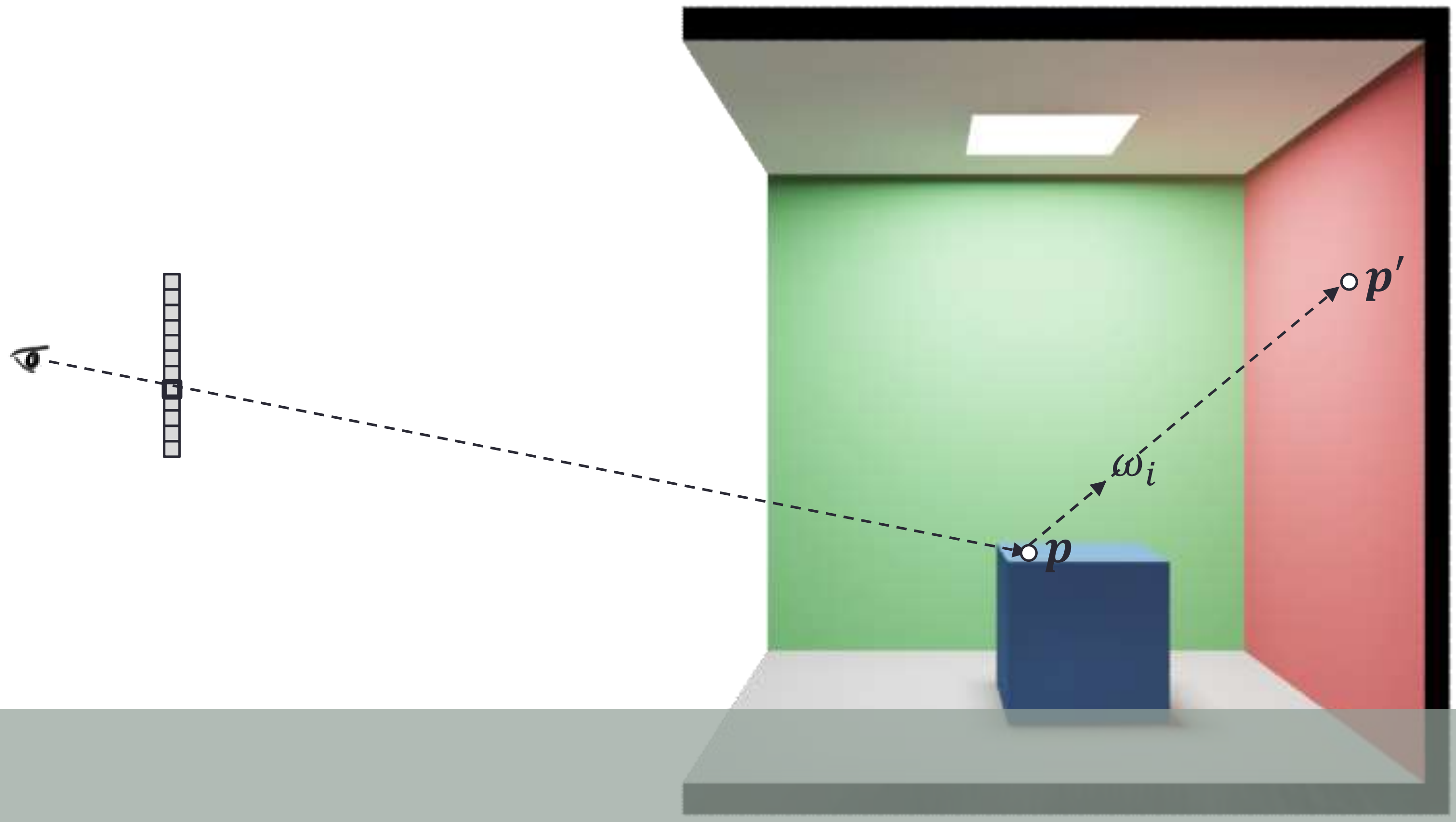
$$L_o(\boldsymbol{p}, \omega) \approx 0 + 2\pi f(\boldsymbol{p}, \omega, \omega_i) L_o(\boldsymbol{p}', -\omega_i) \cos(\boldsymbol{n}, \omega_i)$$

Naive Pathtracing



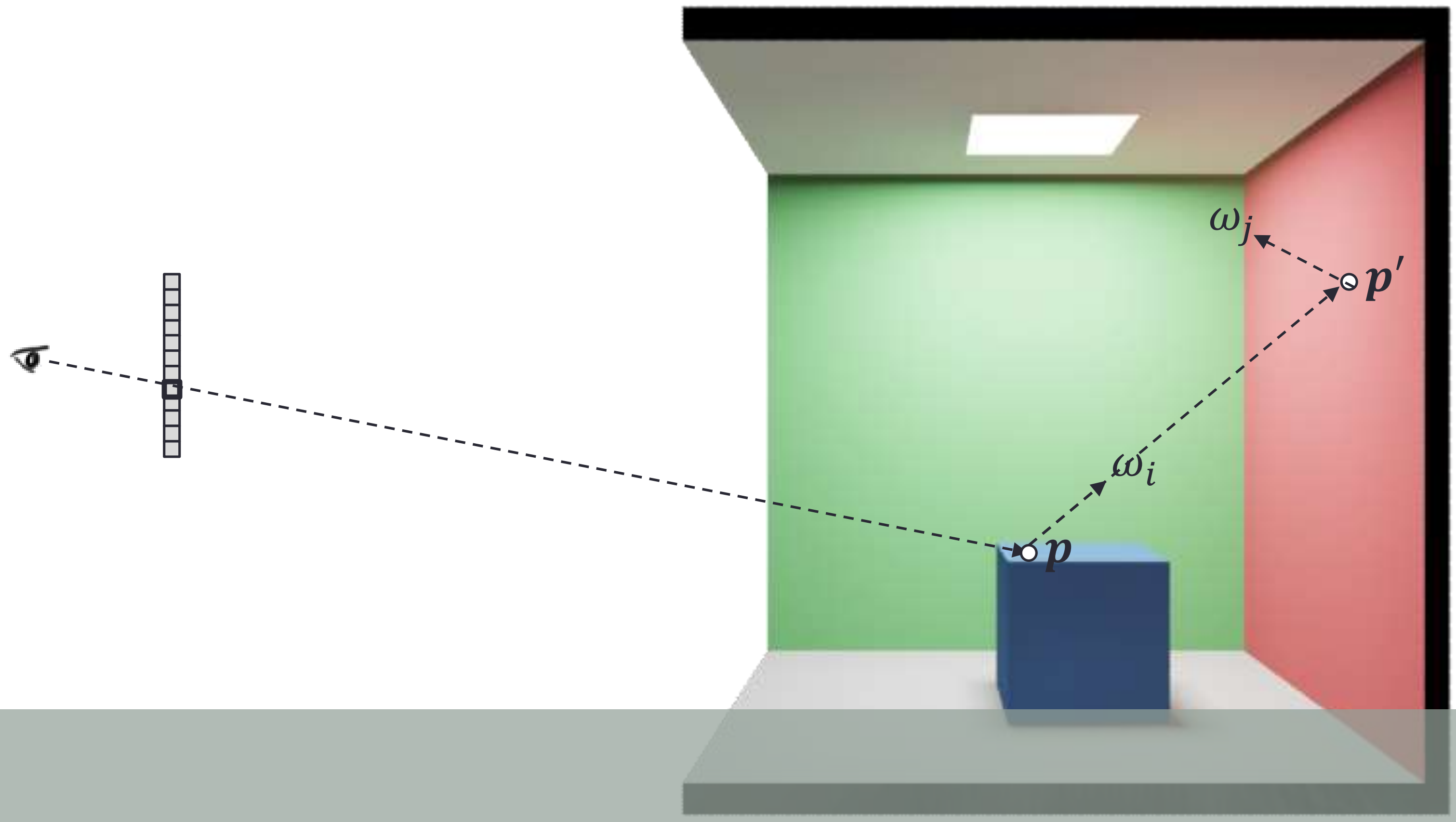
$$L_o(\mathbf{p},\omega) \approx 0 + 2\pi f(\mathbf{p},\omega,\omega_i)L_i(\mathbf{p},\omega_i)\cos(\mathbf{n},\omega_i)$$

Naive Pathtracing



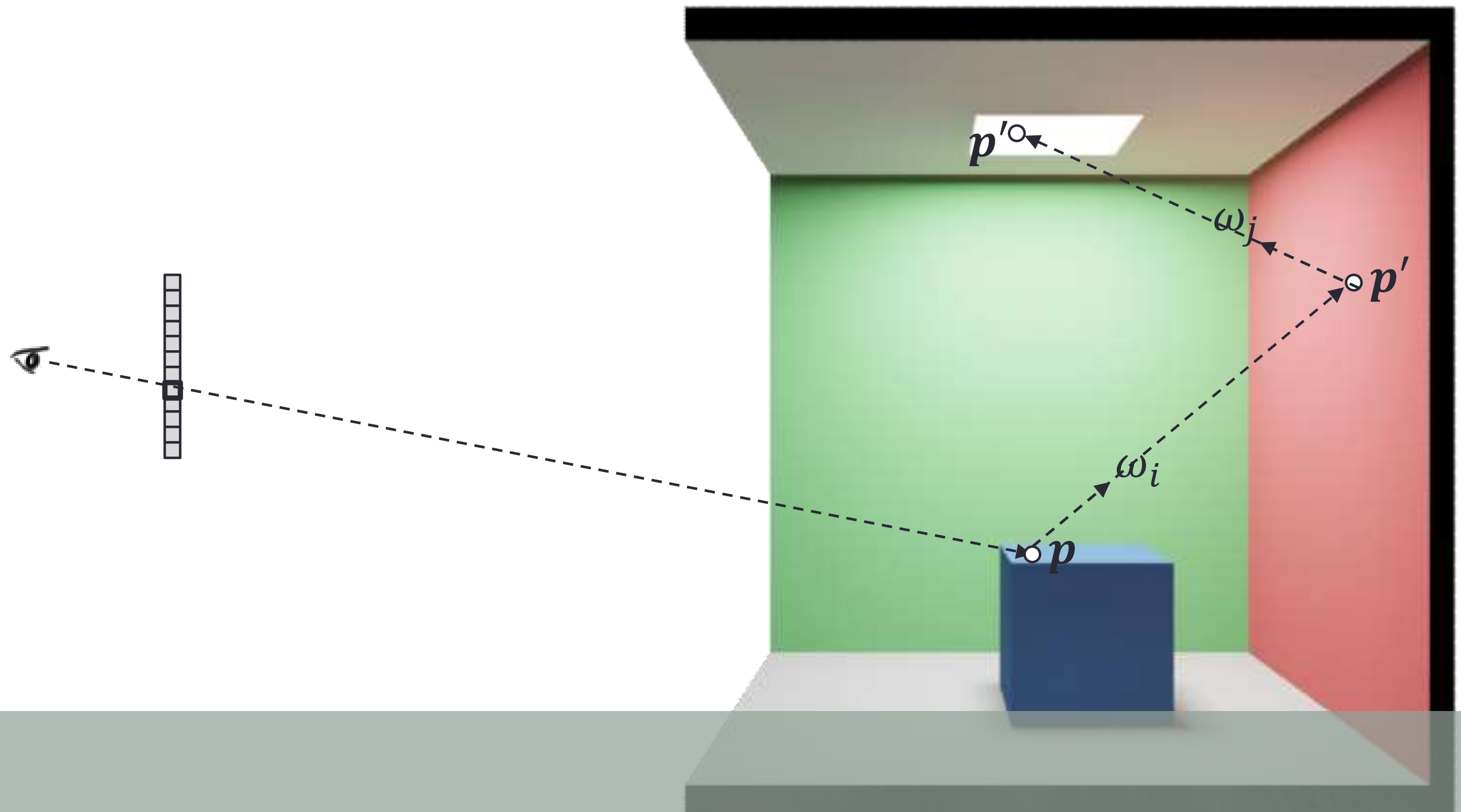
$$L_o(\boldsymbol{p}, \omega) \approx 0 + 2\pi f(\boldsymbol{p}, \omega, \omega_i)\left[L_e(\boldsymbol{p}', -\omega_i) + 2\pi f(\boldsymbol{p}', -\omega_i, \omega_j) L_i(\boldsymbol{p}', \omega_j) \cos(\boldsymbol{n}', \omega_j)\right] \cos(\boldsymbol{n}, \omega_i)$$

Naive Pathtracing



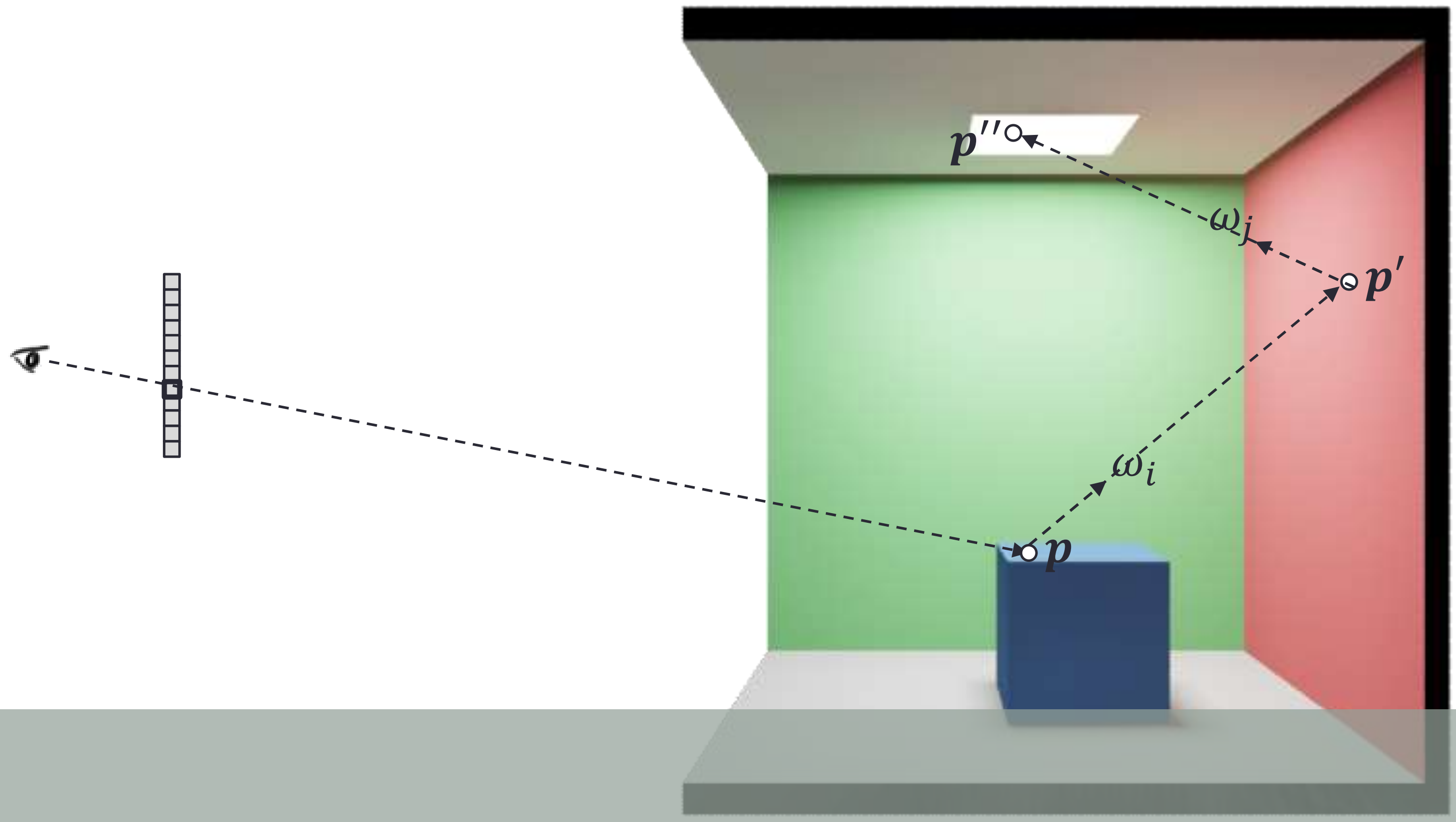
$$L_o(\boldsymbol{p},\omega) \approx 0 + 2\pi f(\boldsymbol{p},\omega,\omega_i)\left[0 + 2\pi f\left(\boldsymbol{p}',-\omega_i,\omega_j\right)L_i\left(\boldsymbol{p}',\omega_j\right)\cos\left(\boldsymbol{n}',\omega_j\right)\right]\cos(\boldsymbol{n},\omega_i)$$

Naive Pathtracing



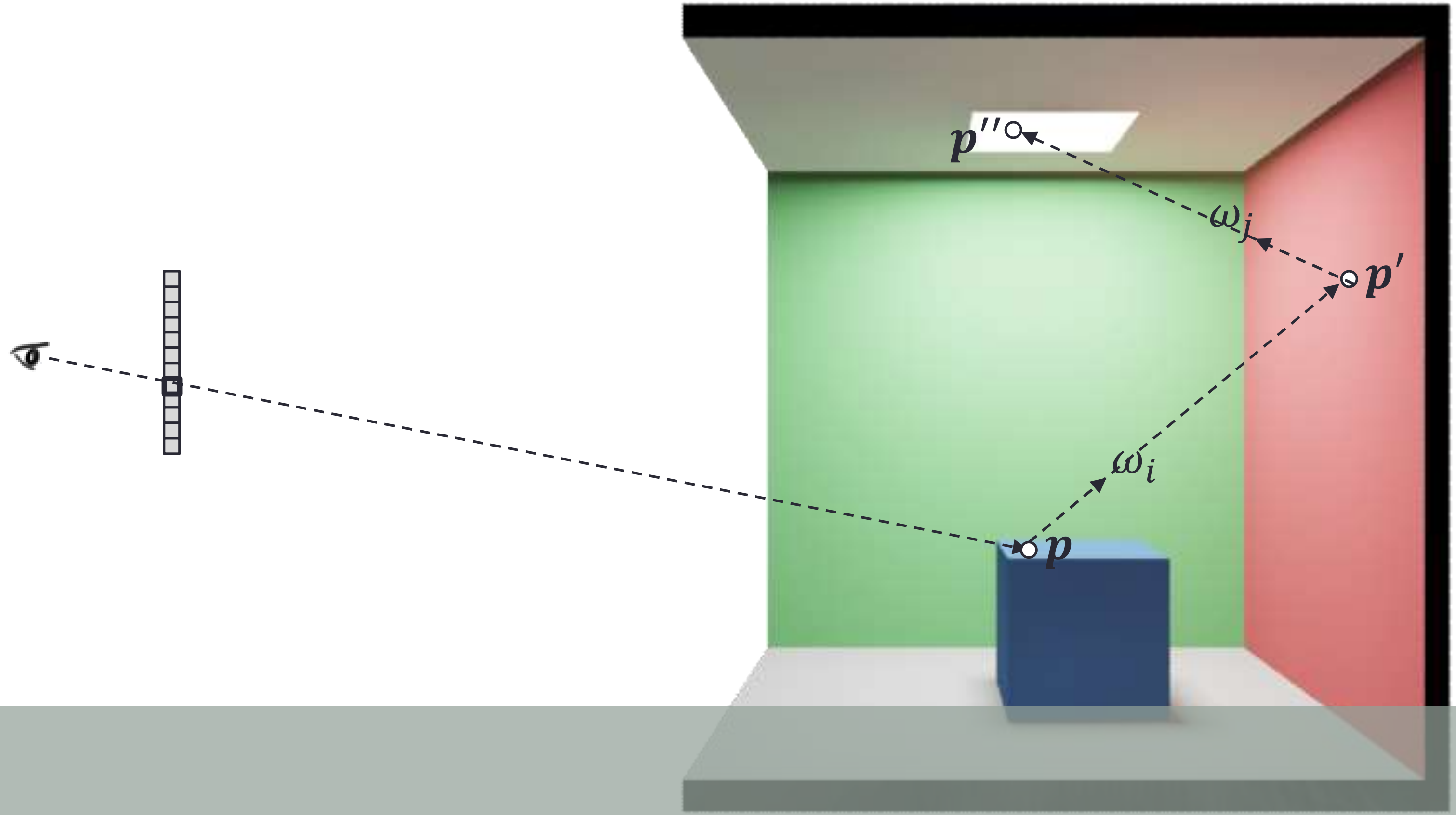
$$L_o(\mathbf{p},\omega) \approx 0 + 2\pi f(\mathbf{p},\omega,\omega_i)\left[0 + 2\pi f(\mathbf{p}',-\omega_i,\omega_j)L_i(\mathbf{p}',\omega_j)\cos(\mathbf{n}',\omega_j)\right]\cos(\mathbf{n},\omega_i)$$

Naive Pathtracing



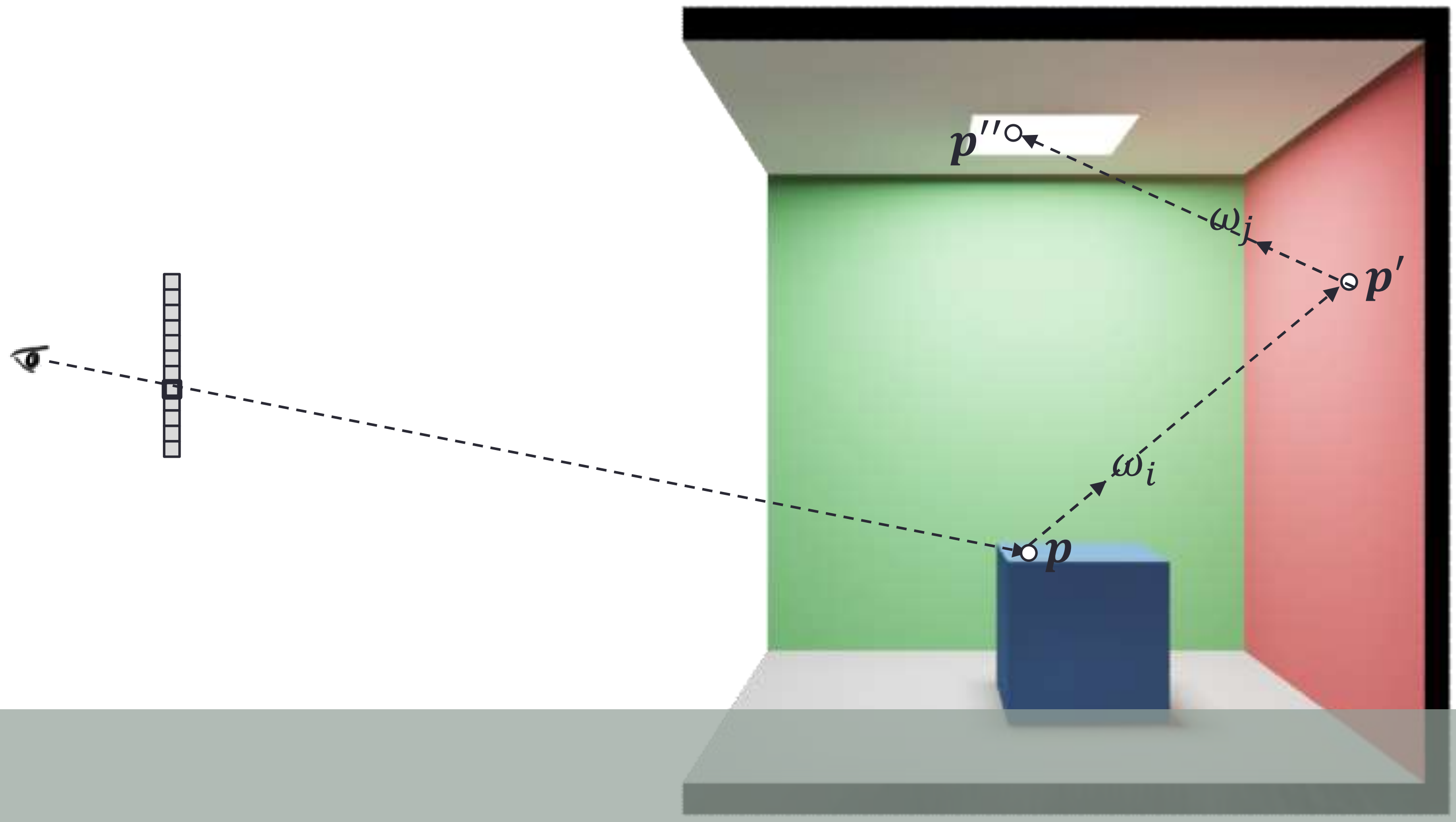
$$L_o(\boldsymbol{p},\omega) \approx 0 + 2\pi f(\boldsymbol{p},\omega,\omega_i)\left[0 + 2\pi f(\boldsymbol{p}',-\omega_i,\omega_j) L_o(\boldsymbol{p}'',-\omega_j)\cos(\boldsymbol{n}',\omega_j)\right]\cos(\boldsymbol{n},\omega_i)$$

Naive Pathtracing



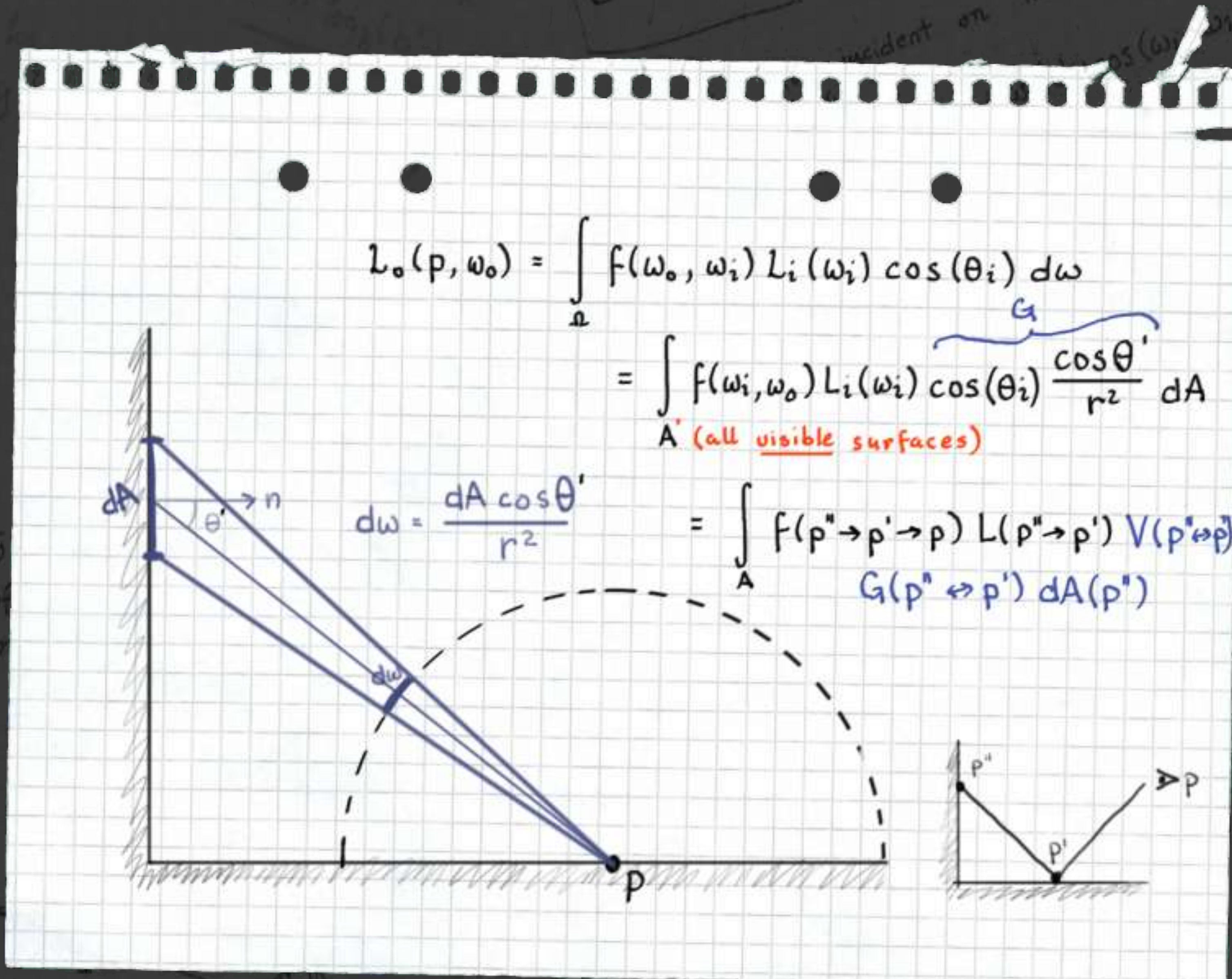
$$L_o(\boldsymbol{p},\omega) \approx 0 + 2\pi f(\boldsymbol{p},\omega,\omega_i)\left[0 + 2\pi f\left(\boldsymbol{p}',-\omega_i,\omega_j\right)L_e(\boldsymbol{p}'',-\omega_i)\cos\left(\boldsymbol{n}',\omega_j\right)\right]\cos(\boldsymbol{n},\omega_i)$$

Naive Pathtracing

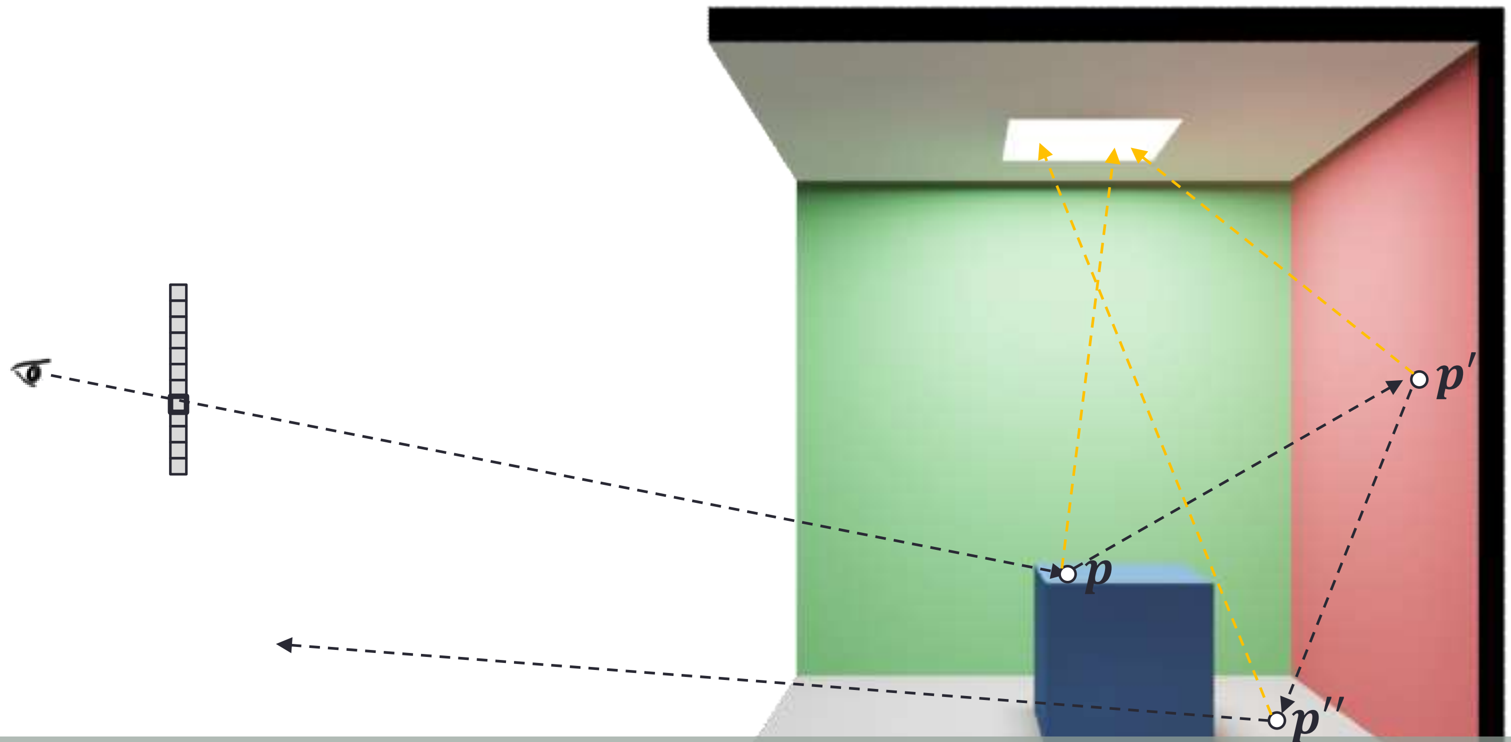


What's so naïve about this?

Surface form of LTE



Separating Direct And Indirect Illumination

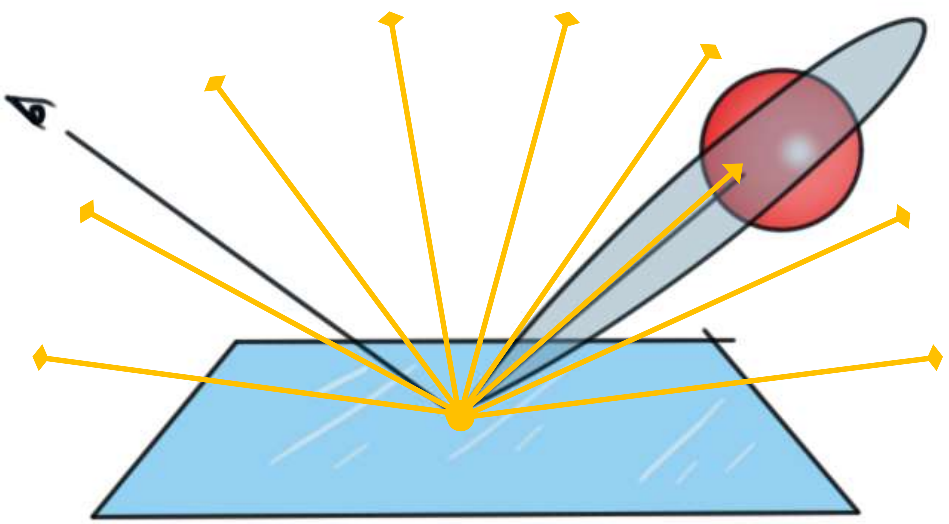


direct illumination — *indirect illumination*

$$L_o(\mathbf{p},\omega) = \underbrace{\int_A f(\mathbf{p},\mathbf{q}\rightarrow\mathbf{p})L_i(\mathbf{p},\mathbf{q}\rightarrow\mathbf{p})G(\mathbf{p},\mathbf{q})V(\mathbf{p},\mathbf{q})dA}_{\text{direct illumination}} + \underbrace{\int_\Omega f(\mathbf{p},\omega,\omega')L_i(\mathbf{p},\omega')\cos(\mathbf{n},\omega')d\omega'}_{\text{indirect illumination}}$$

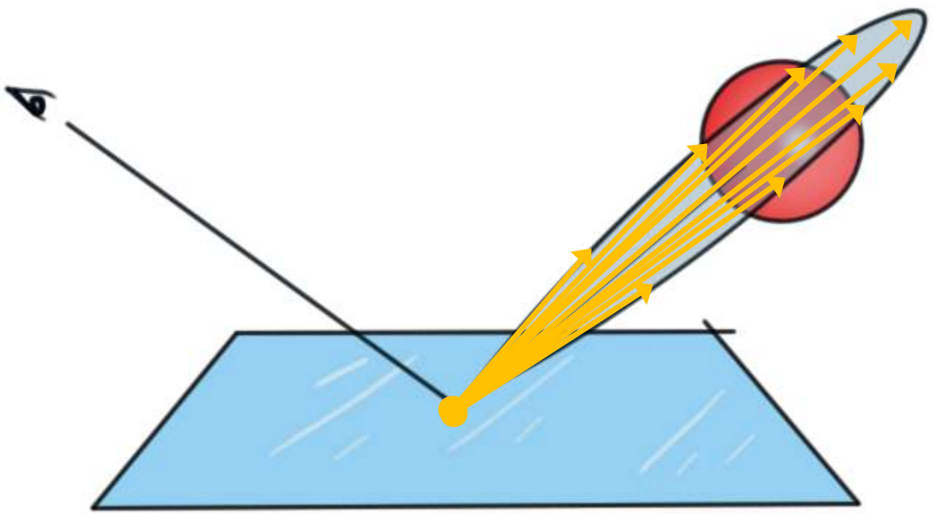
Importance Sampling

- So far we have sampled incoming light uniformly over the hemisphere. Why is that a bad idea?
- We want to shoot more samples where the function we are integrating is high!



Importance Sampling

- So far we have sampled incoming light uniformly over the hemisphere. Why is that a bad idea?
- We want to shoot more samples where the function we are integrating is high!



Probability Density Function

Not logged in Talk Contributions Create account Log in

WIKIPEDIA The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information

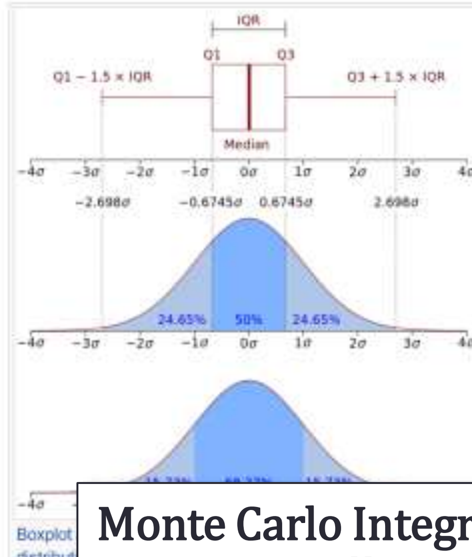
Article Talk Read Edit View history Search

Probability density function

From Wikipedia, the free encyclopedia

In probability theory, a **probability density function** (**PDF**), or **density** of a continuous random variable, is a function that describes the relative likelihood for this random variable to take on a given value. The probability of the random variable falling within a particular range of values is given by the integral of this variable's density over that range—that is, it is given by the area under the density function but above the horizontal axis and between the lowest and greatest values of the range. The probability density function is nonnegative everywhere, and its integral over the entire space is equal to one.

The terms "*probability distribution function*"[2] and "*probability function*"[3] have also sometimes been used to denote the probability density function. However, this use is not standard among probabilists and statisticians. In other sources, "probability distribution function" may be used when the probability distribution is defined as a function over general sets of values, or it may refer to the cumulative distribution function, or it may be a probability mass function (PMF) rather than the density. Further confusion of terminology exists because *density function* has also been used for what is here called the "probability mass function" (PMF).[4] In general though, the PMF is used in the context of discrete random variables (random variables that take values on a discrete set), while PDF is used in the context of continuous random variables.



Boxplot ... distribut...

$$p(X) = \frac{dP(X)}{dX}$$

$$\int p(X)\,dX = 1$$

Monte Carlo Integration

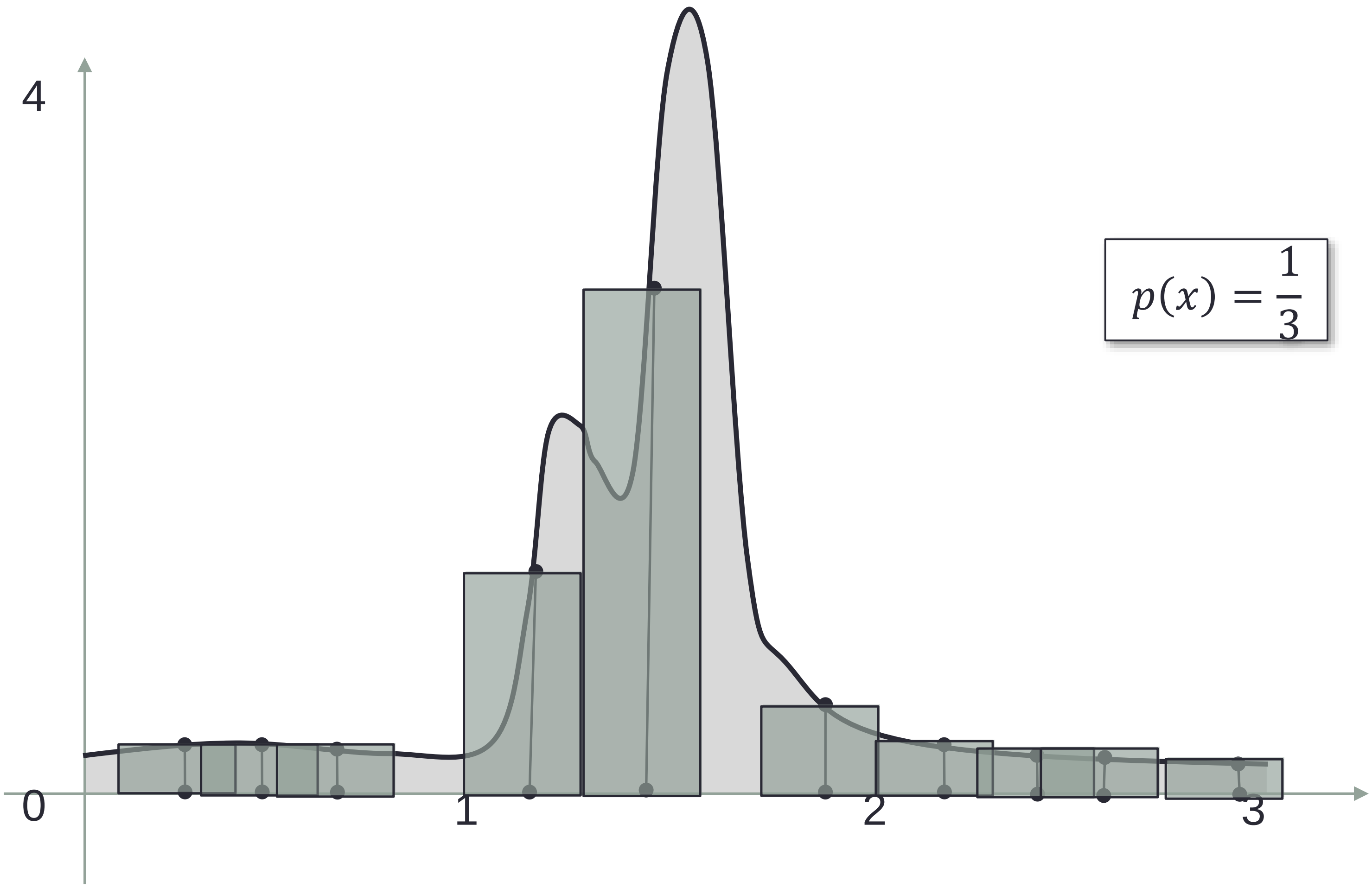
$$F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}$$

Importance Sampling

Monte Carlo Integration

$$F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}$$

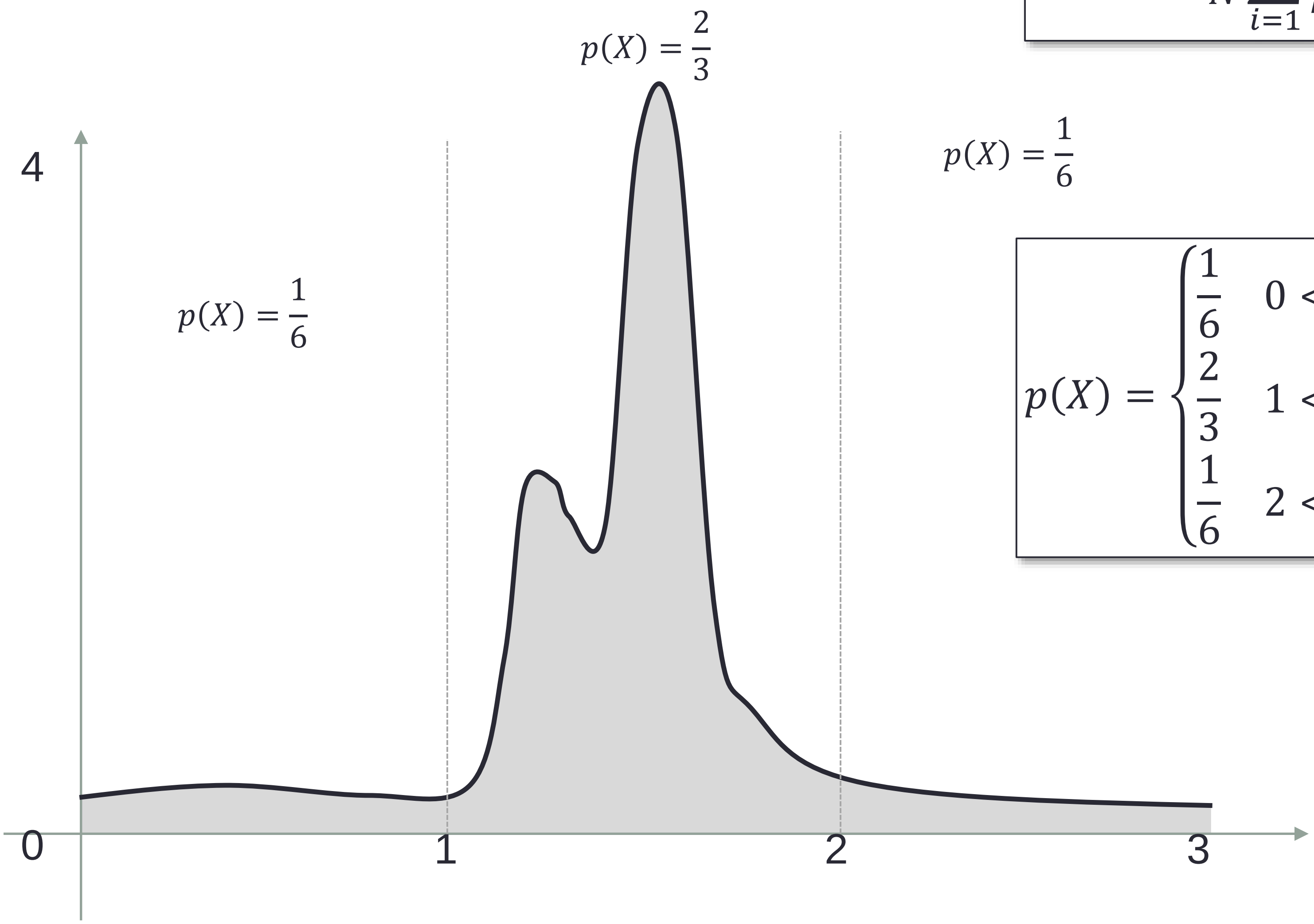
$$p(x) = \frac{1}{3}$$



Importance Sampling

Monte Carlo Integration

$$F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}$$

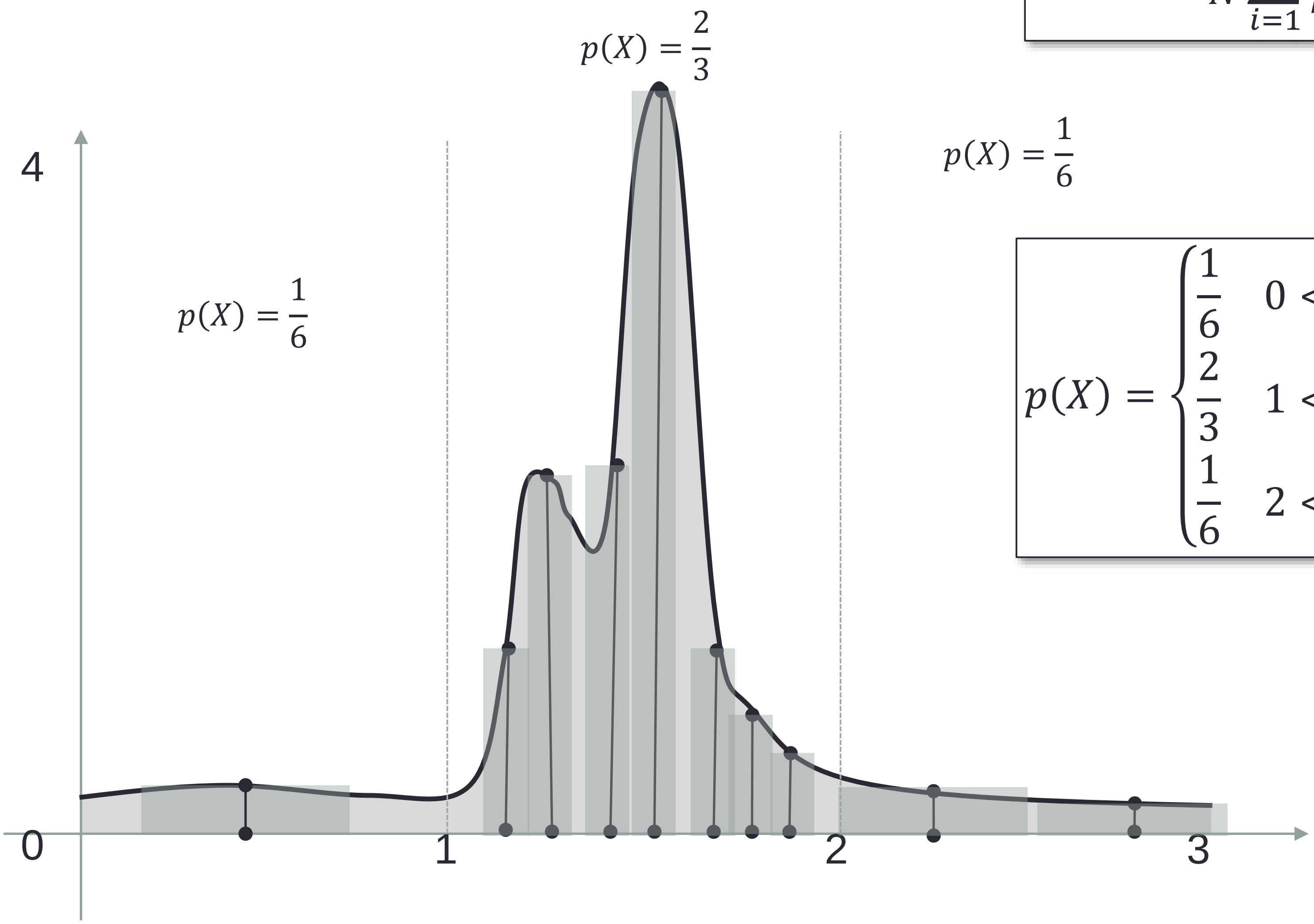


$$p(X) = \begin{cases} \frac{1}{6} & 0 < X < 1 \\ \frac{2}{3} & 1 < X < 2 \\ \frac{1}{6} & 2 < X < 3 \end{cases}$$

Importance Sampling

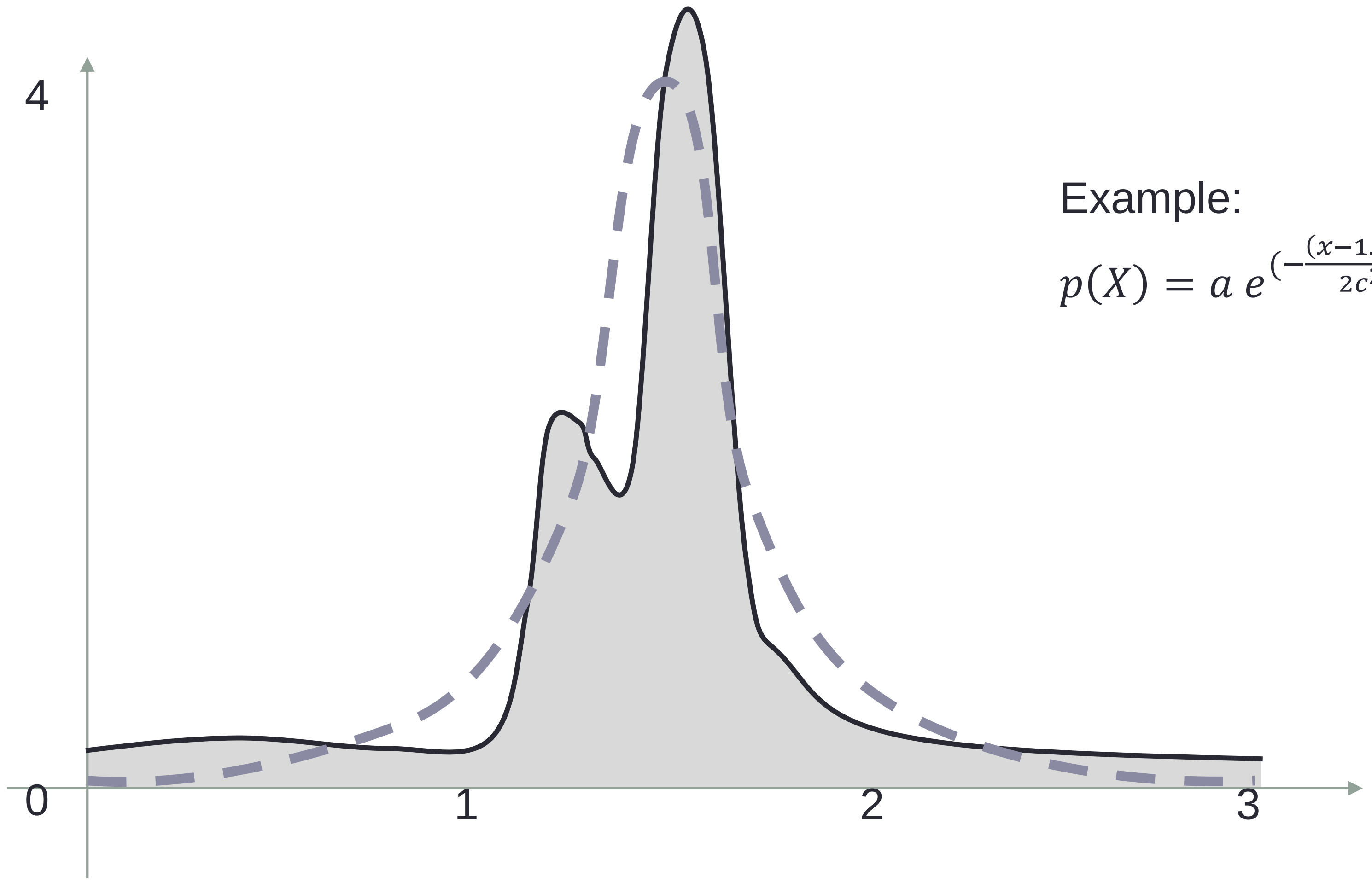
Monte Carlo Integration

$$F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}$$



$$p(X) = \begin{cases} \frac{1}{6} & 0 < X < 1 \\ \frac{2}{3} & 1 < X < 2 \\ \frac{1}{6} & 2 < X < 3 \end{cases}$$

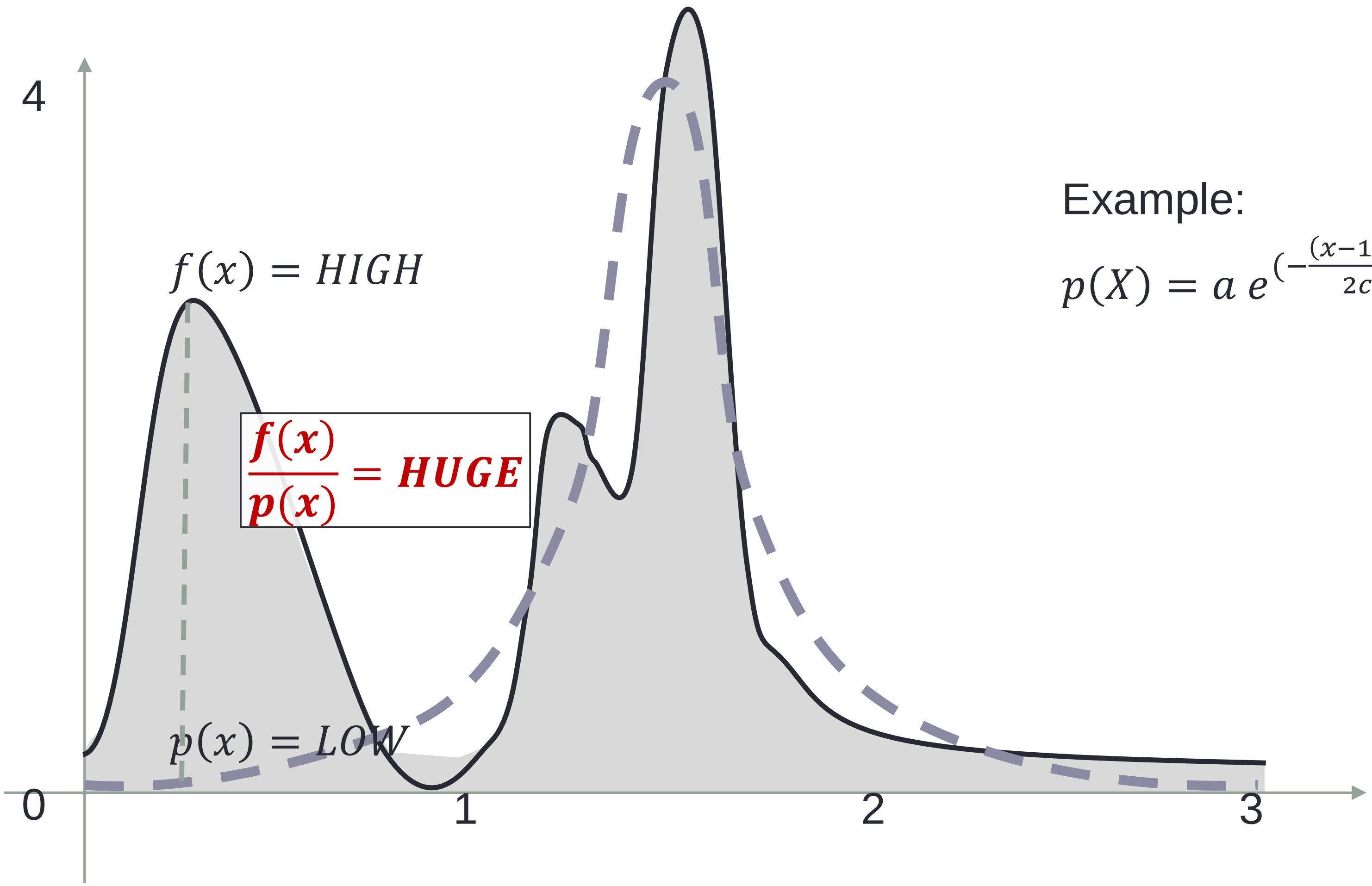
Importance Sampling



Example:

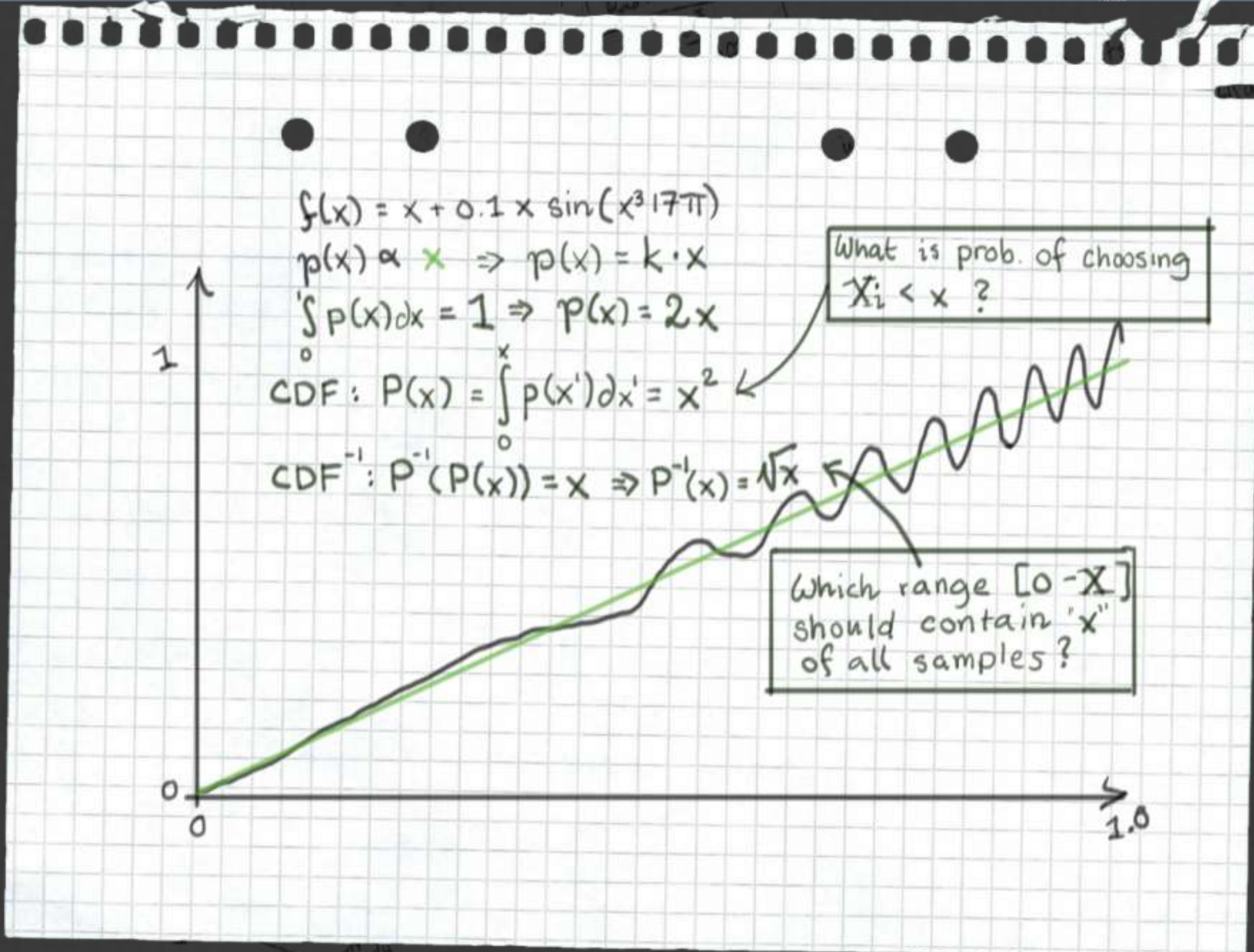
$$p(X) = a\ e^{\left(-\frac{(x-1.5)^2}{2c^2}\right)}$$

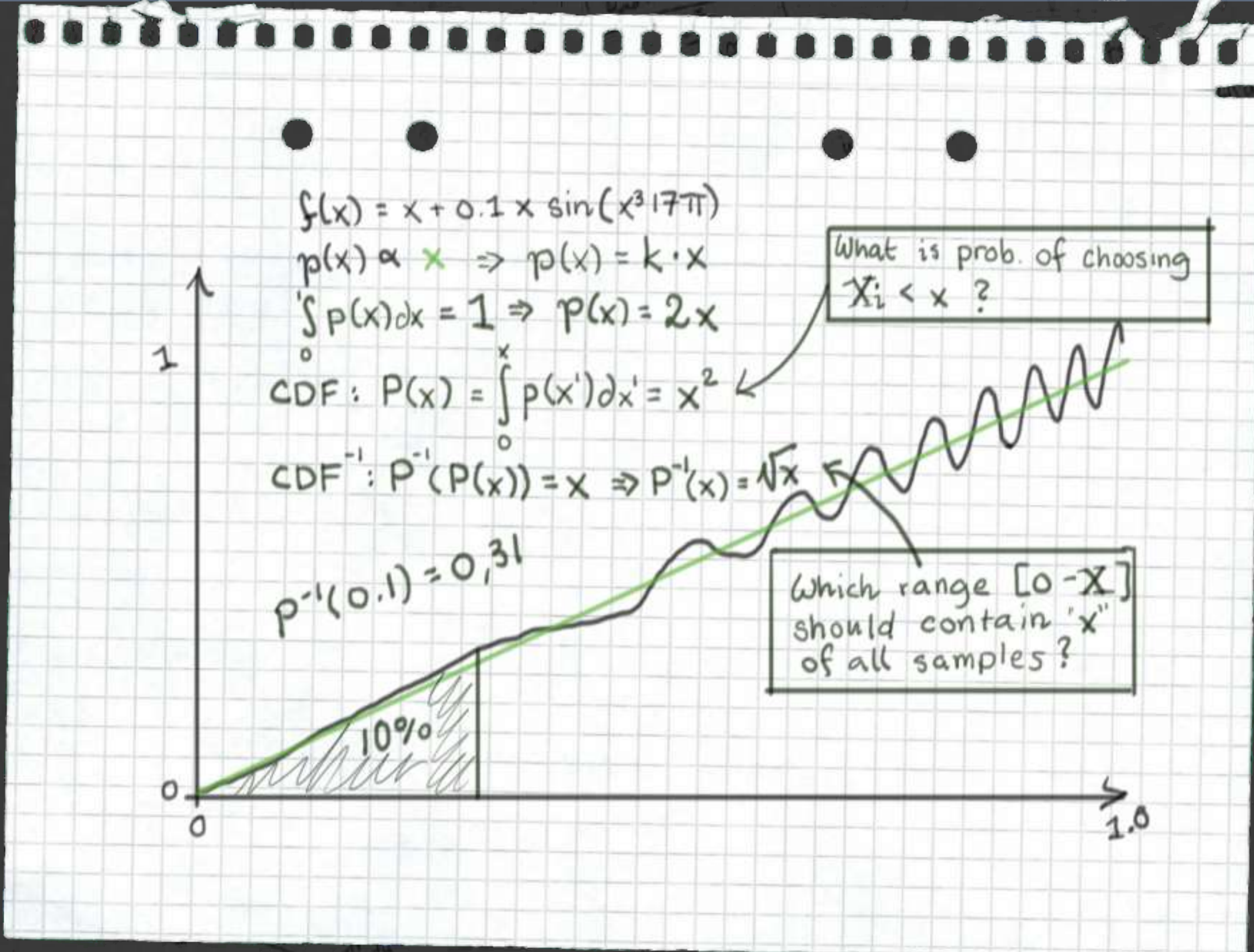
Importance Sampling



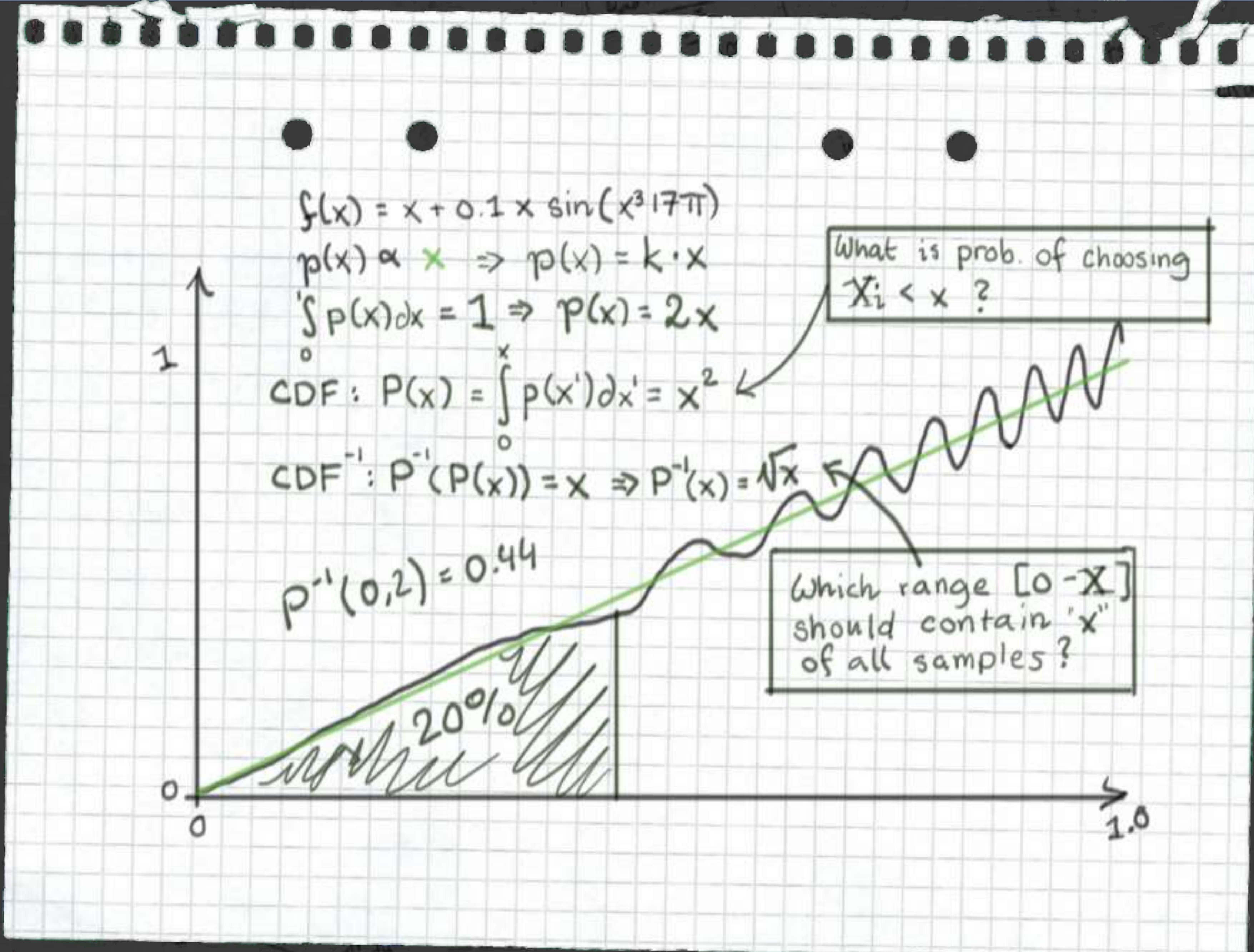
Example:

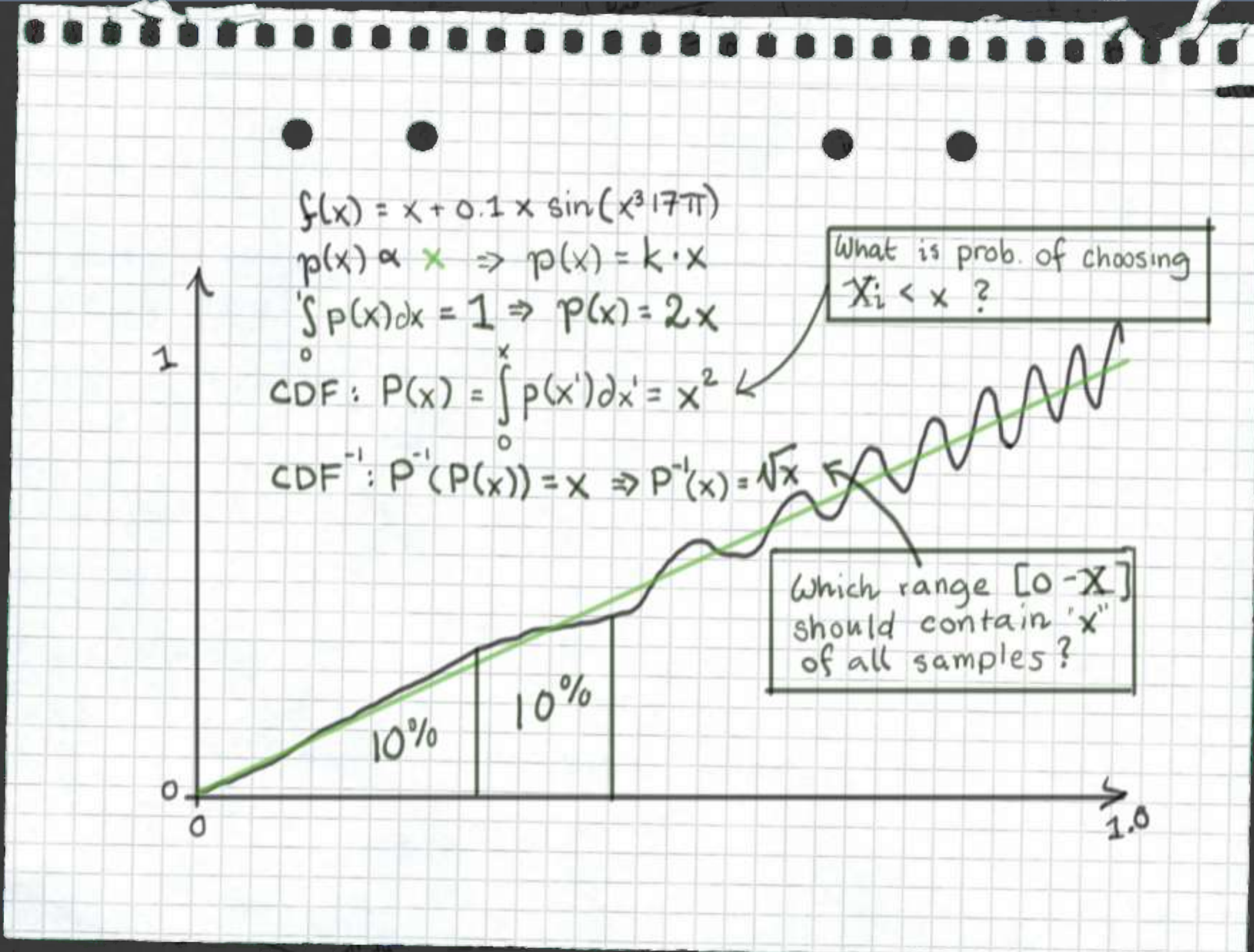
$$p(X) = a\ e^{\left(-\frac{(x-1.5)^2}{2c^2}\right)}$$

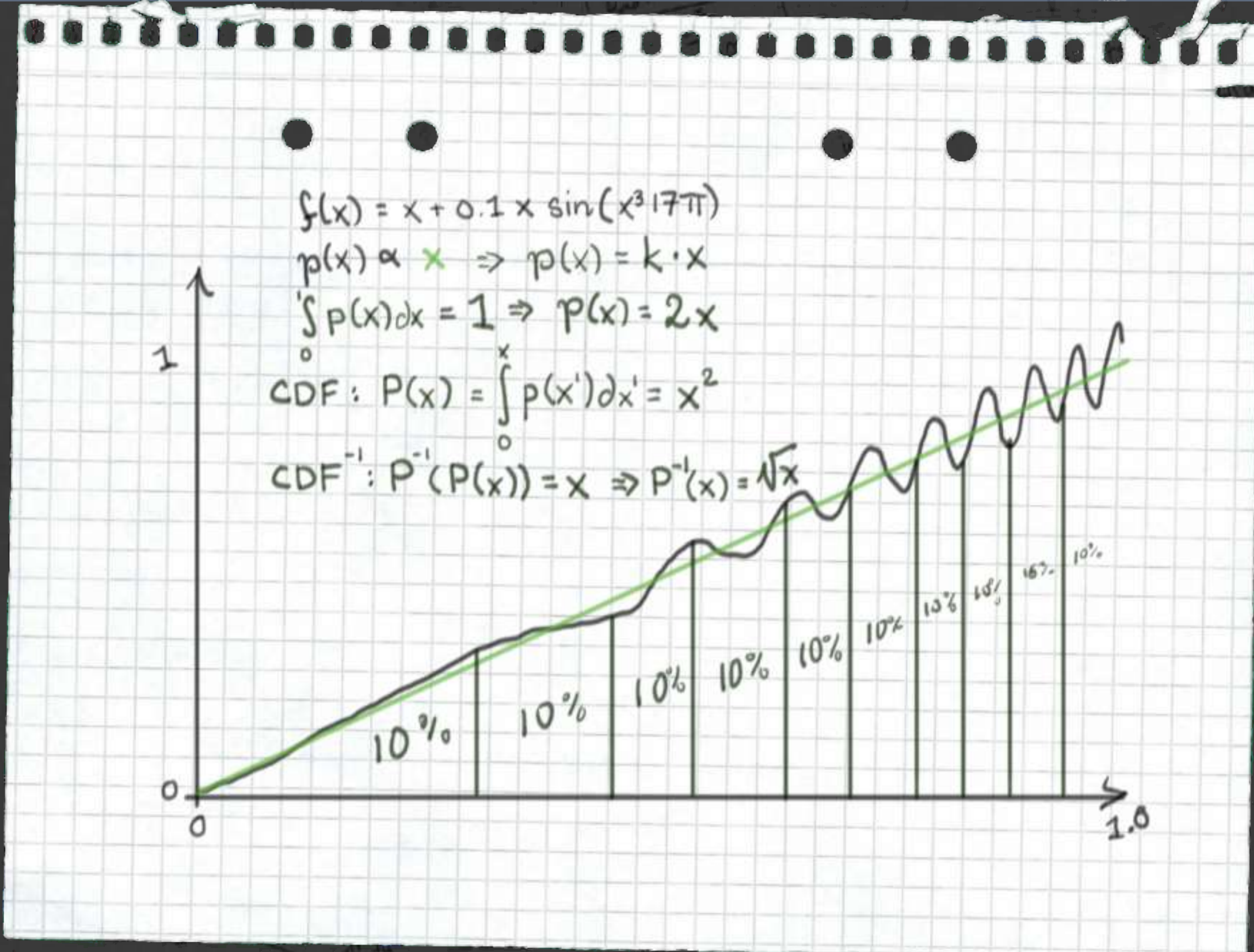

f(x) = x + 0.1 x sin(x³ 17π)
p(x) ∝ x ⇒ p(x) = k · x
∫₀¹ p(x)dx = 1 ⇒ p(x) = 2x
CDF: P(x) = ∫₀ˣ p(x')dx' = x²
CDF⁻¹: P⁻¹(P(x)) = x ⇒ P⁻¹(x) = √x
What is prob. of choosing Xᵢ < x ?
Which range [0 - X] should contain 'x' of all samples?
1
0
0
1.0



f(x) = x + 0.1 x sin(x³ 17π)
p(x) ∝ x ⇒ p(x) = k · x
∫₀¹ p(x)dx = 1 ⇒ p(x) = 2x
CDF: P(x) = ∫₀ˣ p(x')dx' = x²
CDF⁻¹: P⁻¹(P(x)) = x ⇒ P⁻¹(x) = √x
P⁻¹(0.1) ≈ 0,31
What is prob. of choosing Xᵢ < x ?
Which range [0 - X] should contain "x" of all samples?
10%
1
0
0
1.0


f(x) = x + 0.1 x sin(x³ 17π)
p(x) ∝ x ⇒ p(x) = k · x
∫₀¹ p(x)dx = 1 ⇒ p(x) = 2x
CDF: P(x) = ∫₀ˣ p(x')dx' = x²
CDF⁻¹: P⁻¹(P(x)) = x ⇒ P⁻¹(x) = √x
P⁻¹(0,2) = 0.44
20%
What is prob. of choosing Xᵢ < x ?
Which range [0 - X] should contain 'x' of all samples?
1
0
0
1.0

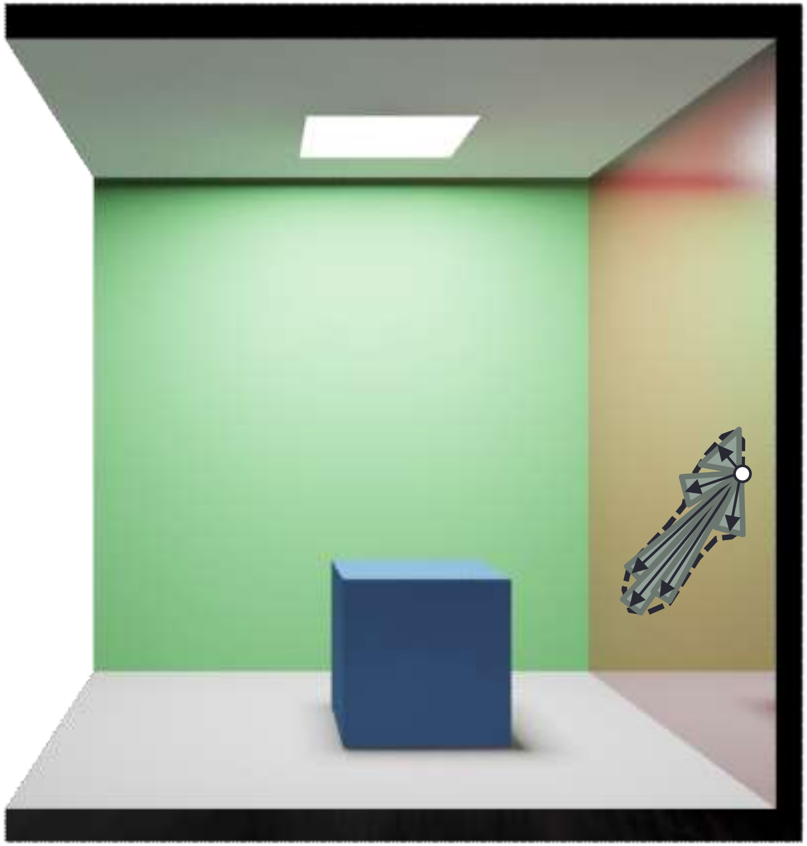

f(x) = x + 0.1 x sin(x³ 17π)
p(x) ∝ x ⇒ p(x) = k · x
∫₀¹ p(x)dx = 1 ⇒ p(x) = 2x
CDF: P(x) = ∫₀ˣ p(x')dx' = x²
CDF⁻¹: P⁻¹(P(x)) = x ⇒ P⁻¹(x) = √x
What is prob. of choosing Xi < x ?
Which range [0 - X] should contain 'x' of all samples?
10%
10%
1
0
0
1.0


f(x) = x + 0.1 x sin(x³ 17π)
p(x) ∝ x ⇒ p(x) = k · x
∫₀¹ p(x)dx = 1 ⇒ p(x) = 2x
CDF: P(x) = ∫₀ˣ p(x')dx' = x²
CDF⁻¹: P⁻¹(P(x)) = x ⇒ P⁻¹(x) = √x
1
0
0
1.0
10%
10%
10%
10%
10%
10%
10%
10%
10%
10%

Importance Sampling

$$L_o(\mathbf{p},\omega) = \int_{\Omega} f(\mathbf{p},\omega,\omega')L_i(\mathbf{p},\omega')\cos(\mathbf{n},\omega')\,d\omega'$$

- When evaluating the Rendering Equation, we do not know the function we want to integrate
  - Since it depends on the incoming light over the hemisphere
  - But we *do* know the BRDF, so we importance sample on that



Importance sampling Blinn MF BRDF

$$f(\omega_o, \omega_i) = \frac{F(\omega_o) G(\omega_h) D(\omega_h)}{4 \cos\theta_o \cos\theta_i}$$

$$D(\omega_h) = ((n+2)/2\pi)(\cos\theta_h)^n$$

Need to sample ω_h (and find ω_o from that)

ϕ_h does not affect D: $\phi_h = 2\pi\xi_2$

PDF must be normalized: $D(\cos\theta_h) = (n+2)(\cos\theta_h)^n$

$$p_h(\cos\theta_h) = k\, D(\cos\theta_h)$$

$$\left[\int k\, D(\cos\theta_h) = 1 \right]$$

$$p_h(\cos\theta_h) = (n+1)\cos^n\theta_h$$

Power distribution in $\cos\theta_h$:

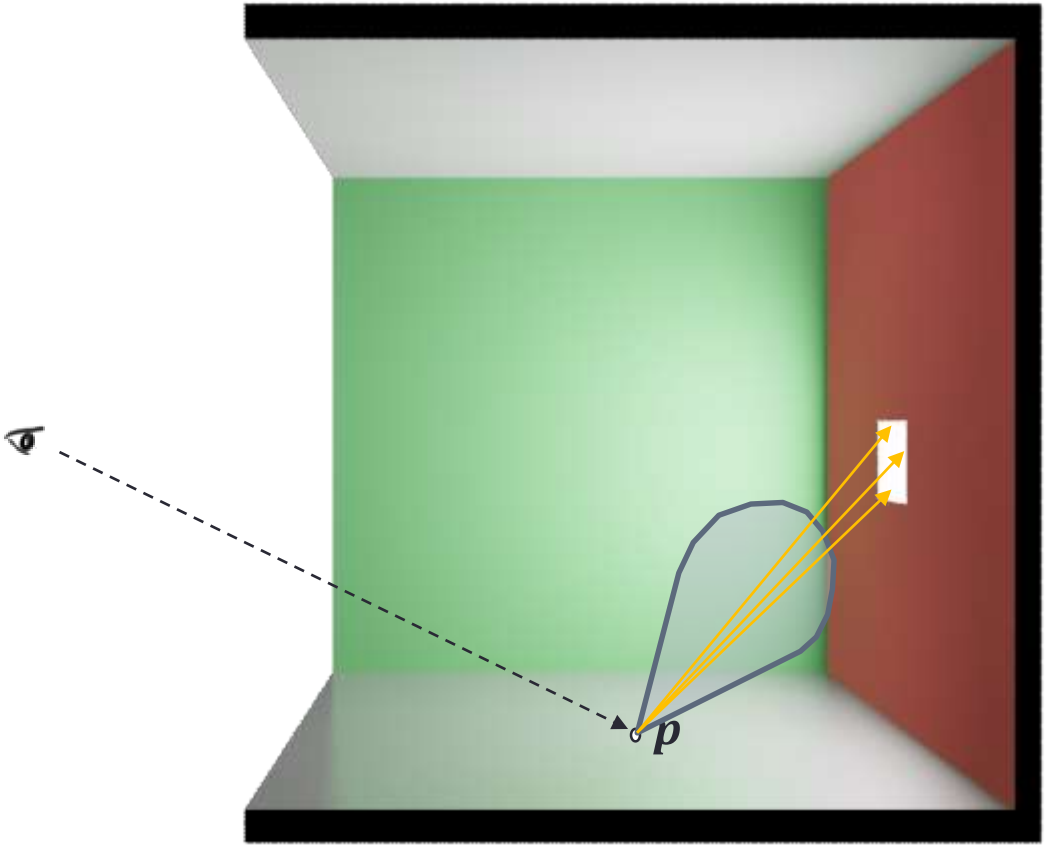
$$\cos\theta_h = \sqrt[n+1]{\xi_1}$$

We know how to sample ω_h with PDF $\sim D(\omega_h)$ need PDF for ω_i:

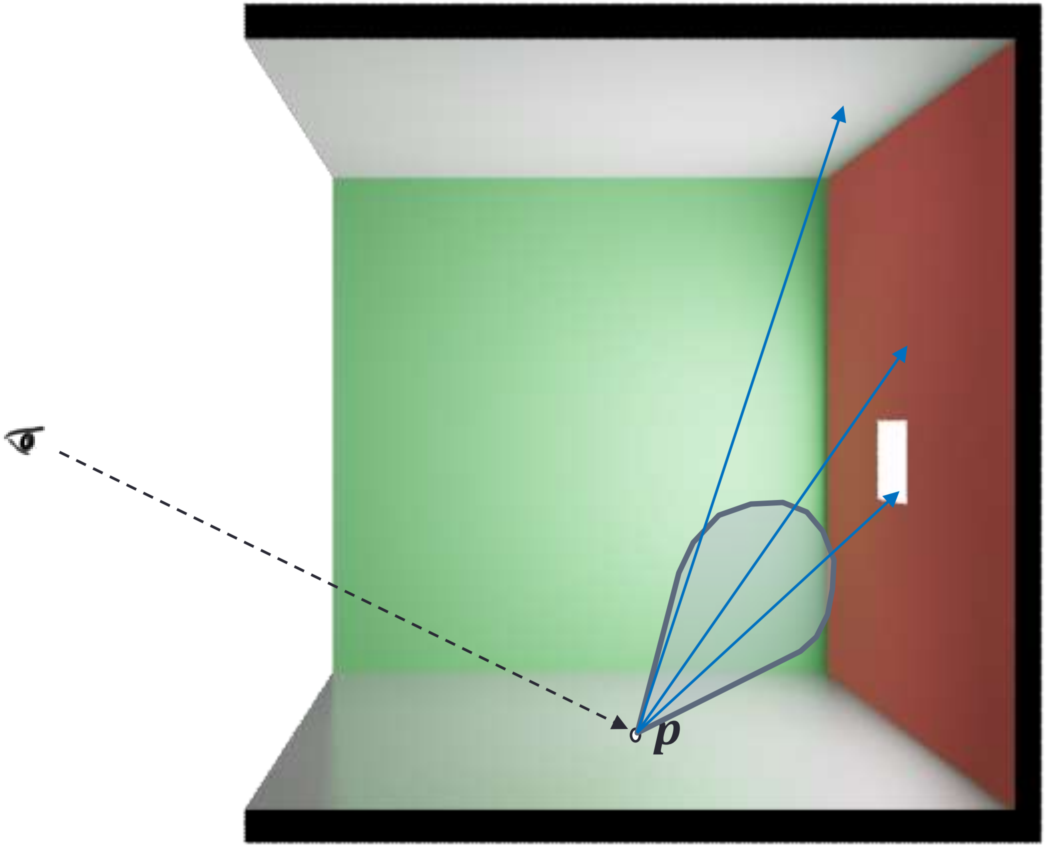
Recall: $\frac{d\omega_h}{d\omega_i} = \frac{1}{4\cos\theta_h}$

$$p(\theta_i) = \frac{p(\theta_h)}{4\cos\theta_h} \qquad p(\omega_i) = \frac{p(\theta_i)}{2\pi}$$

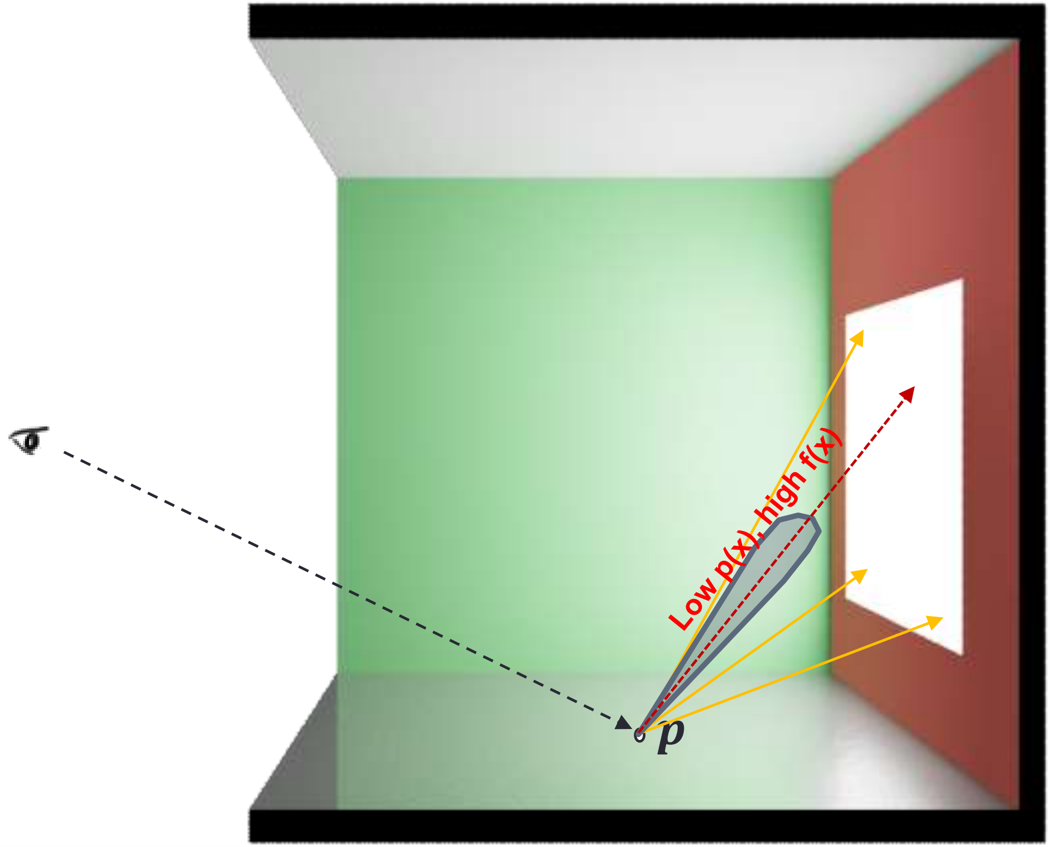
Multiple Importance Sampling



Multiple Importance Sampling

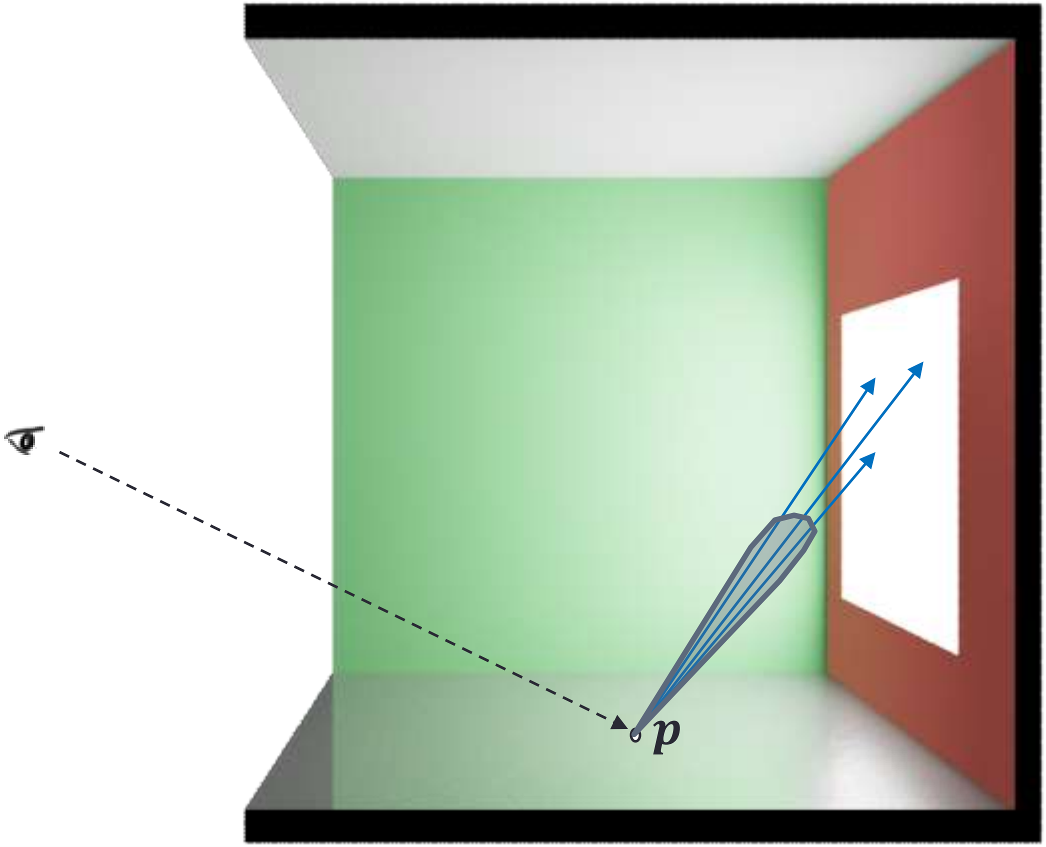


Multiple Importance Sampling



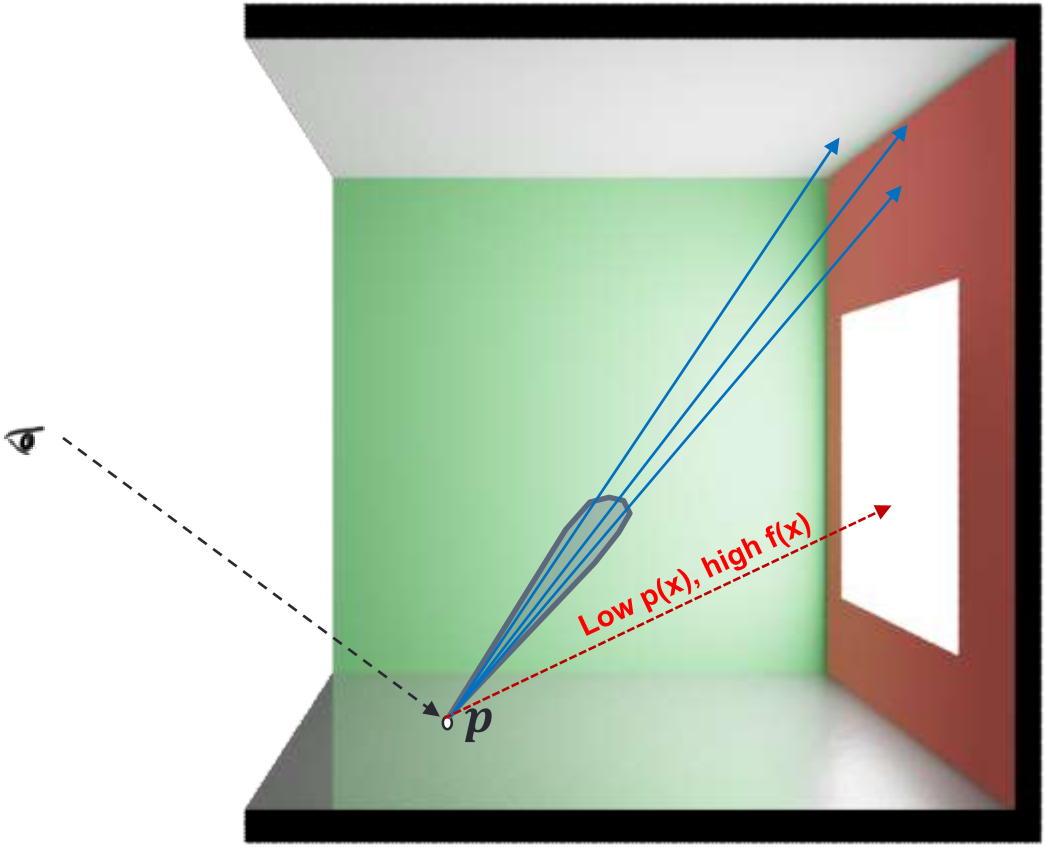
$$L_o(\boldsymbol{p}, \omega) = \int_{\Omega} f(\boldsymbol{p}, \omega, \omega') L_i(\boldsymbol{p}, \omega') \cos(\boldsymbol{n}, \omega')\, d\omega'$$

Multiple Importance Sampling



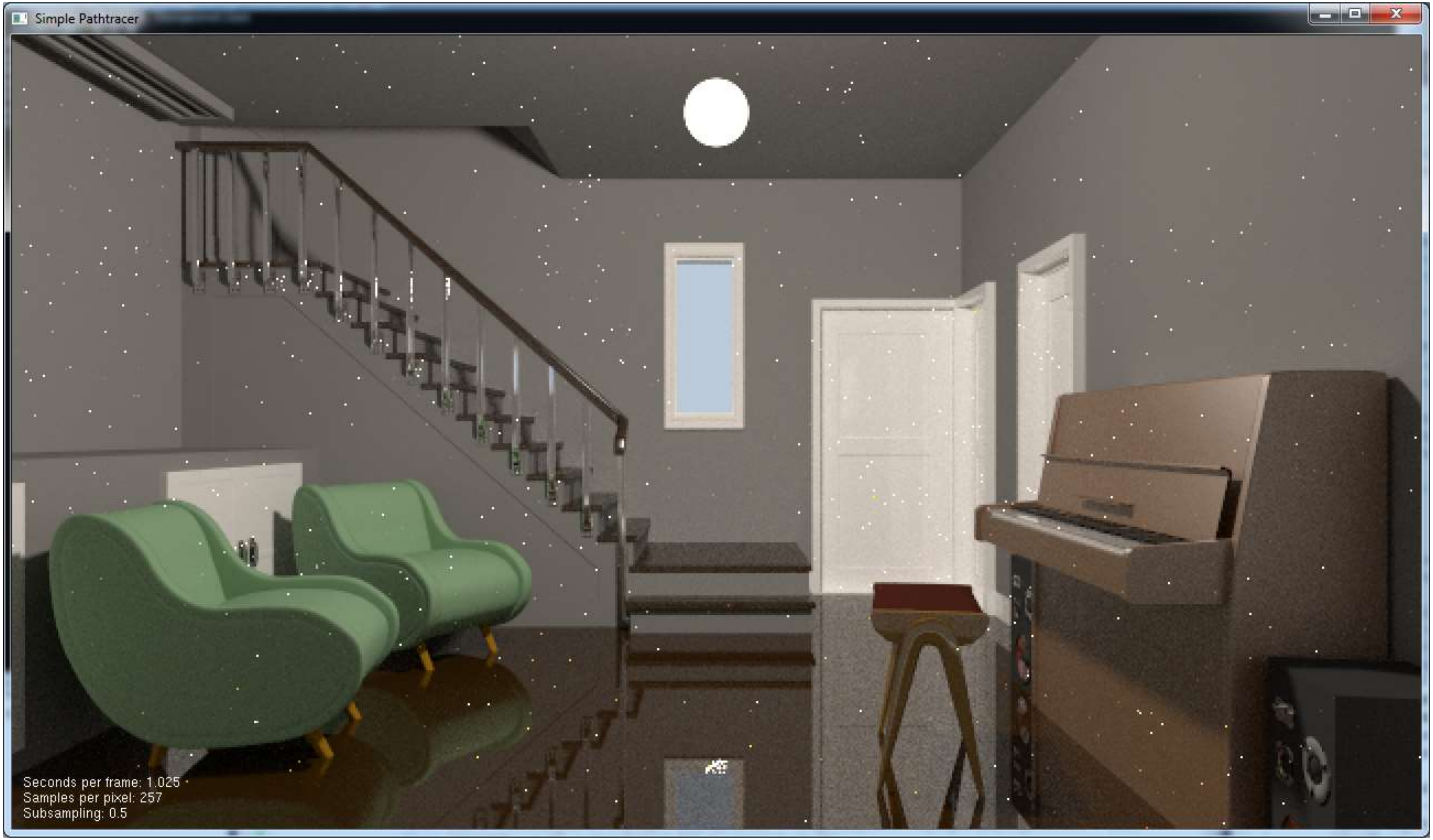
$$L_o(\mathbf{p},\omega) = \int_{\Omega} f(\mathbf{p},\omega,\omega') L_i(\mathbf{p},\omega') \cos(\mathbf{n},\omega')\, d\omega'$$

Multiple Importance Sampling

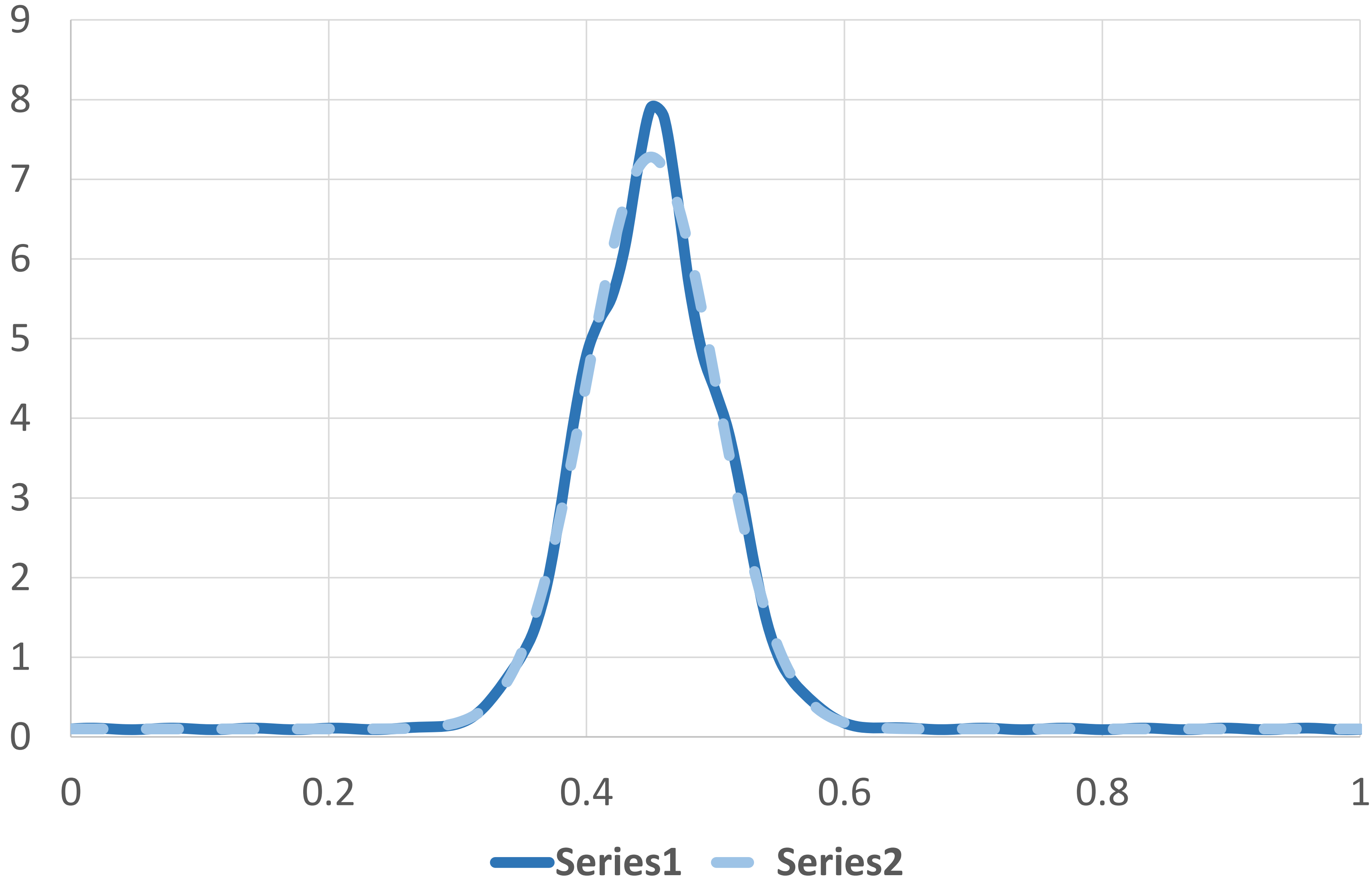


$$L_o(\mathbf{p},\omega) = \int_{\Omega} f(\mathbf{p},\omega,\omega') L_i(\mathbf{p},\omega') \cos(\mathbf{n},\omega')\, d\omega'$$

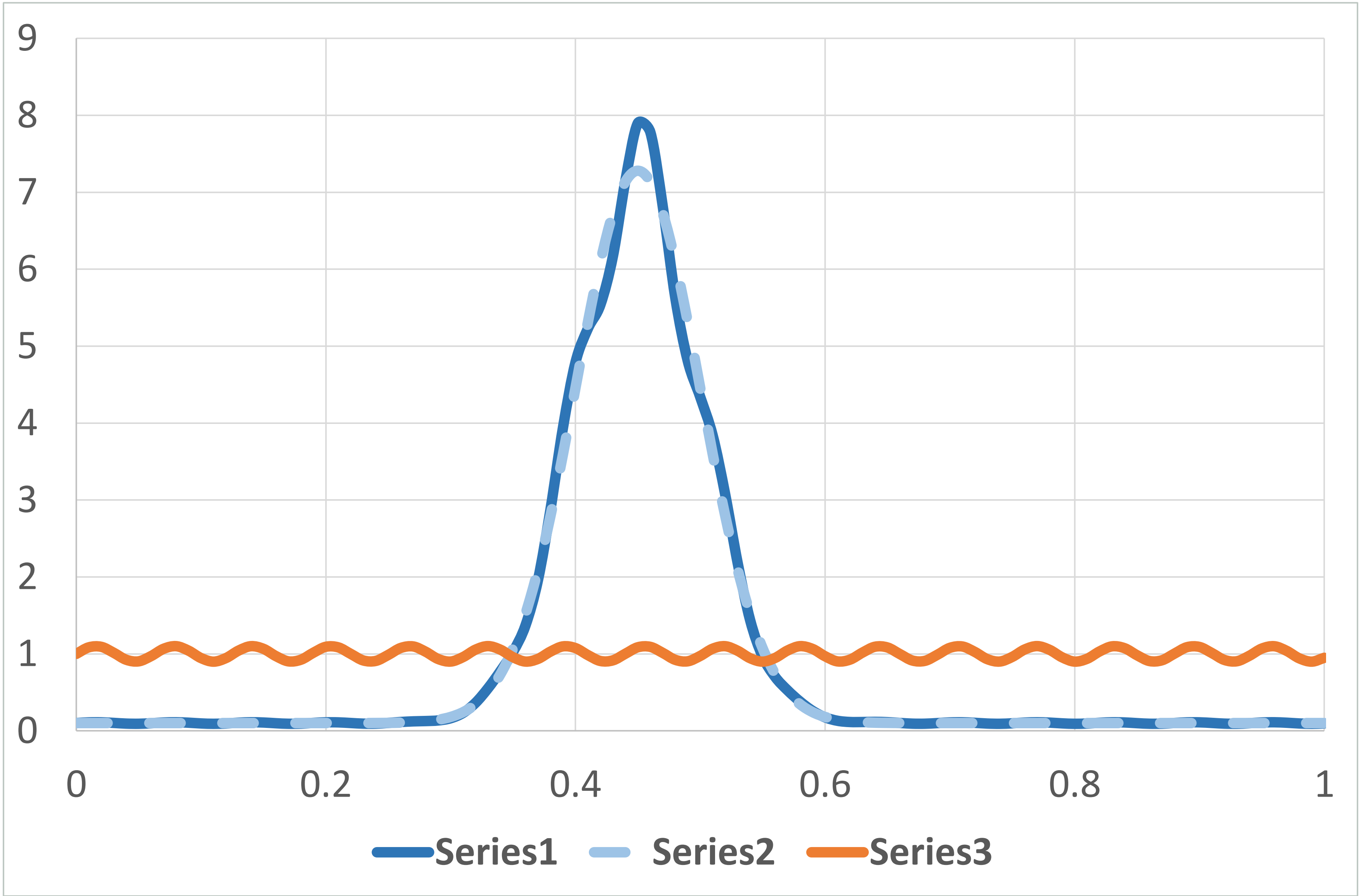
Multiple Importance Sampling



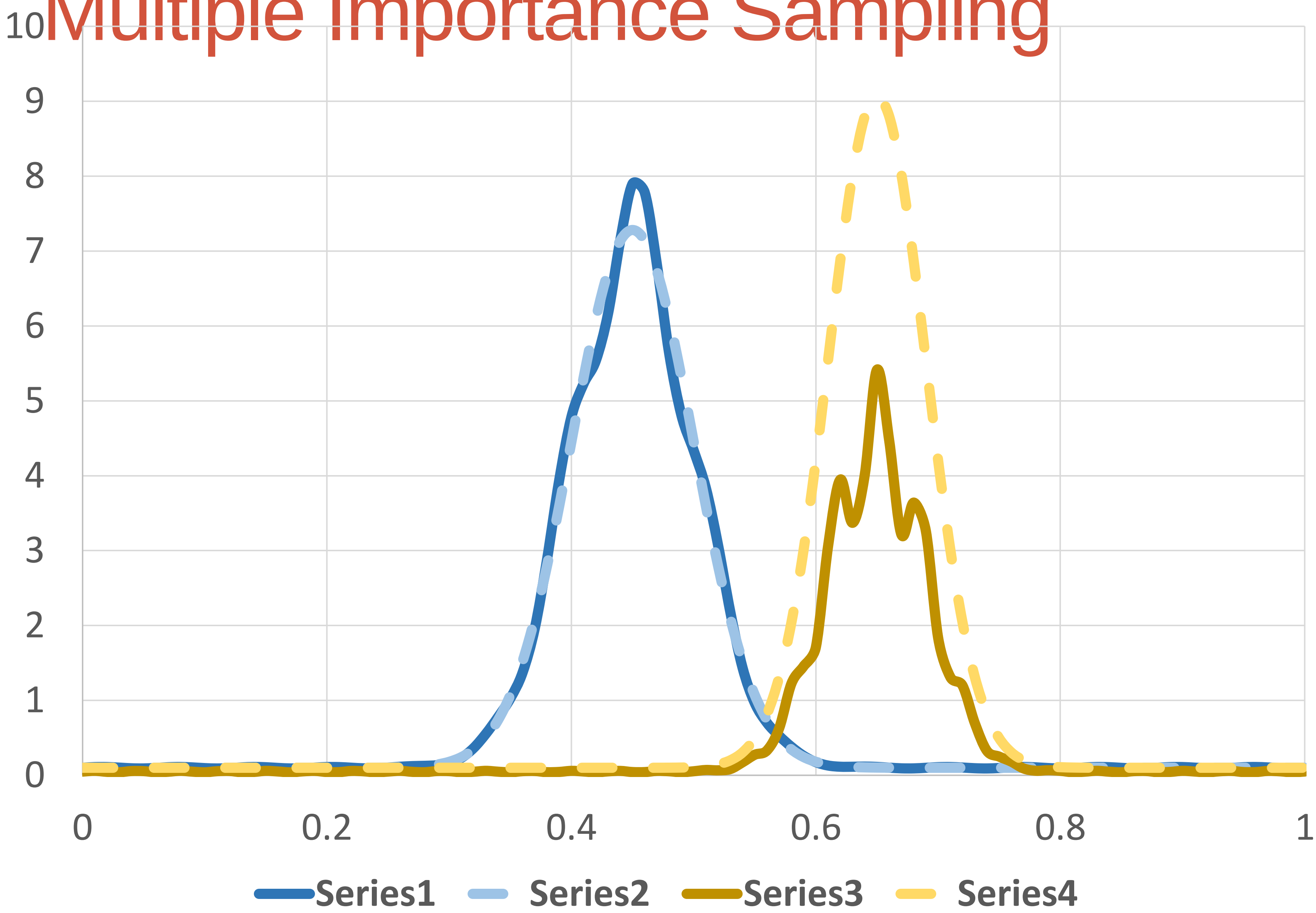
Multiple Importance Sampling



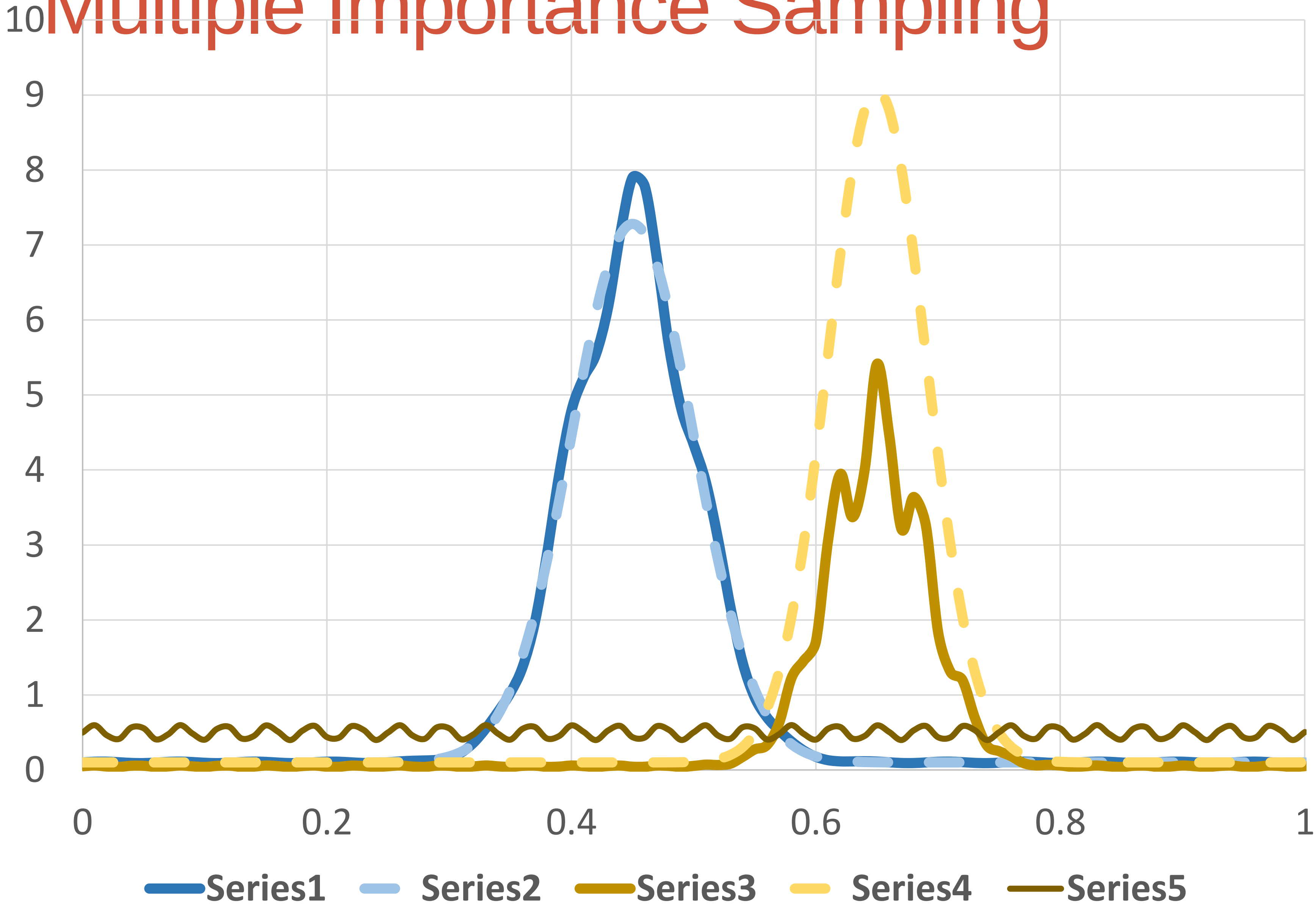
Multiple Importance Sampling



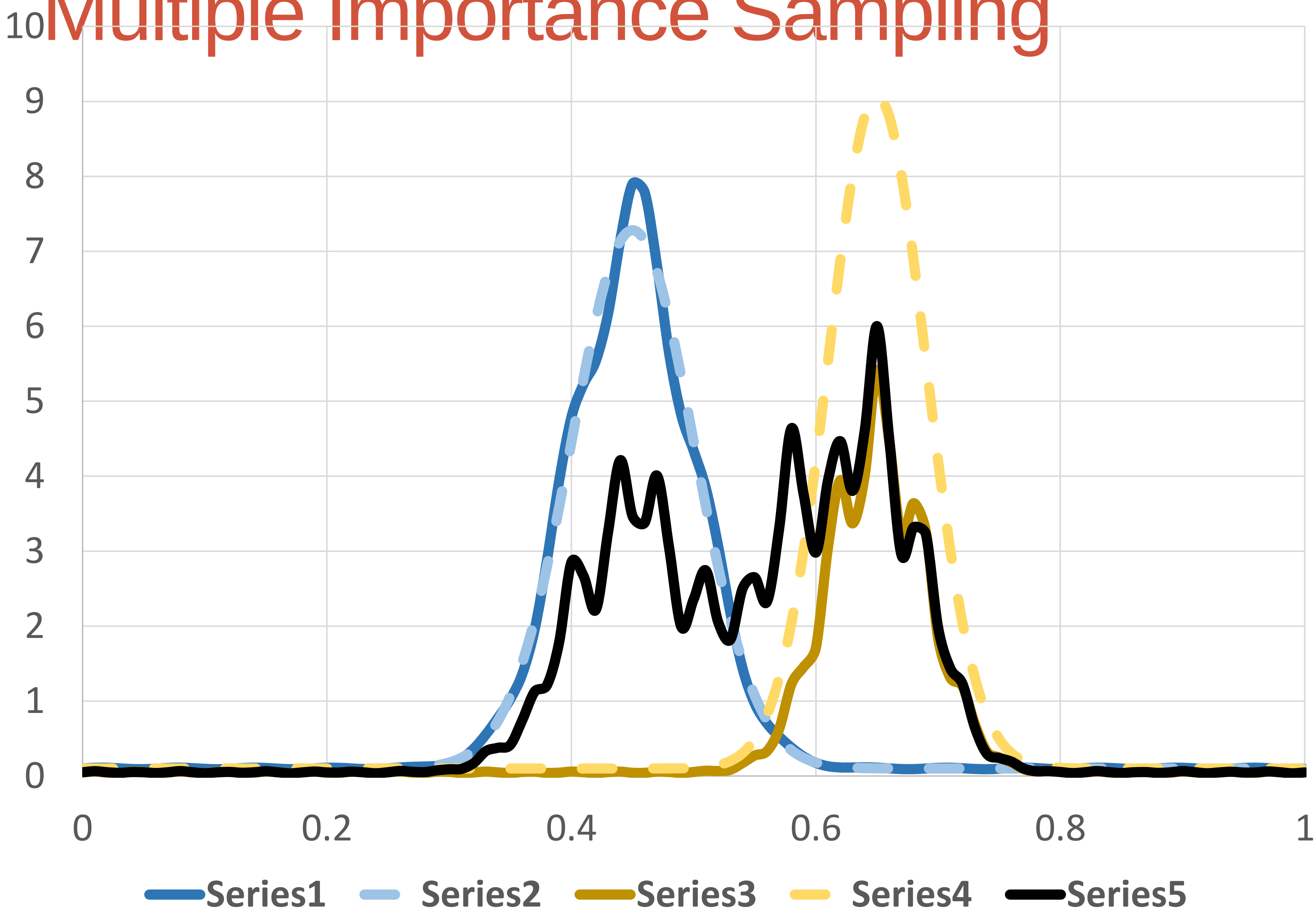
Multiple Importance Sampling



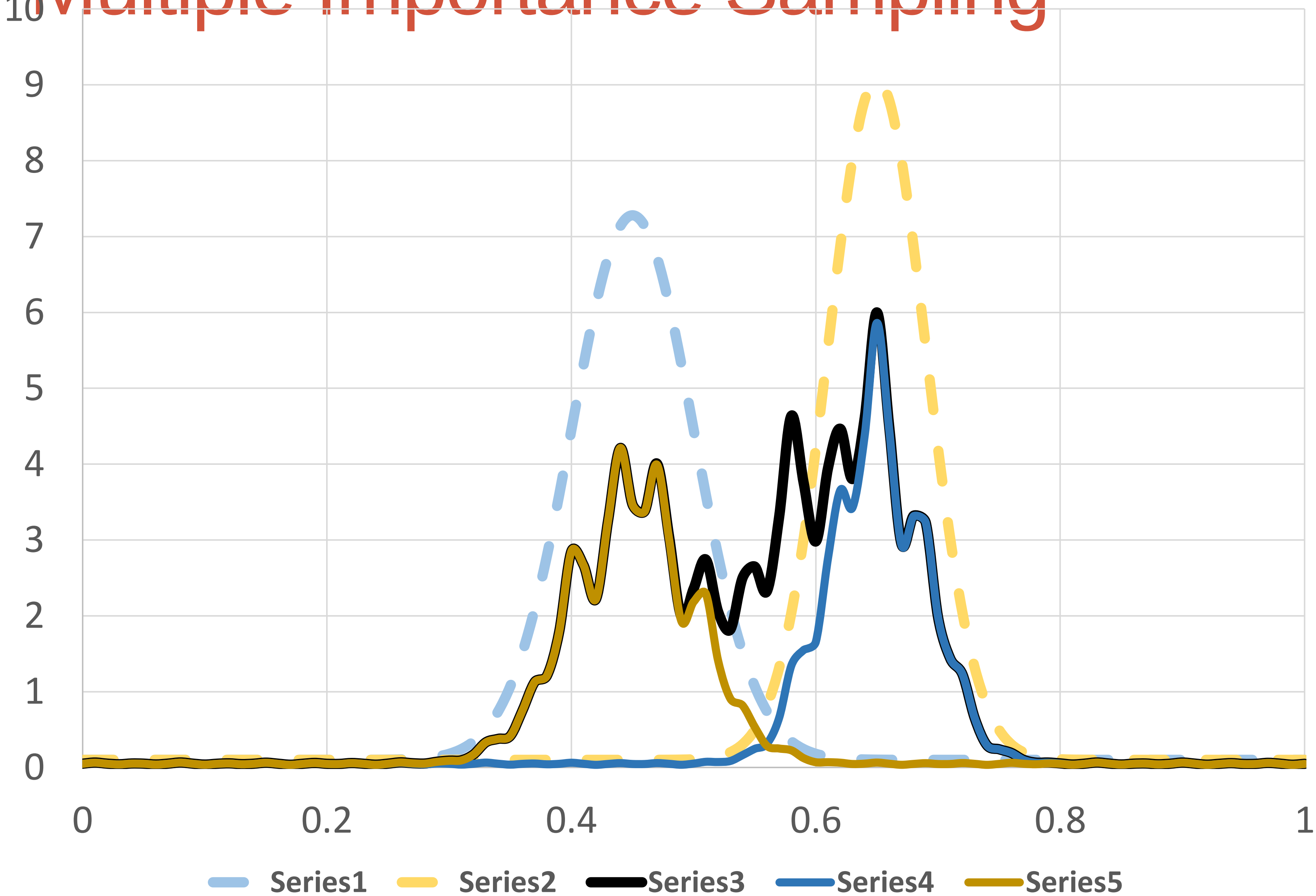
Multiple Importance Sampling



Multiple Importance Sampling



Multiple Importance Sampling



Multiple Importance Sampling

Need to estimate:

$$\int f(x)g(x)dx$$

Could Use:

$$0.5\left(\frac{f(X)g(X)}{p_f(X)} + \frac{f(Y)g(Y)}{p_g(Y)}\right)?$$

Multiple Importance Sampling

Need to estimate:

$$\int f(x)g(x)dx$$

Could Use:

$$0.5\left(\frac{f(X)g(X)}{p_f(X)} + \frac{f(Y)g(Y)}{p_g(Y)}\right)?$$

Better (MIS):

$$0.5\left(\frac{f(X)g(X)}{0.5(p_f(X) + p_g(X))} + \frac{f(Y)g(Y)}{0.5(p_f(Y) + p_g(Y))}\right)$$

Multiple Importance Sampling

Need to estimate:

$$\int f(x)g(x)dx$$

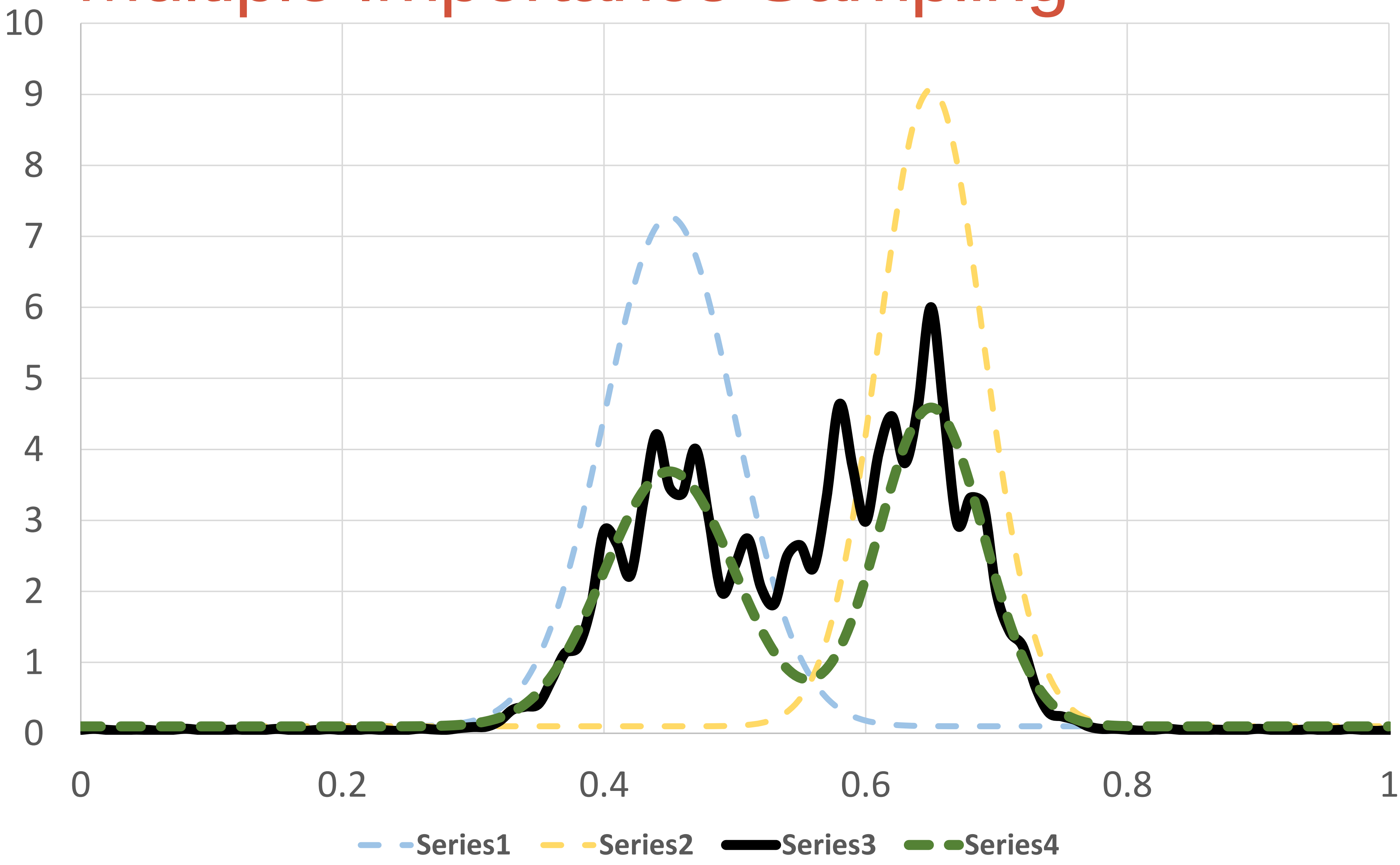
Could Use:

$$0.5\left(\frac{f(X)g(X)}{p_f(X)}+\frac{f(Y)g(Y)}{p_g(Y)}\right)?$$

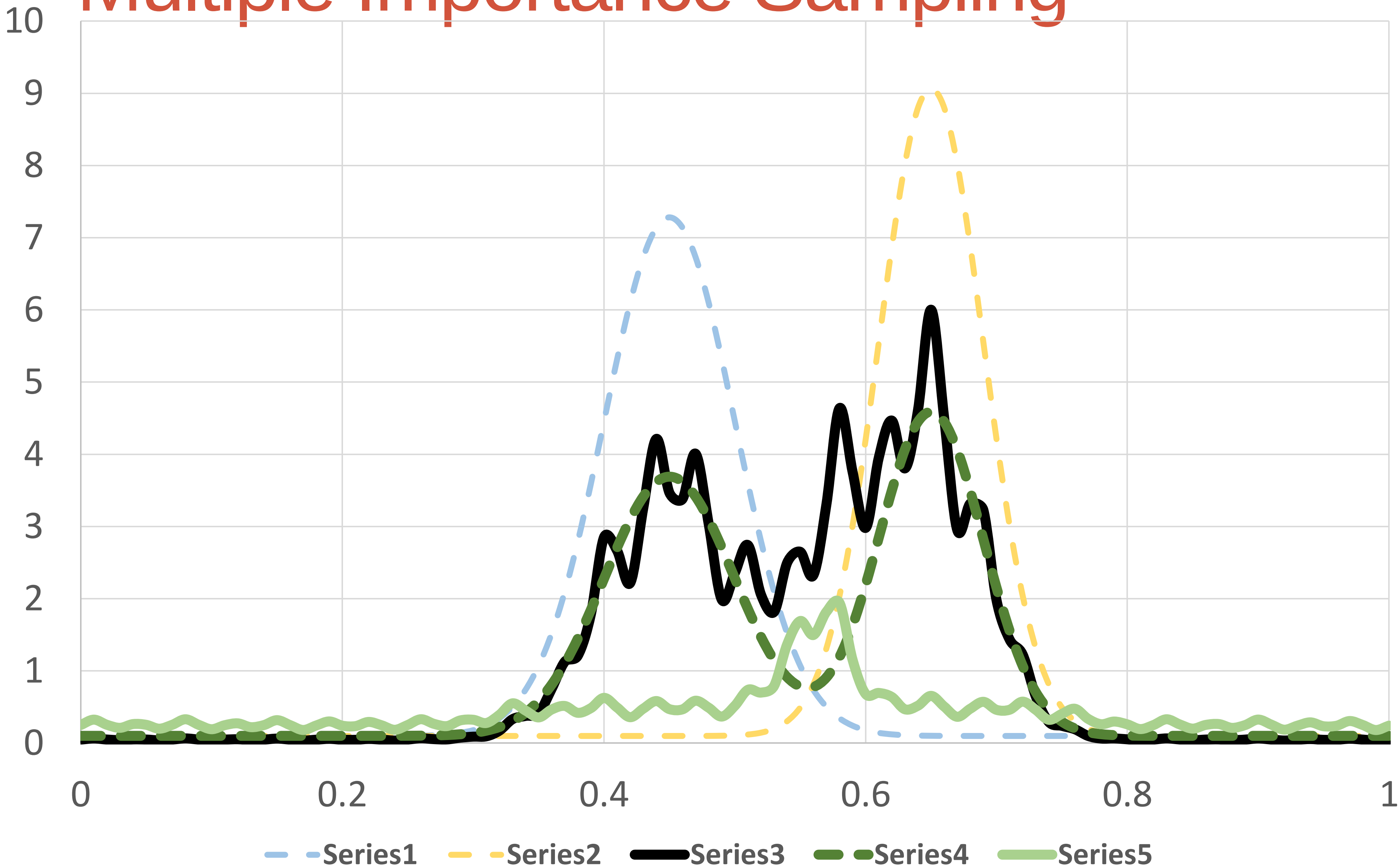
Better (MIS):

$$\frac{f(X)g(X)}{p_f(X)+p_g(X)}+\frac{f(Y)g(Y)}{p_f(Y)+p_g(Y)}$$

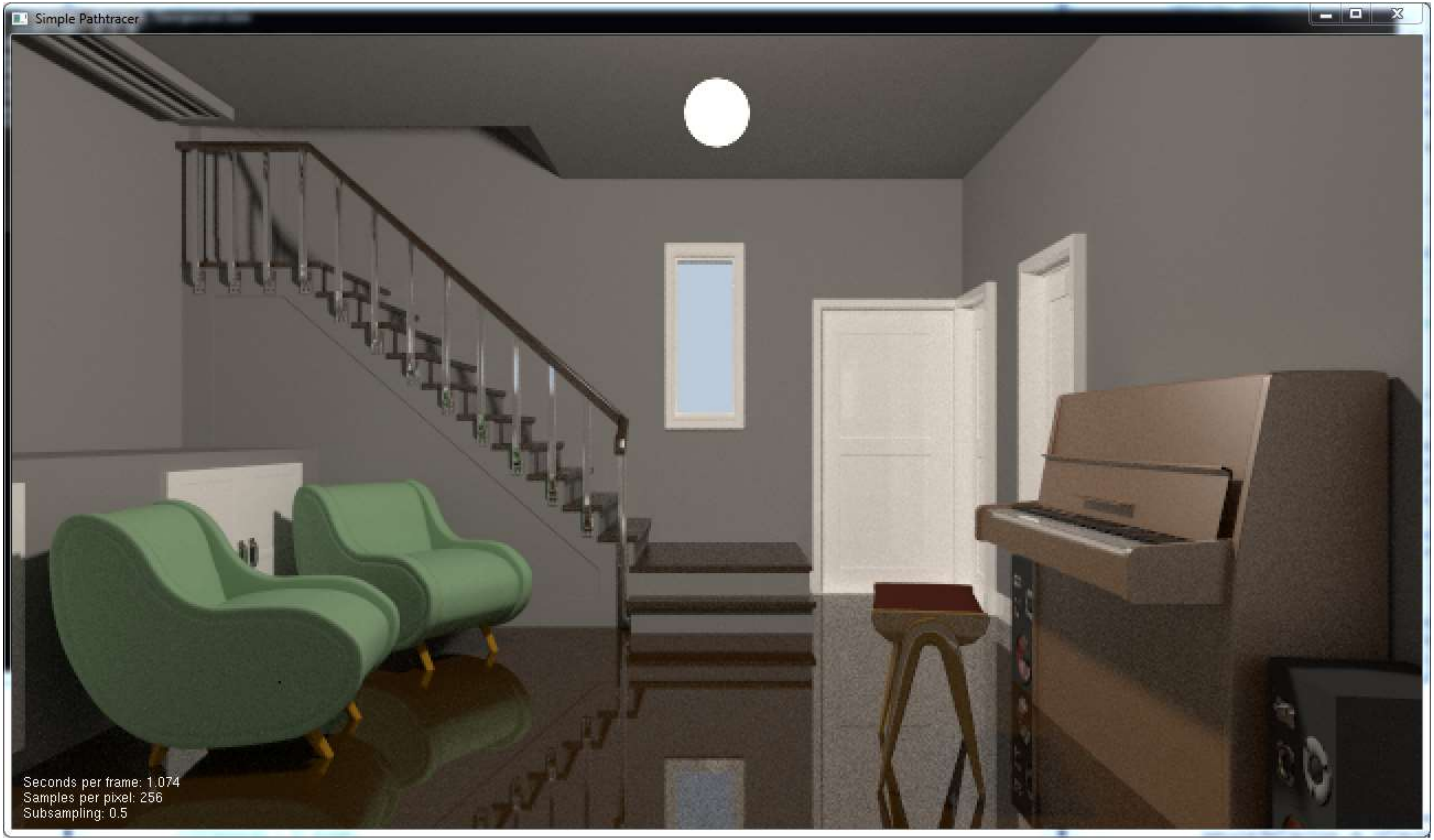
Multiple Importance Sampling



Multiple Importance Sampling

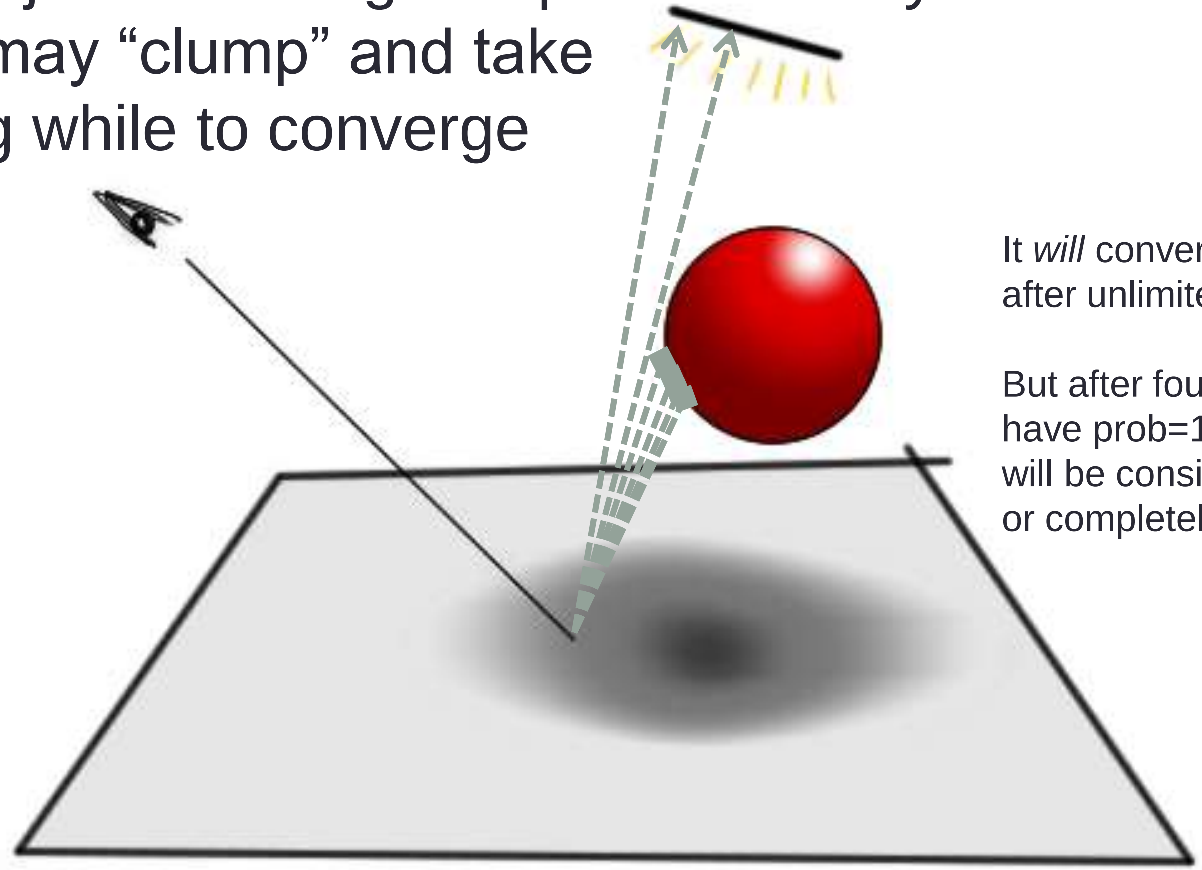


Multiple Importance Sampling



Stratified Sampling

- Another standard variance reduction method
- When just choosing samples randomly over the domain, they may "clump" and take a long while to converge

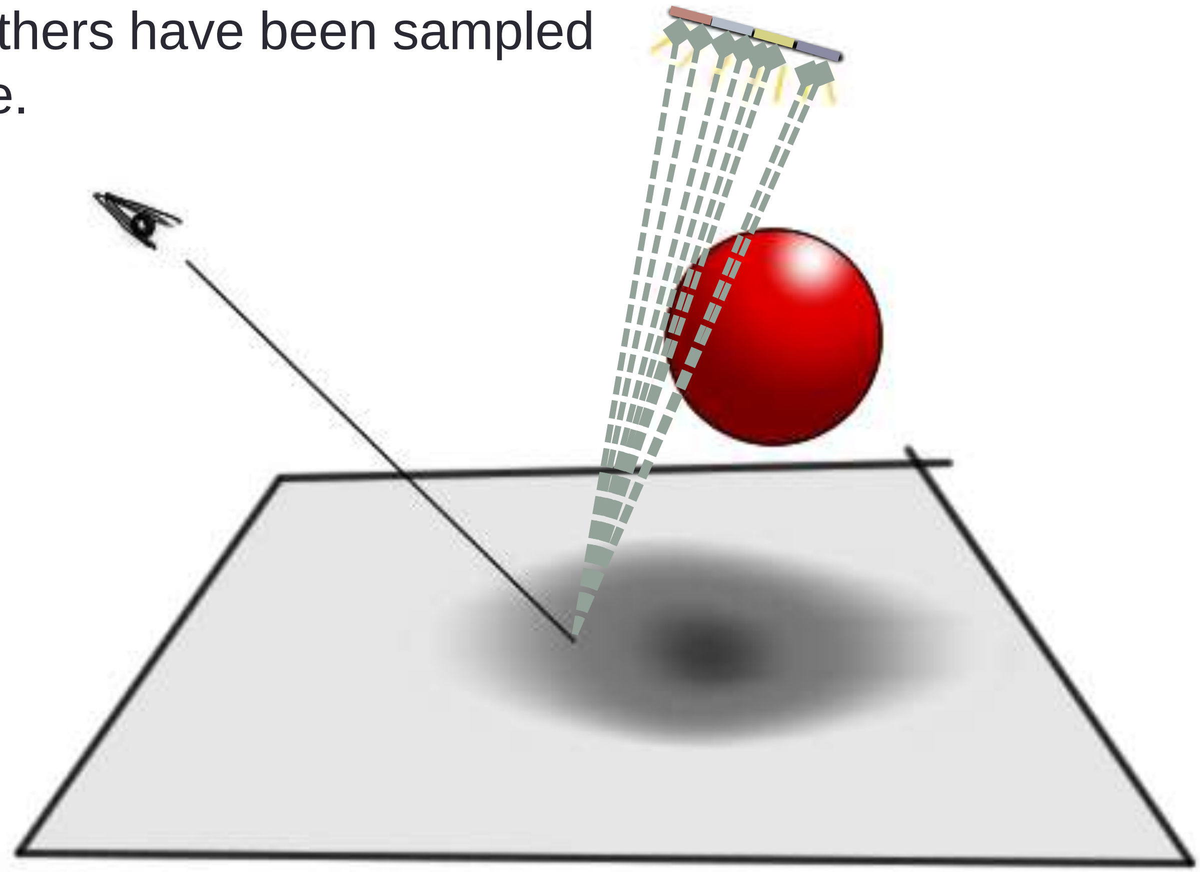


It *will* converge to 0.5 after unlimited time.

But after four samples I still have prob=1/8 that the pixel will be considered all in shadow or completely unshadowed

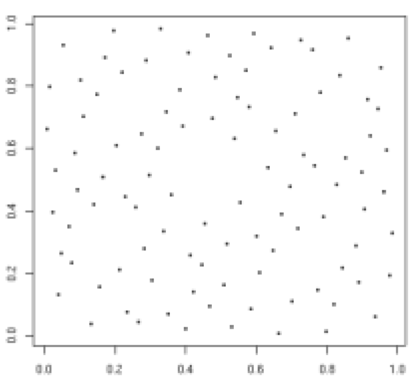
Stratified Sampling

- Divide domain into “strata”
  - Don’t sample one strata again until all others have been sampled once.

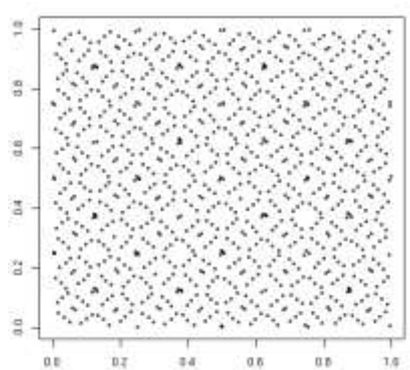


Stratified Sampling

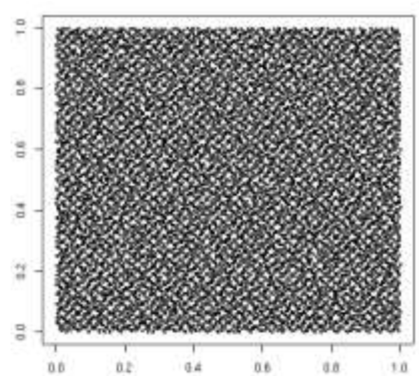
- If we know how many samples we want to take, we can get good stratification from "jittering"
- If not, we want any sequence of samples to have good stratification. We can use a *Low Discrepancy Sequence*



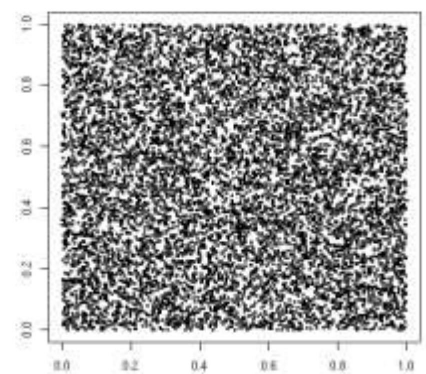
SOBOL: 100 samples



SOBOL: 1000 samples



SOBOL: 10000 samples



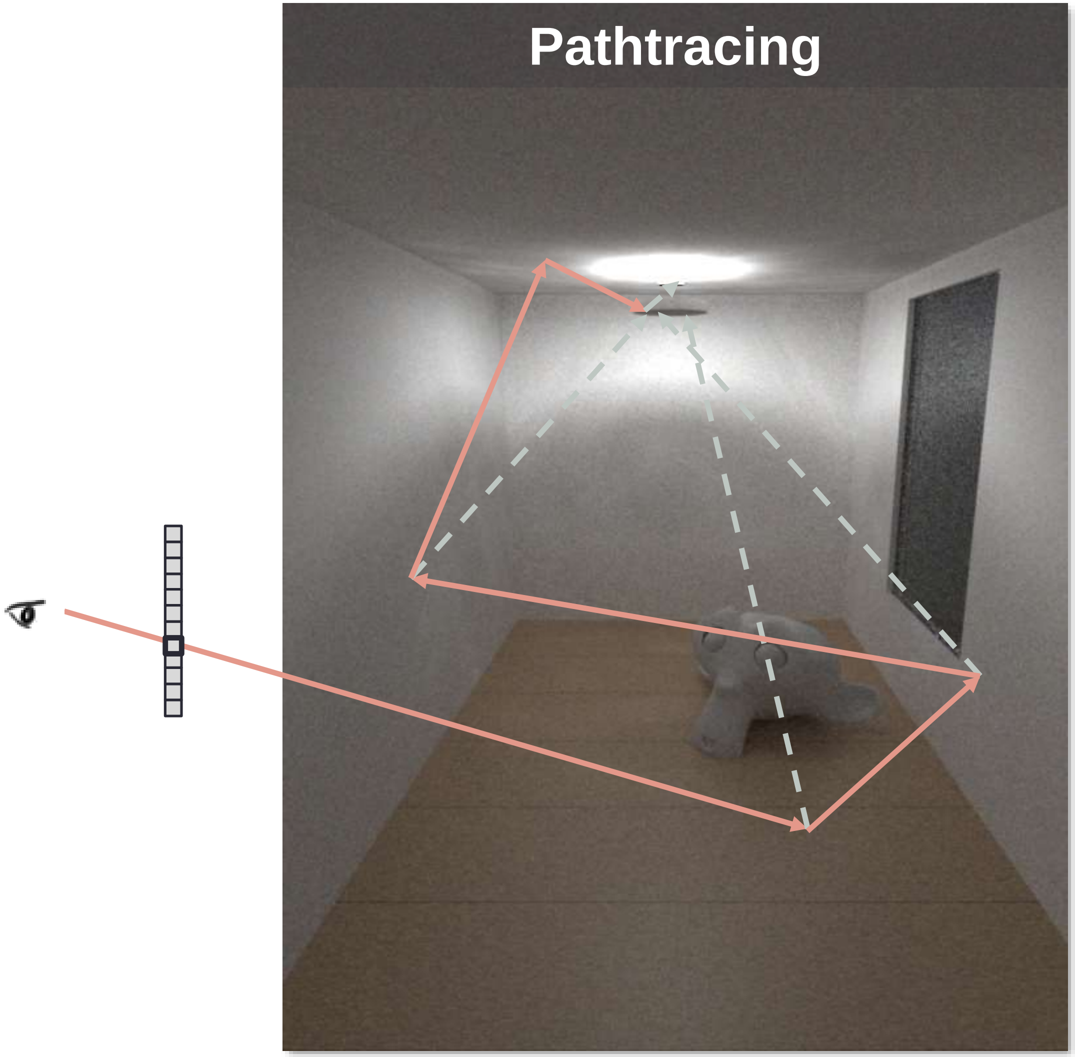
RANDOM: 10000 samples

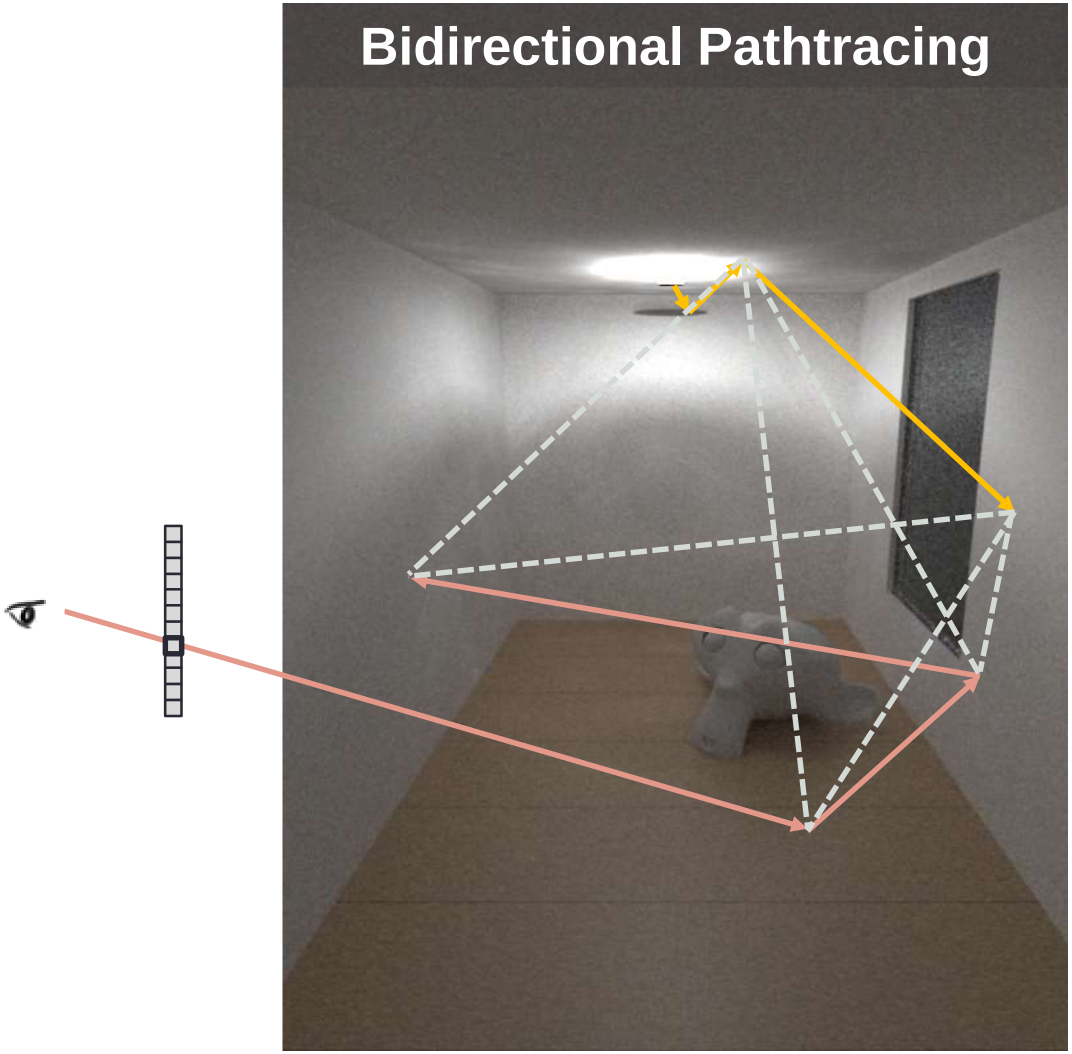
Are we done yet?

Pathtracing 3m

Pathtracing 3m

Bidirectional Pathtracing 3m


Pathtracing


Bidirectional Pathtracing

Bidirectional Path Tracing


Pathtracing 75SPP (~5min)


Bidirectional Pathtracing 45SPP (~5min)

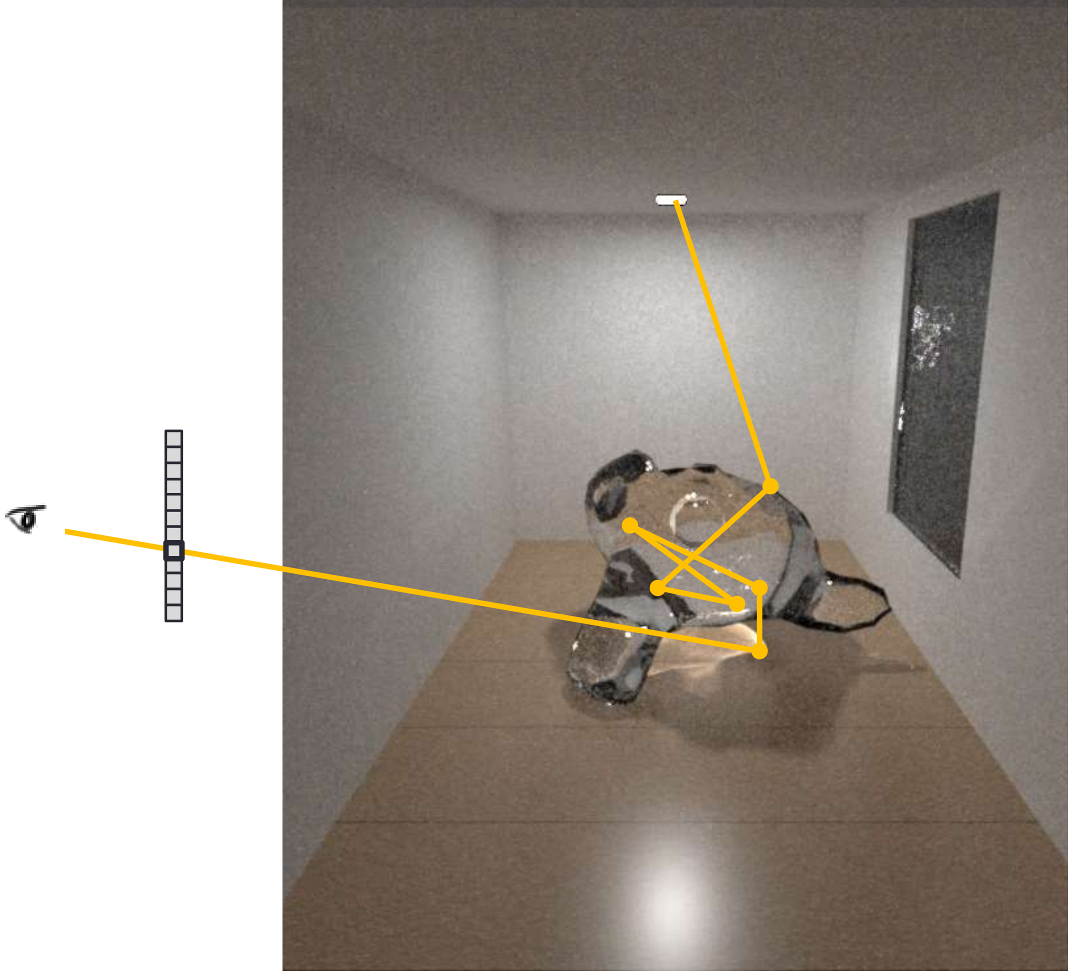
Are we done yet?

Bidirectional Pathtracing 30m

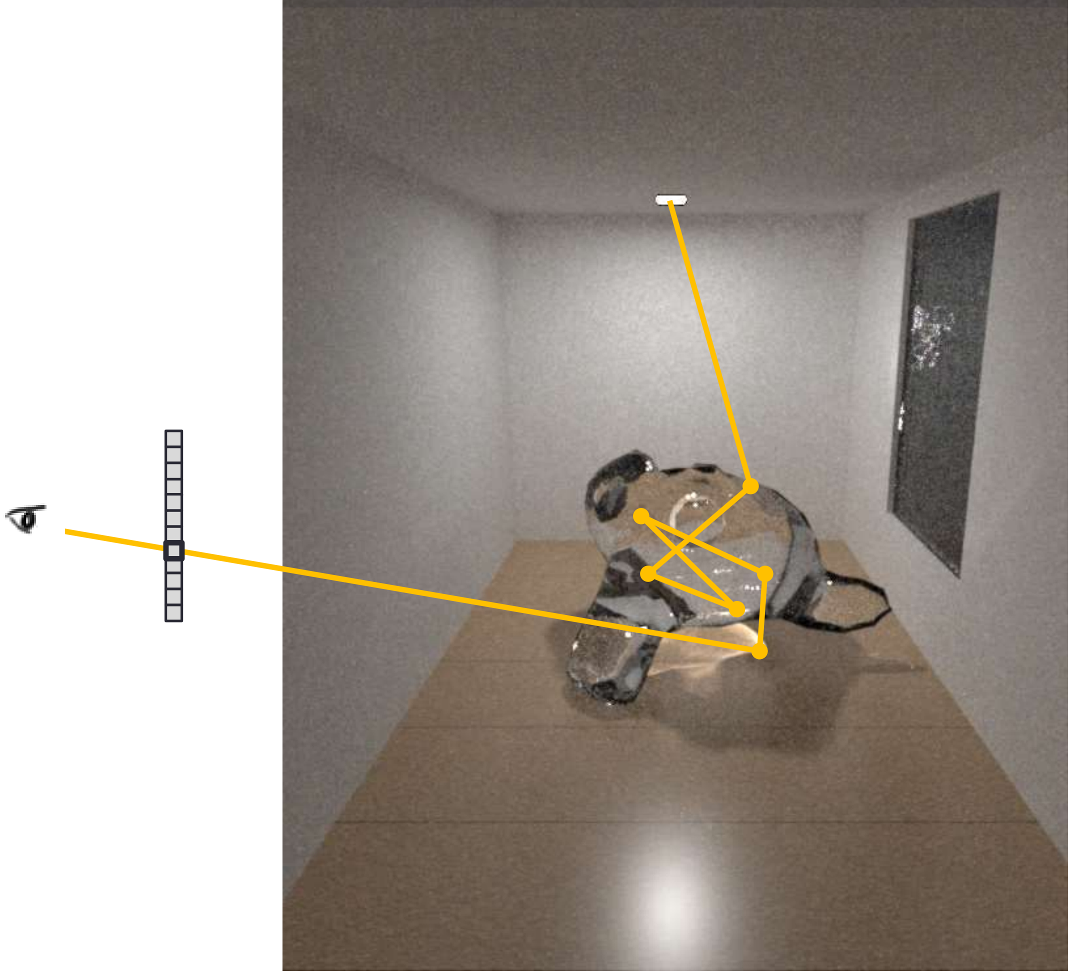
Bidirectional Pathtracing 30m

Metropolis Light Transport 30m

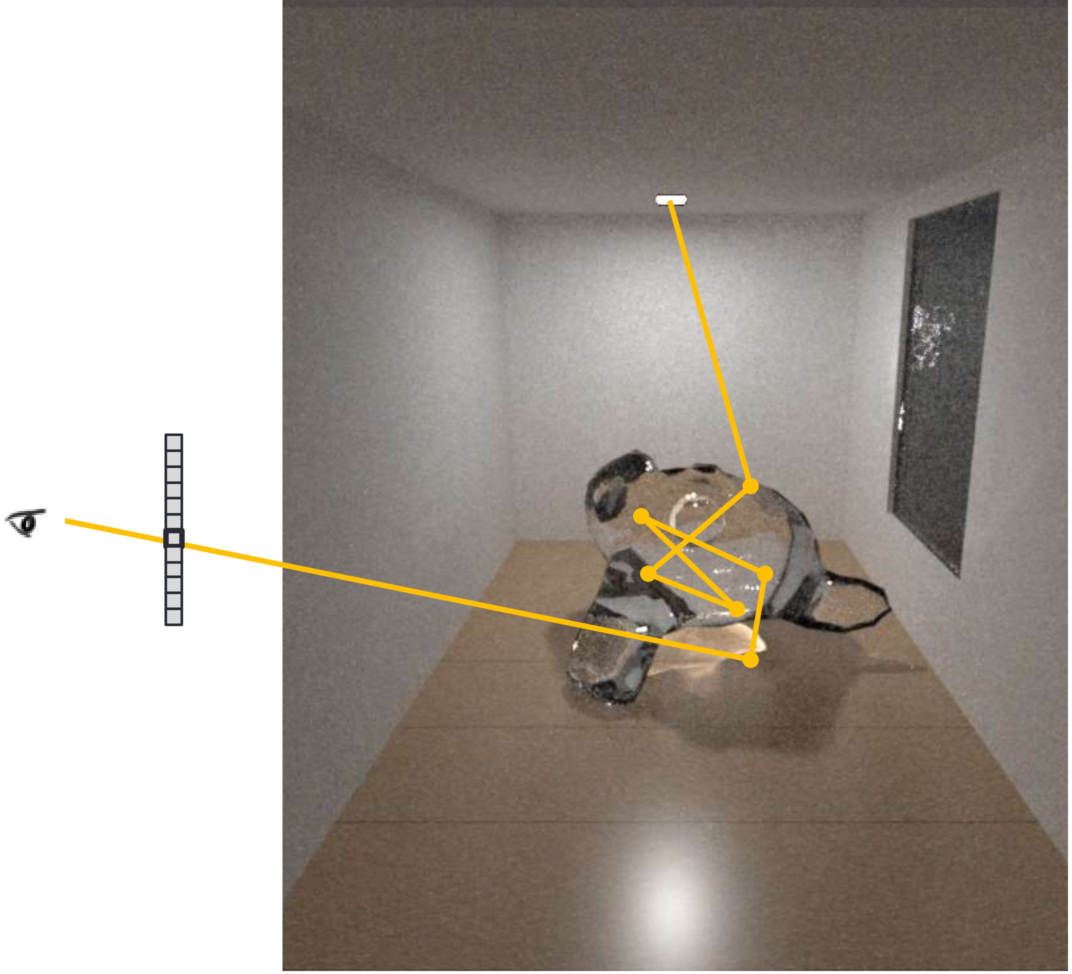
Metropolis Light Transport



Metropolis Light Transport



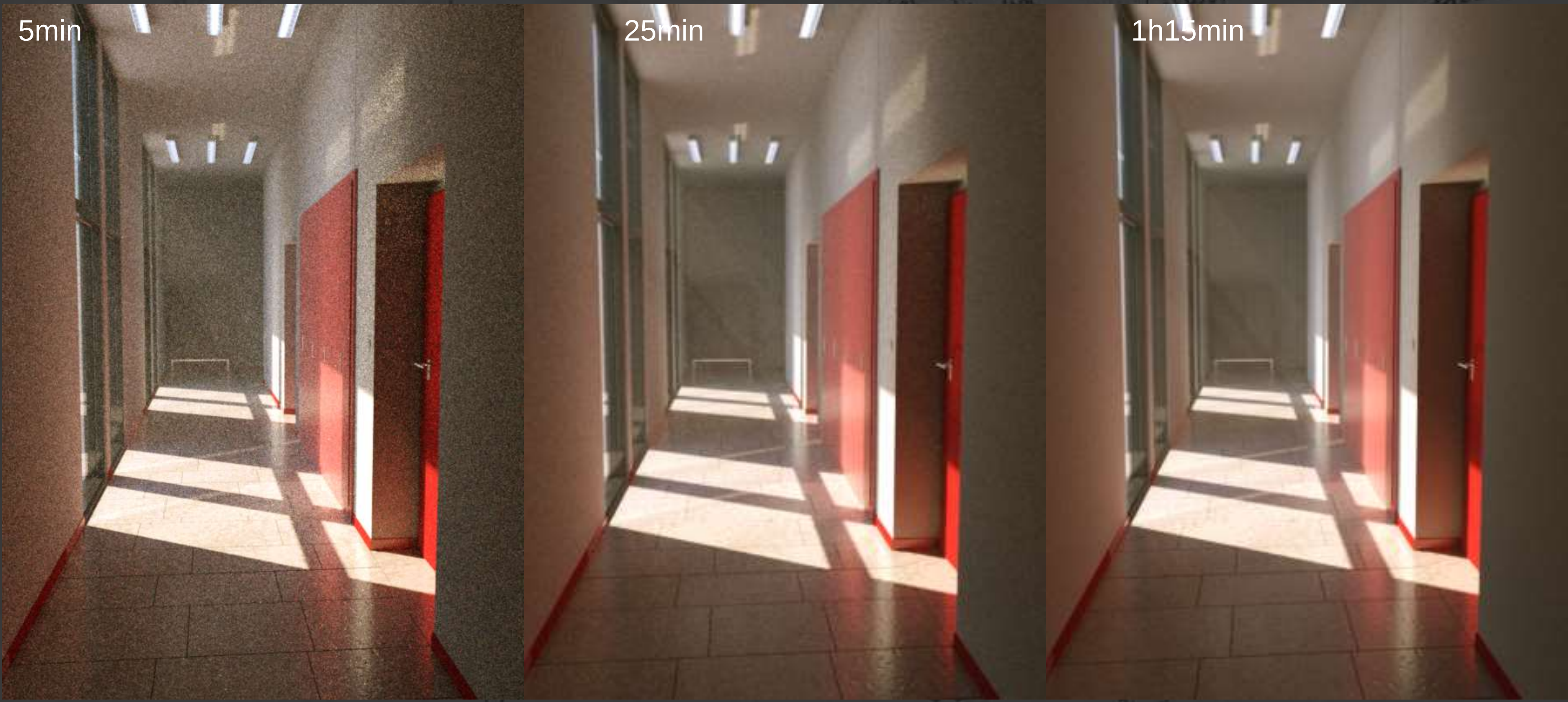
Metropolis Light Transport



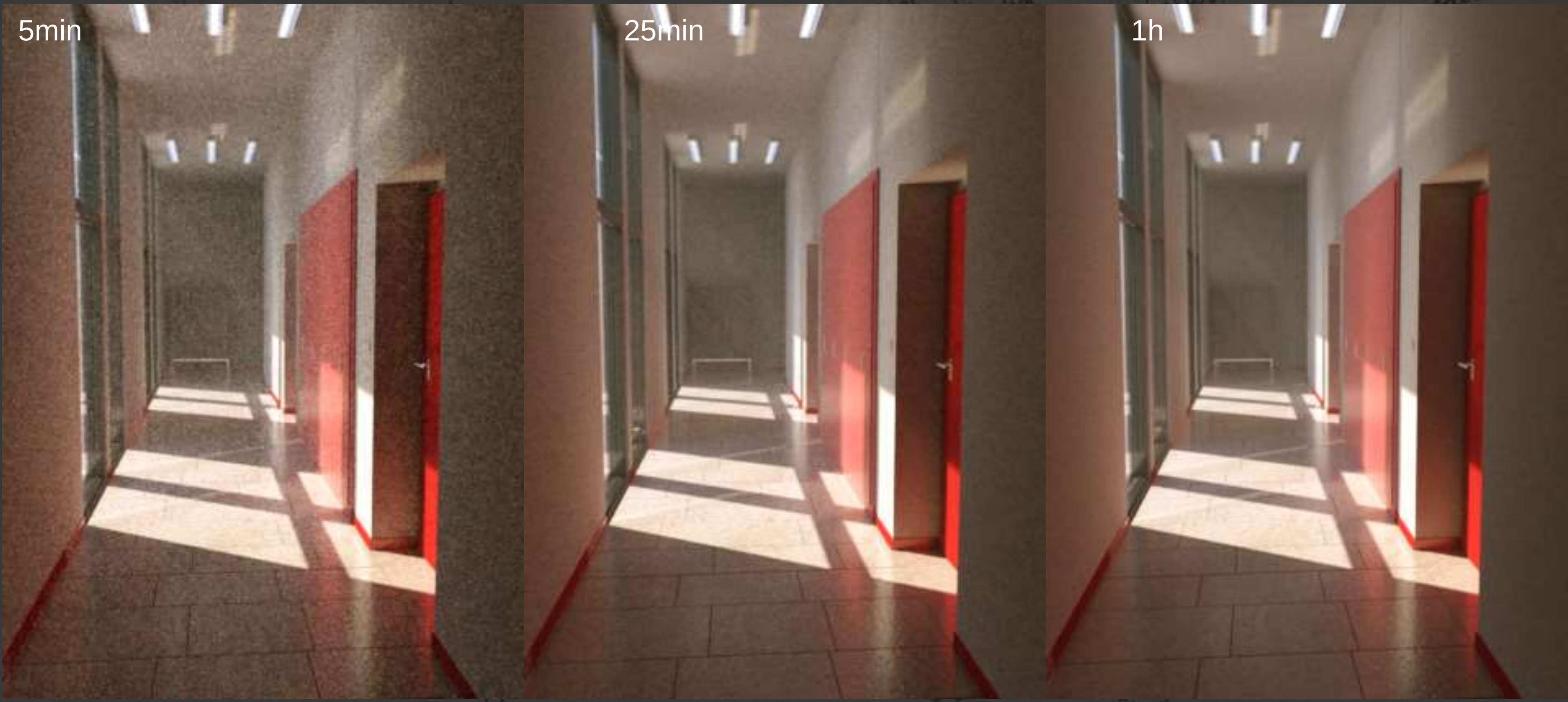
Bidirectional Path Tracing

Metropolis Light Transport

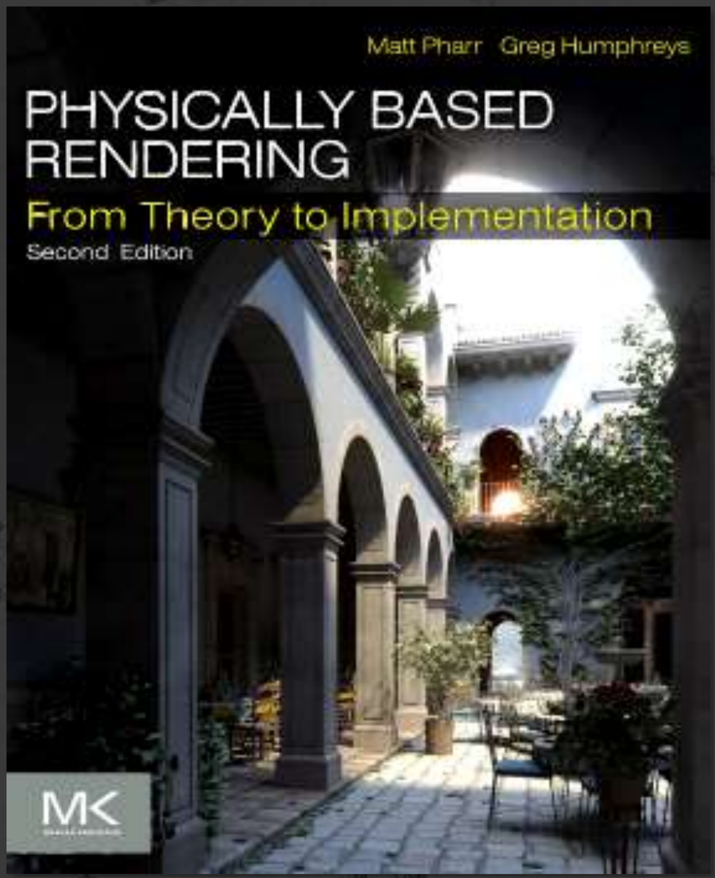
Bidirectional Path Tracing



Metropolis Light Transport



Further Reading



Pathtracing Lab