PA213

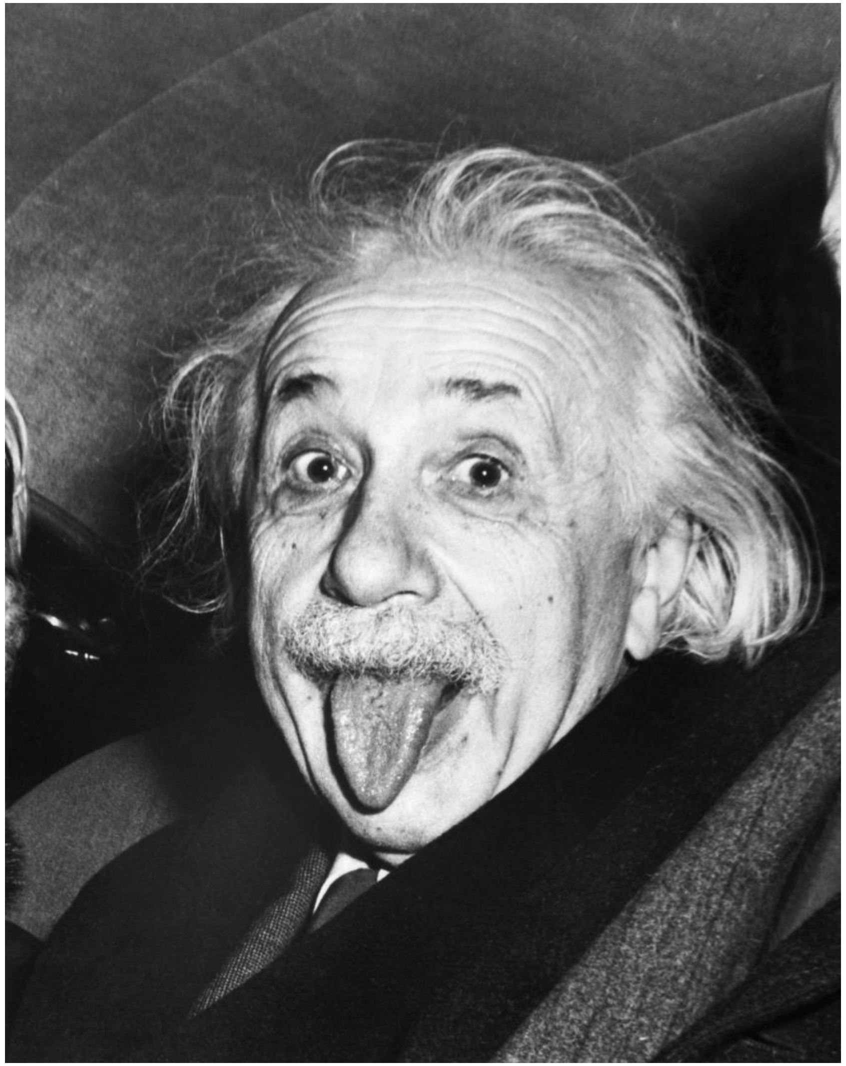
Jan Byška (inspired by Keenan Crane lectures)

Monte Carlo Rendering

HCI LAB visitlab

Mass–Energy Equivalence

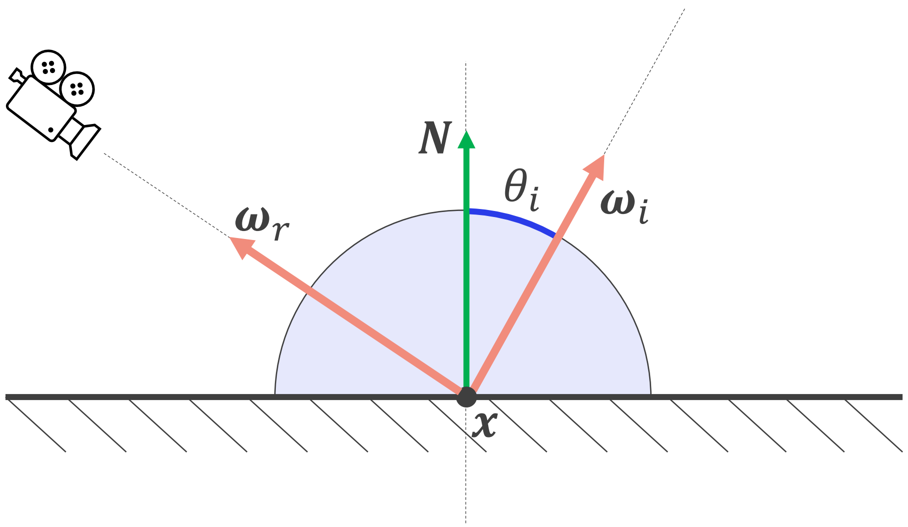
$$E = mc^2$$



Albert Einstein

Rendering Equation

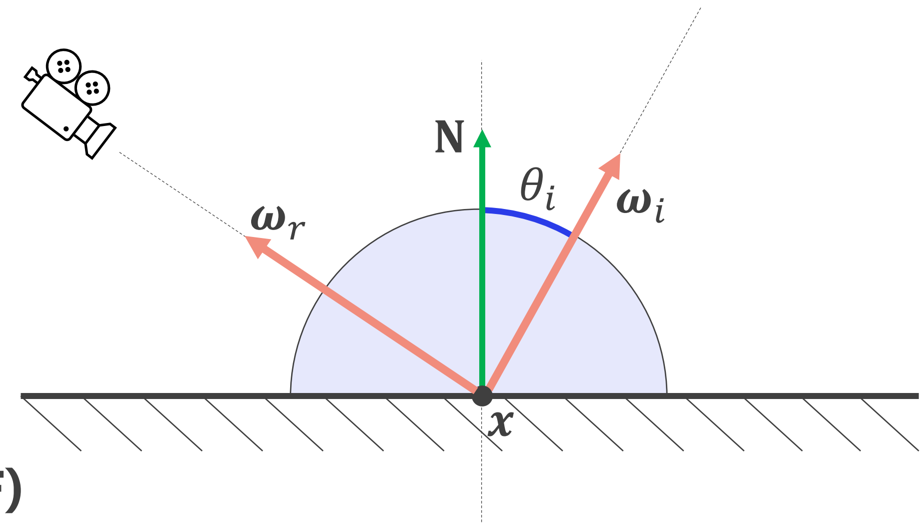
$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \int_\Omega f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \, d\boldsymbol{\omega}_i$$



James Kajiya

Rendering Equation

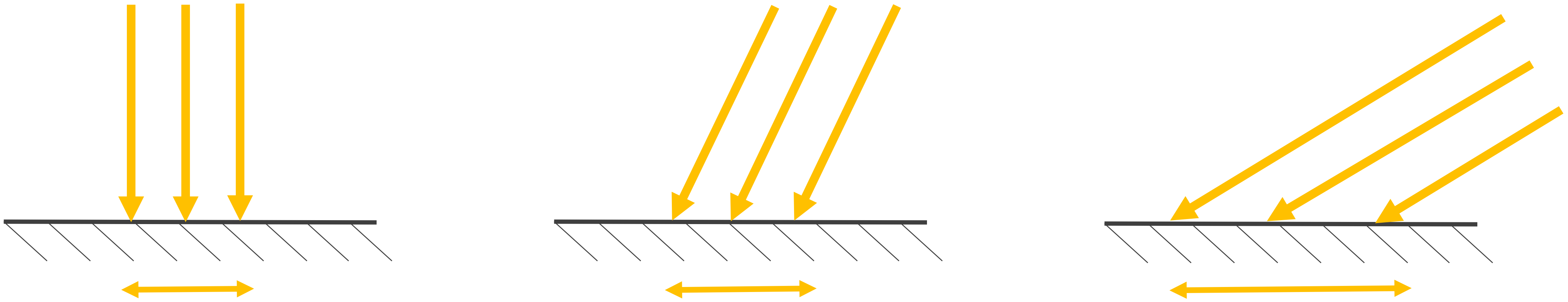
$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \int_\Omega f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \, d\boldsymbol{\omega}_i$$



- L_r is **radiance**
  - amount of light emitted from $\boldsymbol{x}$ in direction $\boldsymbol{\omega}_r$
- L_e is **emitted radiance**
  - non-zero, if $\boldsymbol{x}$ is a light source
- L_i is **incoming radiance**
  - amount of light arriving to $\boldsymbol{x}$ from direction $\boldsymbol{\omega}_i$
- f_r is **bidirectional reflectance distribution function (BRDF)**
  - material property at $\boldsymbol{x}$
  - it is a ratio of the light reflected along $\boldsymbol{\omega}_r$ to the light incoming from $\boldsymbol{\omega}_i$
- $\cos\theta_i = \mathbf{N} \cdot \boldsymbol{\omega}_i$
  - describes Lambert's cosine law

Lambert's Cosine Law

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \int_\Omega f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \underline{\cos\theta_i}\, d\boldsymbol{\omega}_i$$



Bidirectional Reflectance Distribution Function (BRDF)

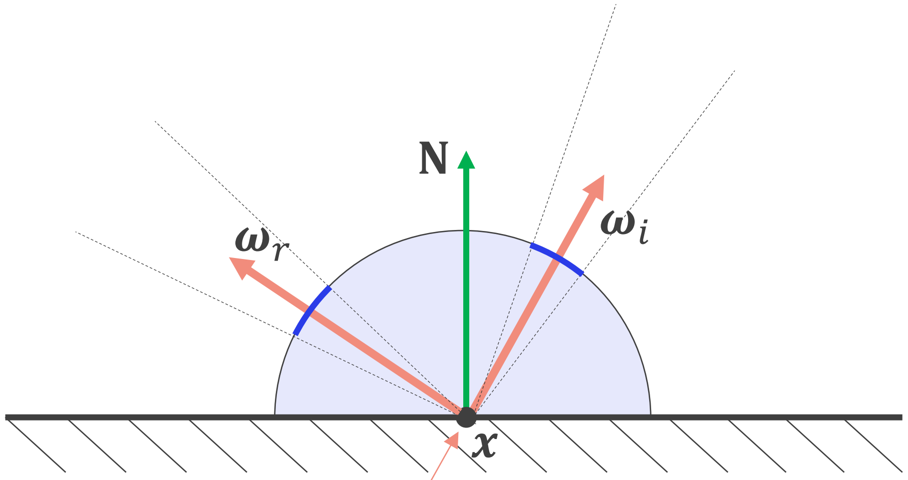
- Light is measured in terms of energy per unit area (W/m^2)
  - Radiance: amount of light from a single direction ($W/sr\ m^2$)
  - Irradiance: amount of light from possibly all directions (W/m^2)

partial radiance: light **reflected** in the infinitely small cone

$$f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) = \frac{dL_r(\omega_r)}{dE_i(\omega_i)} = \frac{dL_r(\omega_r)}{L_i \cos\theta \, d\omega_i}$$

partial irradiance: light **incoming** from the infinitely small cone

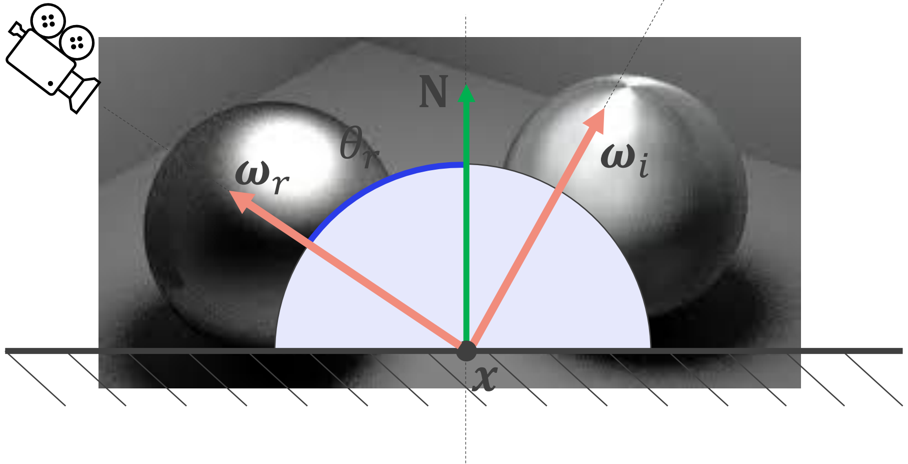
radiance from the solid angle projected onto the surface



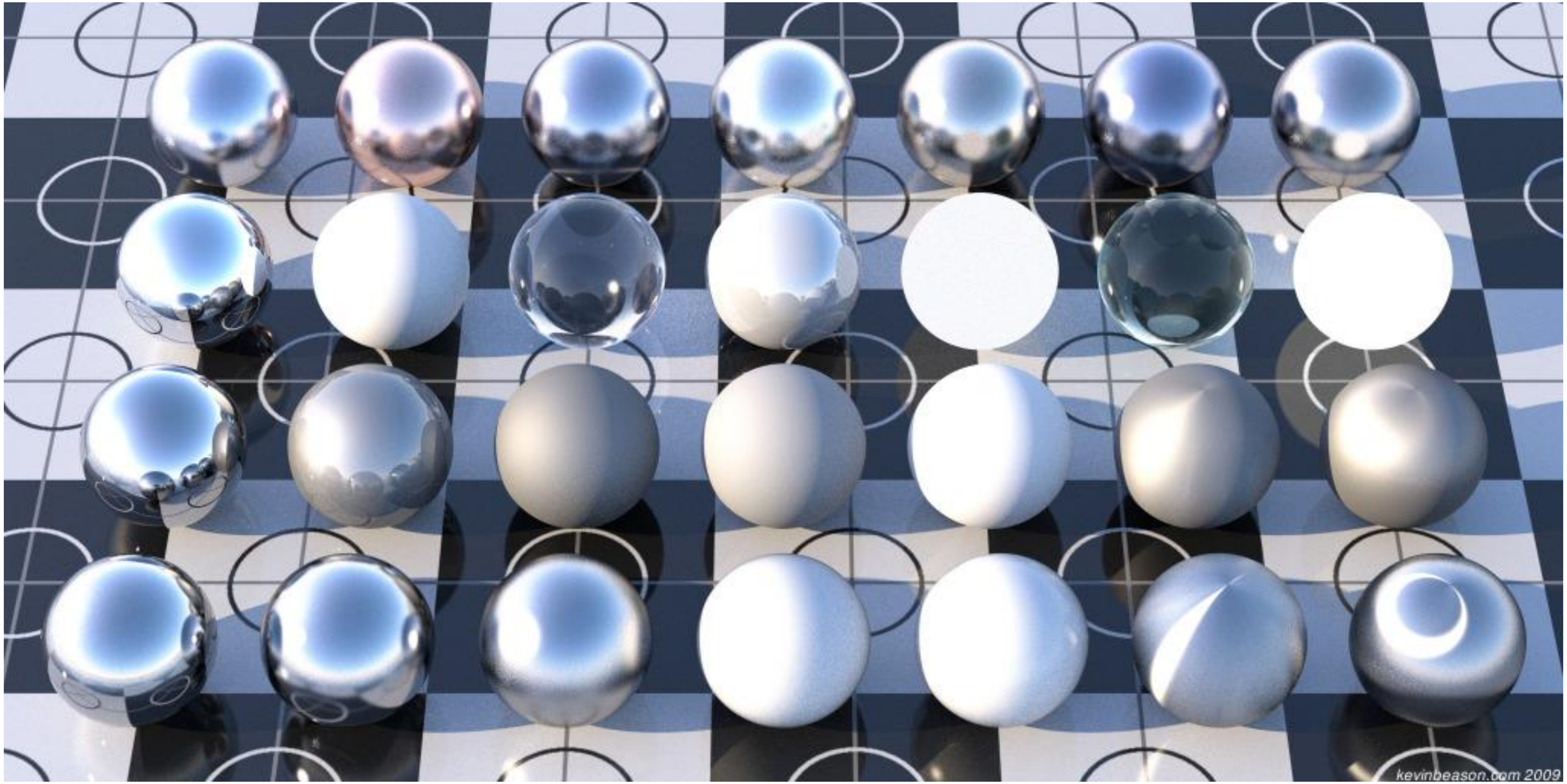
infinitely small part of the surface

Bidirectional Reflectance Distribution Function (BRDF)

- Properties of BRDF $f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r)$:
  - **Energy conservation**: A surface cannot reflect more light than falls on it form $\boldsymbol{\omega}_i$
    - For each $\boldsymbol{\omega}_i$ we have $\int_\Omega f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) \cos\theta_r \, d\boldsymbol{\omega}_r \leq 1$
  - **(Helmholtz) Reciprocity principle**: $f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) = f_r(\boldsymbol{x}, \boldsymbol{\omega}_r, \boldsymbol{\omega}_i)$
    - The value $f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r)$ remains the same if we swap $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_r$
  - **Isotropic**: The value $f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r)$ remains the same, if we rotate the surface about the normal at $\boldsymbol{x}$ by any angle
    - Example: smooth plastic
  - **Anisotropic**: The value $f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r)$ change, if we rotate the surface about the normal at $\boldsymbol{x}$ by some angle
    - Example: Brushed metal, hair

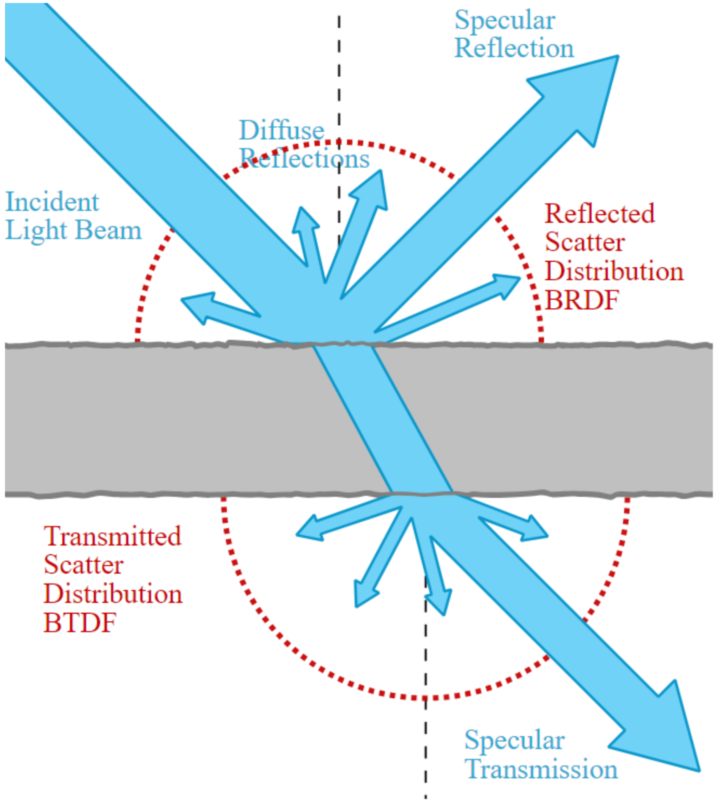


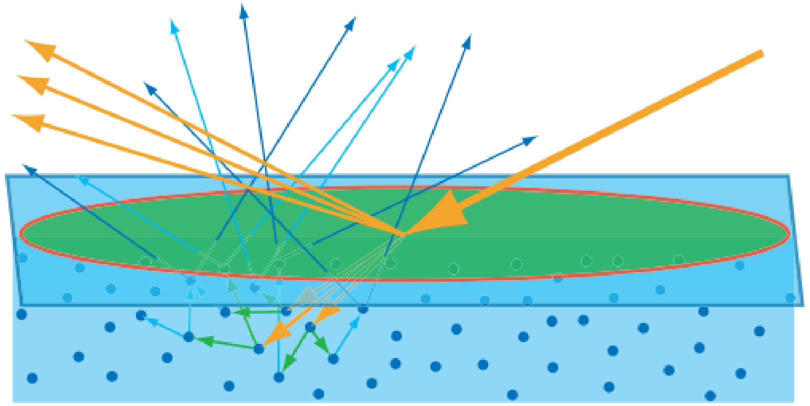
Surface Appearance



Distribution Functions

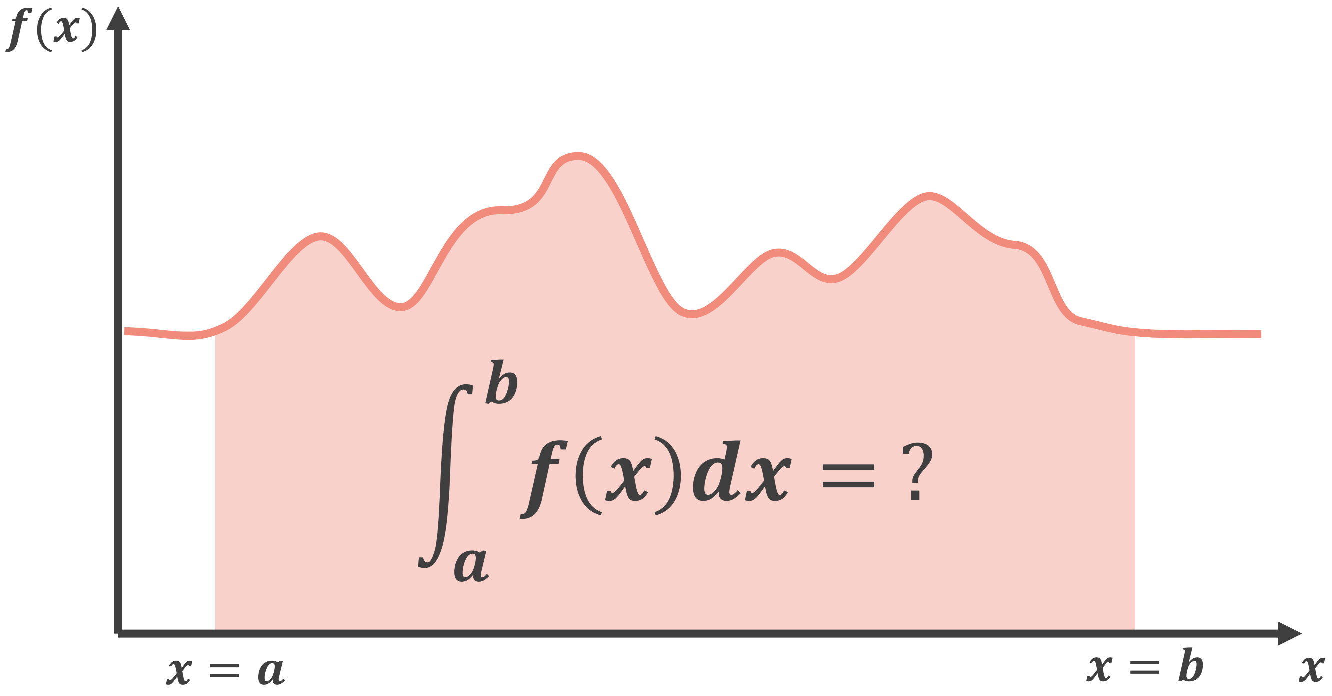
- Bidirectional transmittance distribution function (**BTDF**)
  - Like BRDF but for the opposite side of the surface.
- **BSSRDF + BSSTDF**
  - Like BRDF and BTDF but include subsurface scattering
- Bidirectional Scattering Distribution Function (**BSDF**)
  - Superset and the generalization of the BSSRDF and BSSTDF
  - When subsurface is omitted then BSDF = BRDF + BTDF





Evaluating Rendering Function

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \int_\Omega f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \, d\boldsymbol{\omega}_i$$

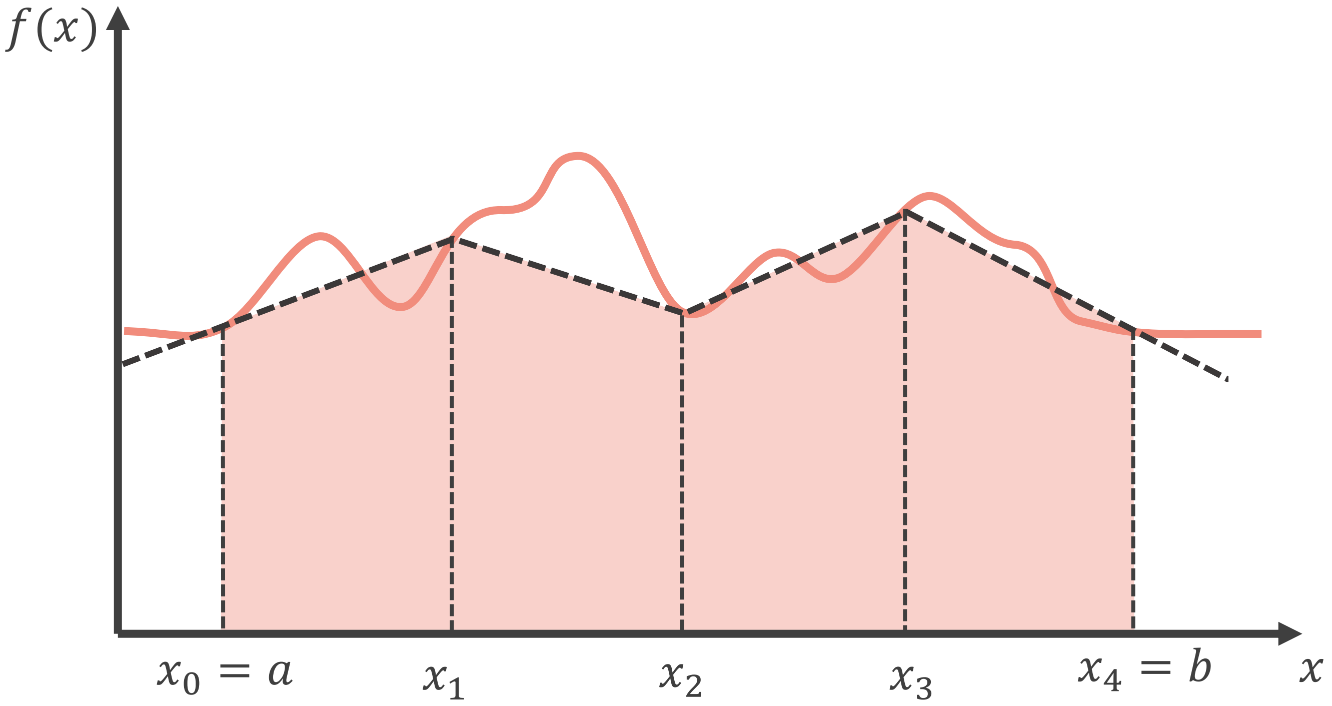


$$\int_a^b f(x)dx = ?$$

Trapezoid Rule

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} \left[\frac{f(x_{i-1}) + f(x_i)}{2} \frac{b-a}{n} \right] = \frac{b-a}{n} \sum_{i=1}^{n} \left[\frac{f(x_{i-1}) + f(x_i)}{2} \right]$$

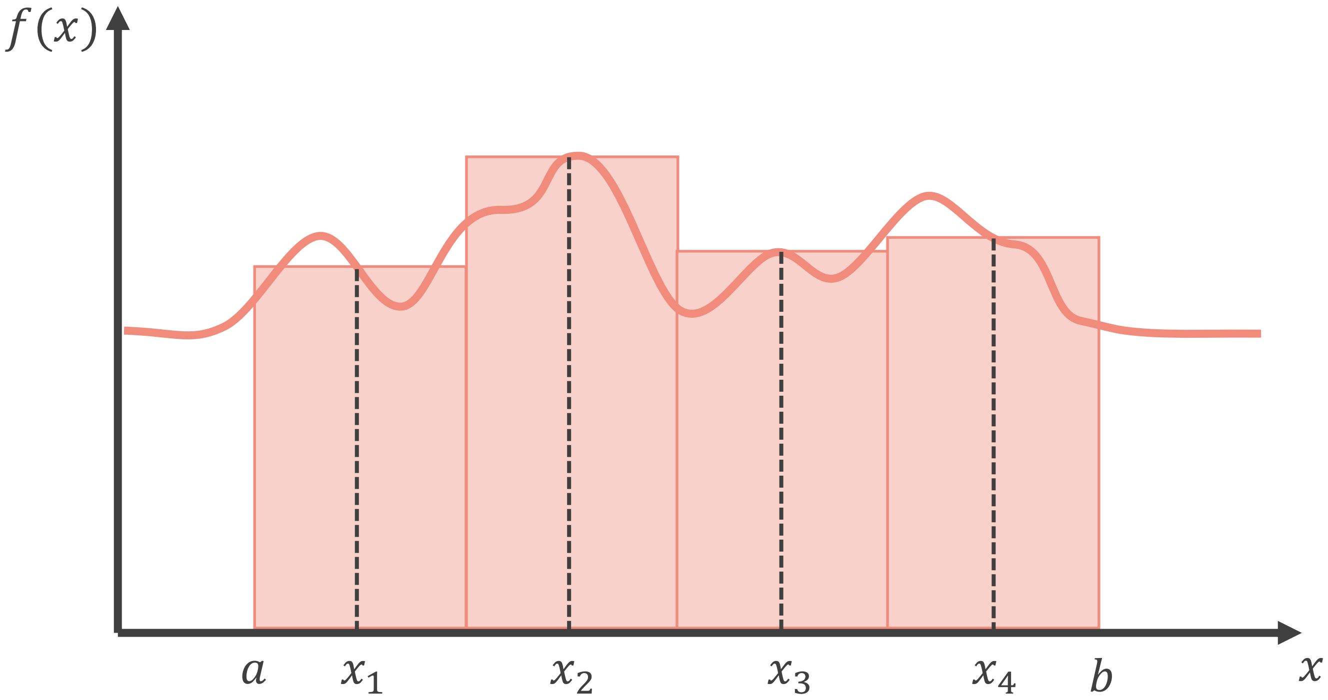
$$\Delta x = \frac{b-a}{n}$$



Rectangle Rule

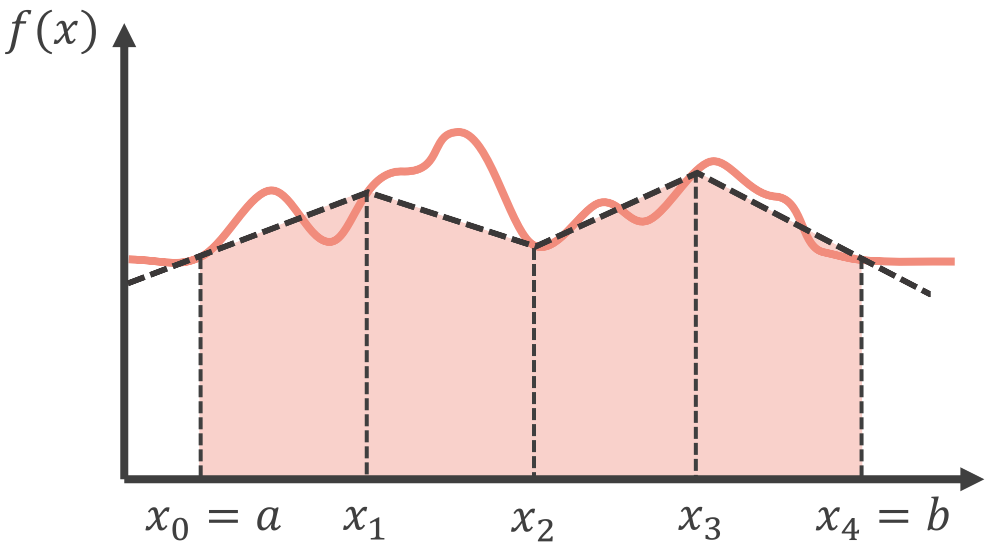
$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} \left[f(x_i) \frac{b-a}{n} \right] = \frac{b-a}{n} \sum_{i=1}^{n} f(x_i)$$

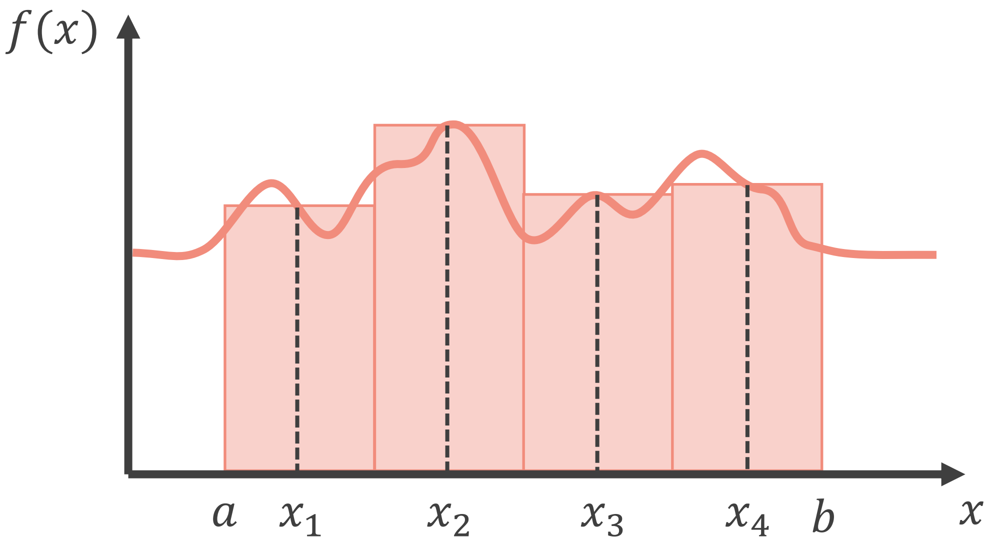
$$\Delta x = \frac{b-a}{n}$$



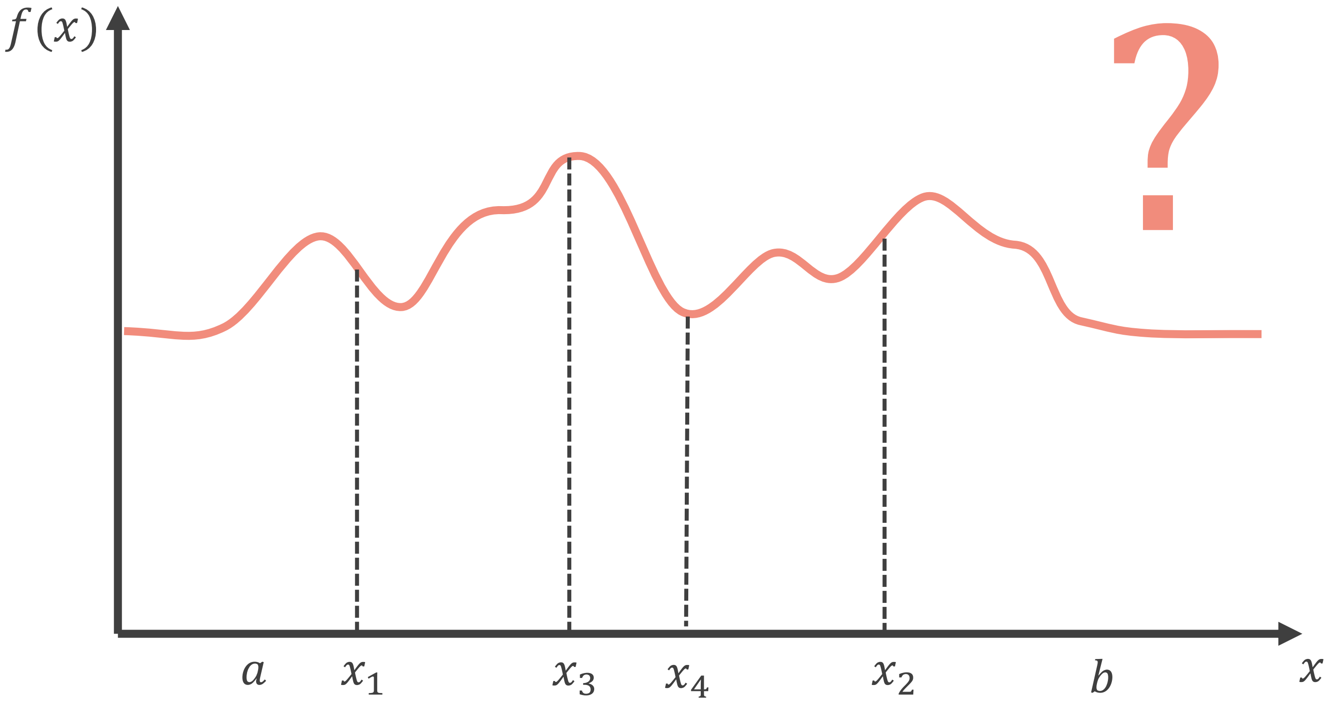
Disadvantages

- Computationally expensive and the complexity increases with dimensions

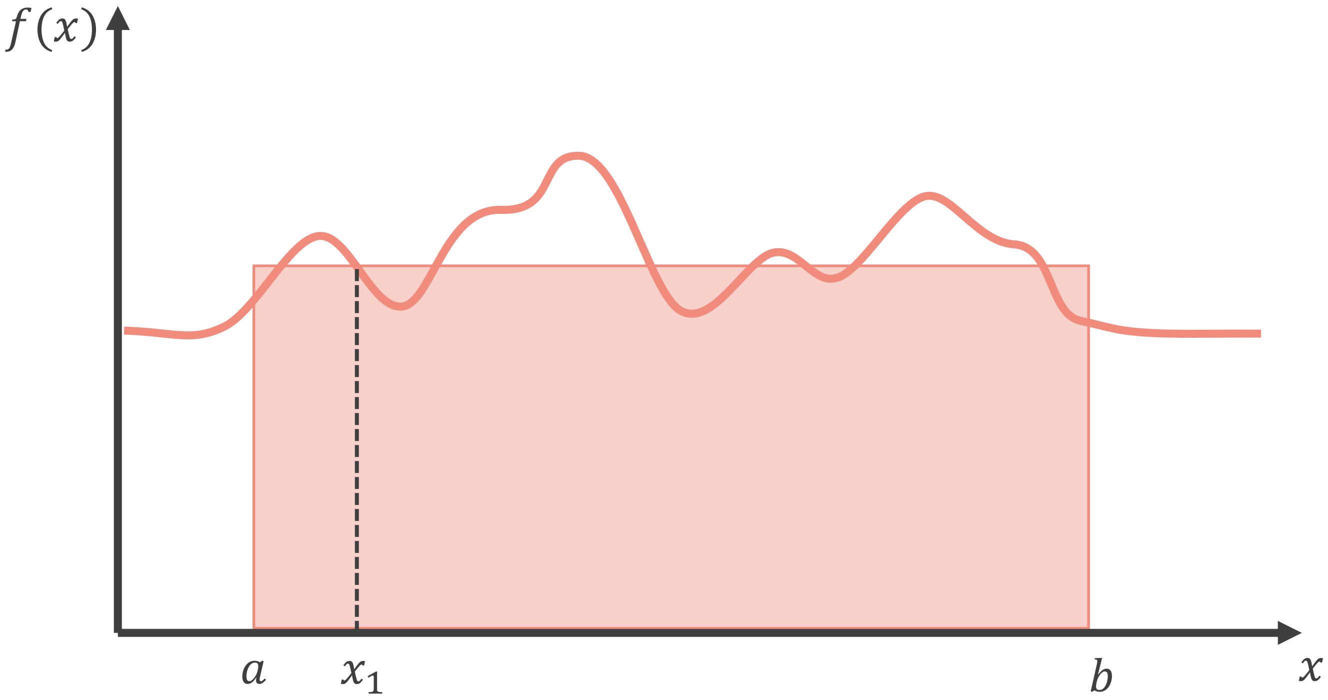




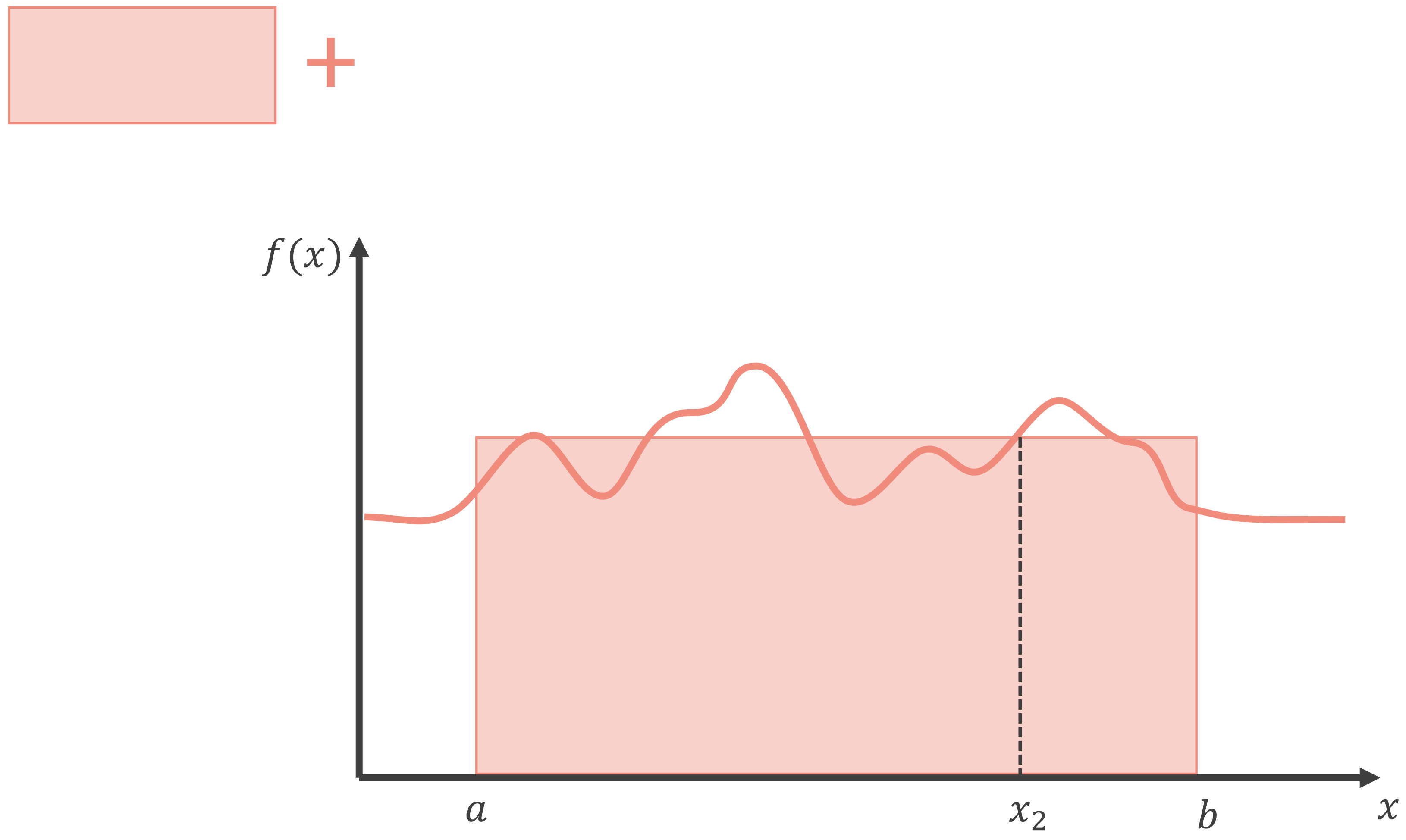
Random Sampling



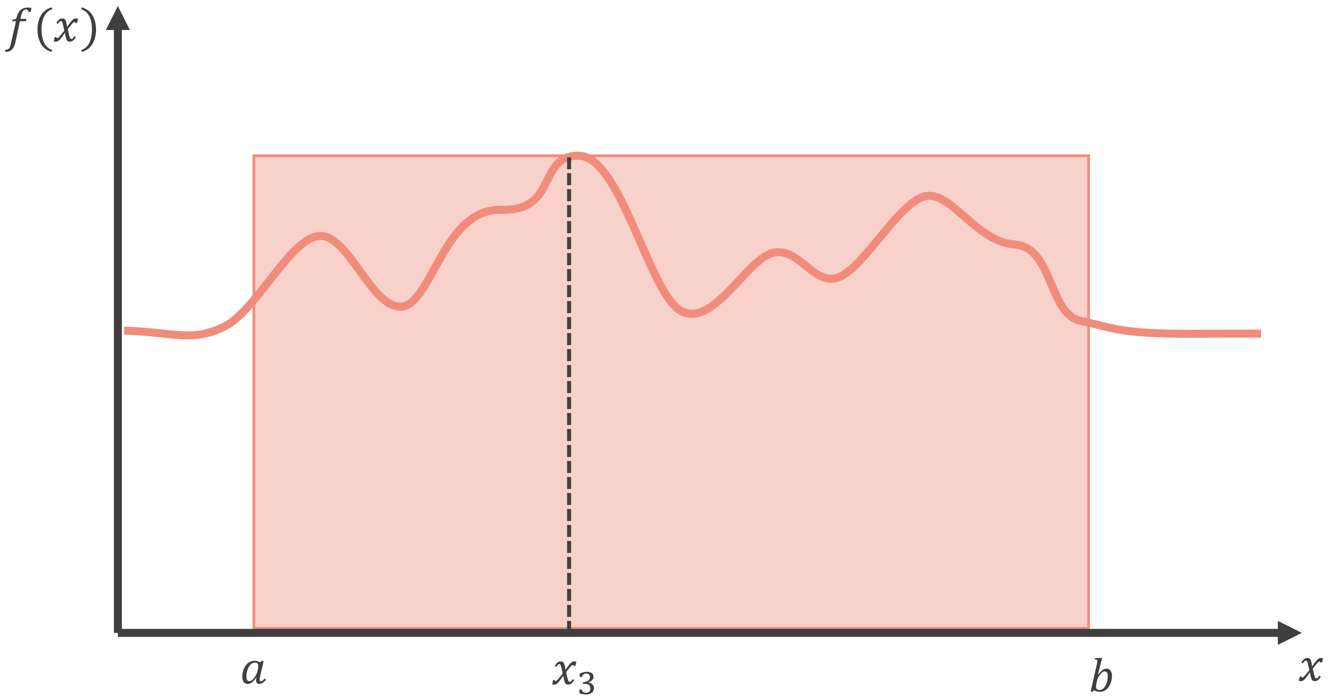
Random Sampling



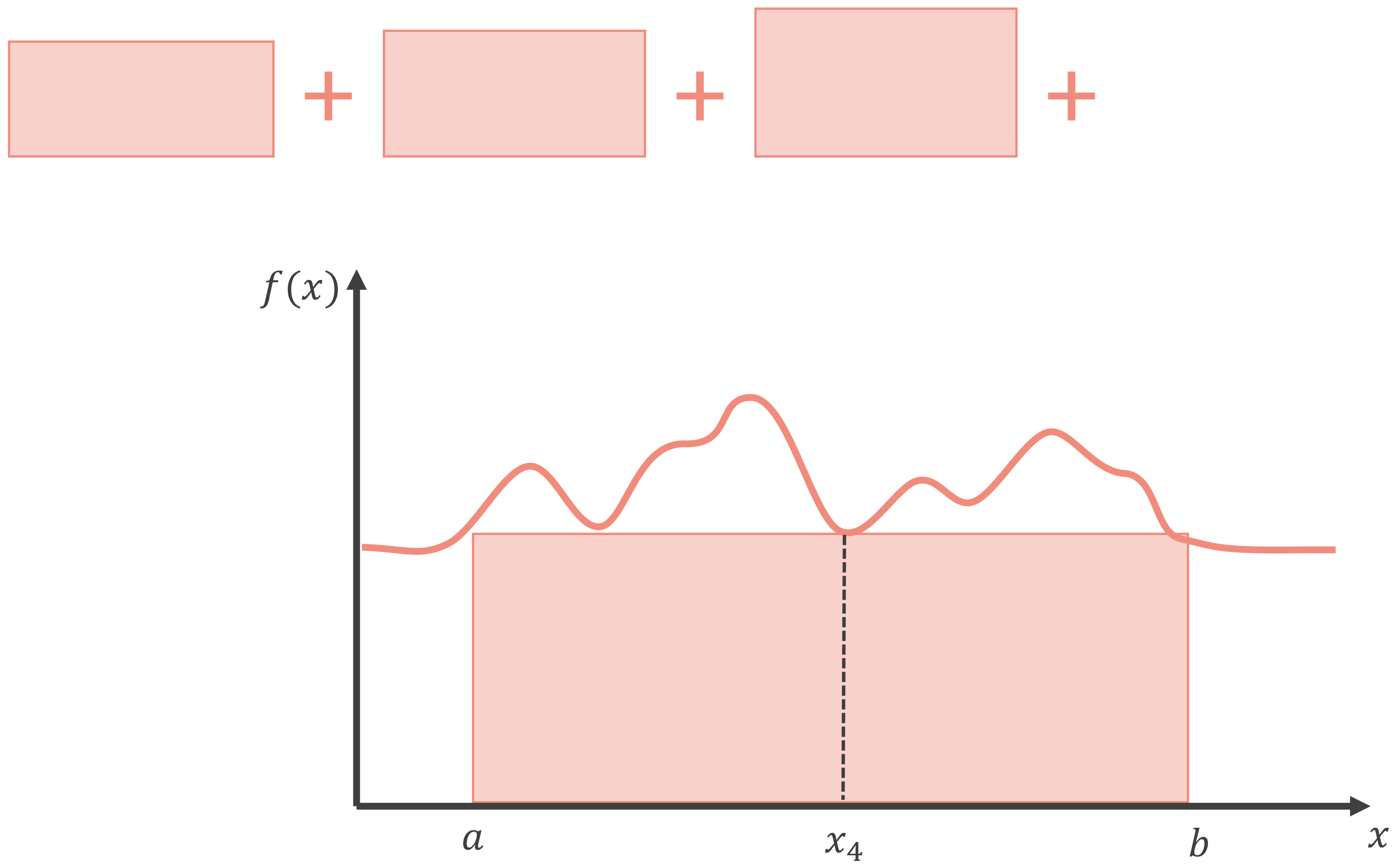
Random Sampling



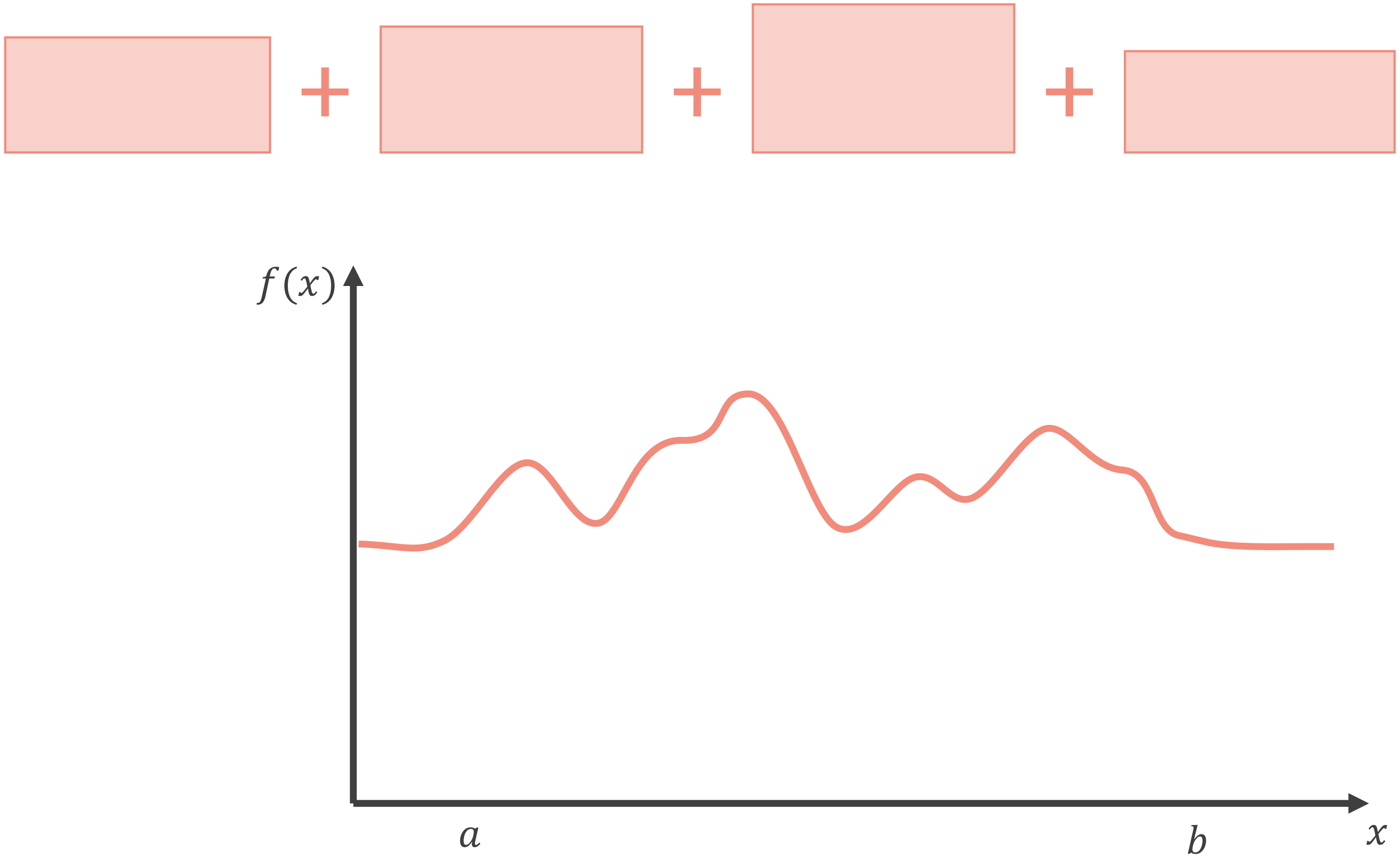
Random Sampling



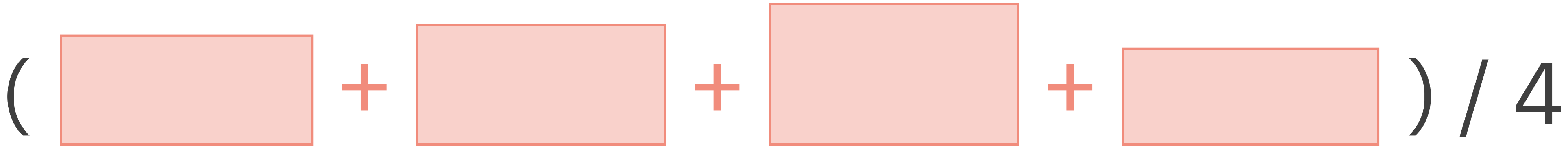
Random Sampling

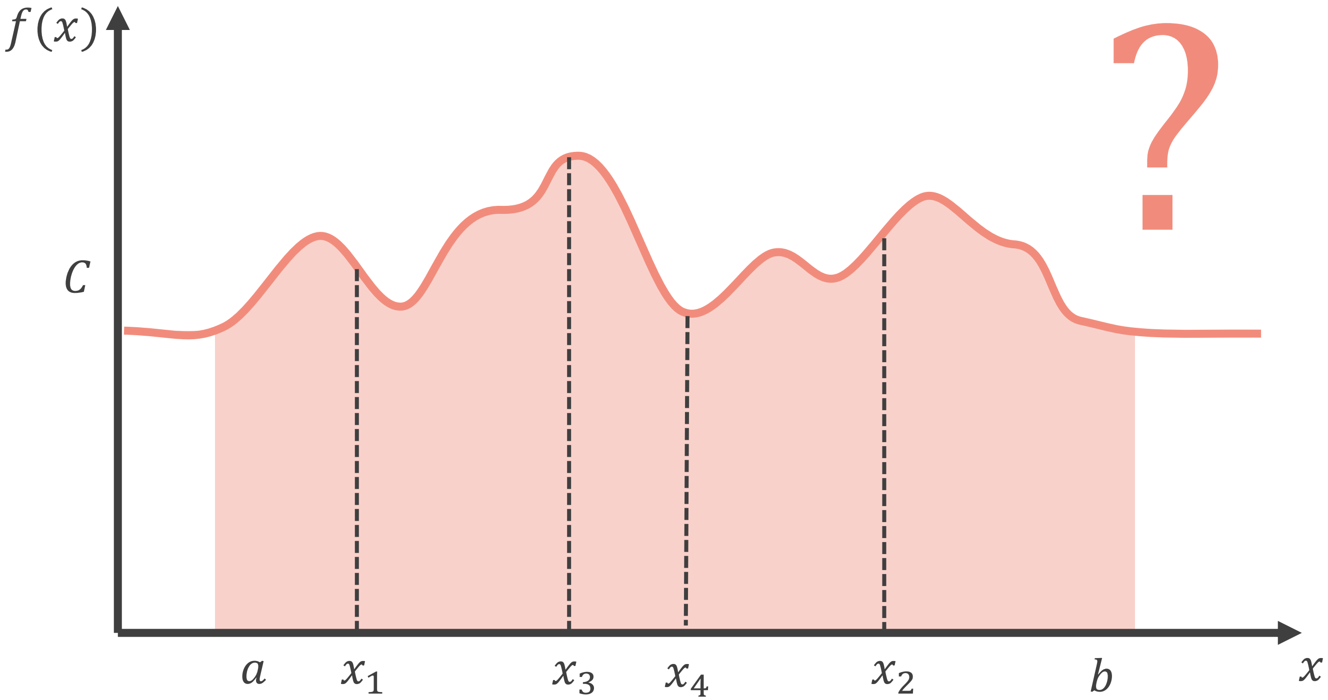


Random Sampling



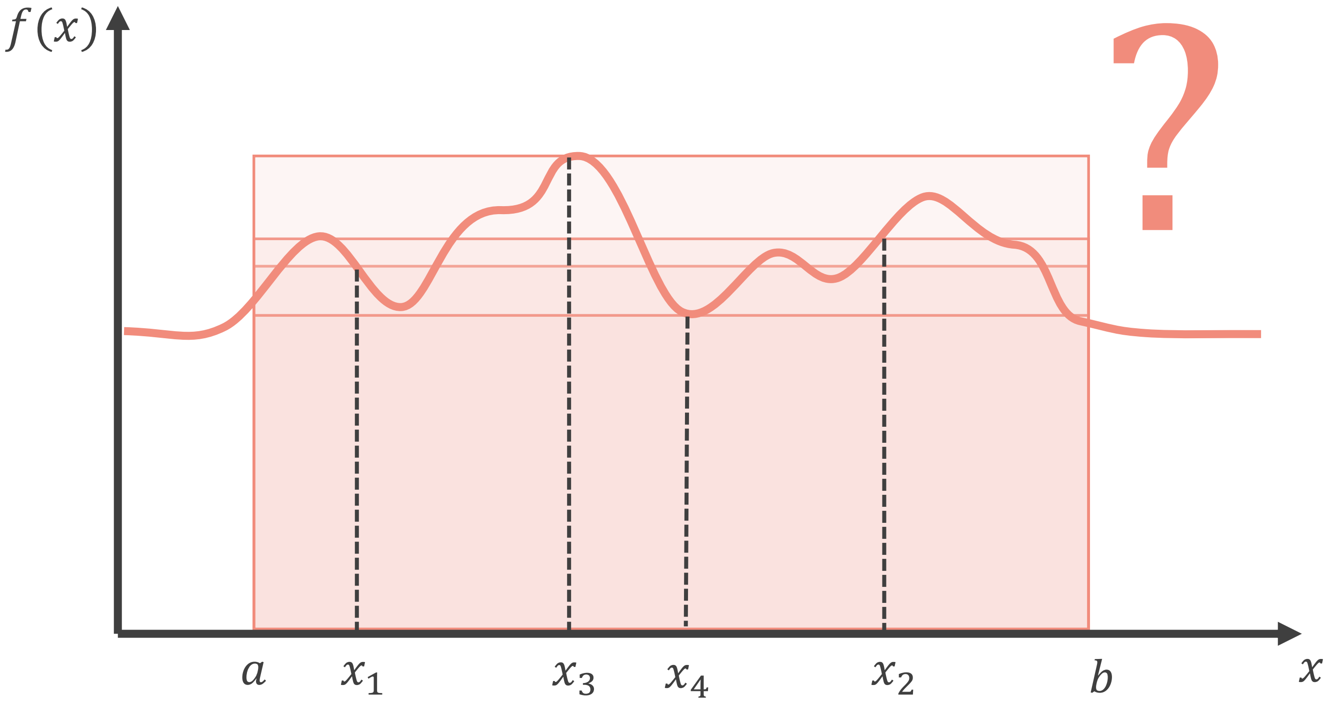
Random Sampling





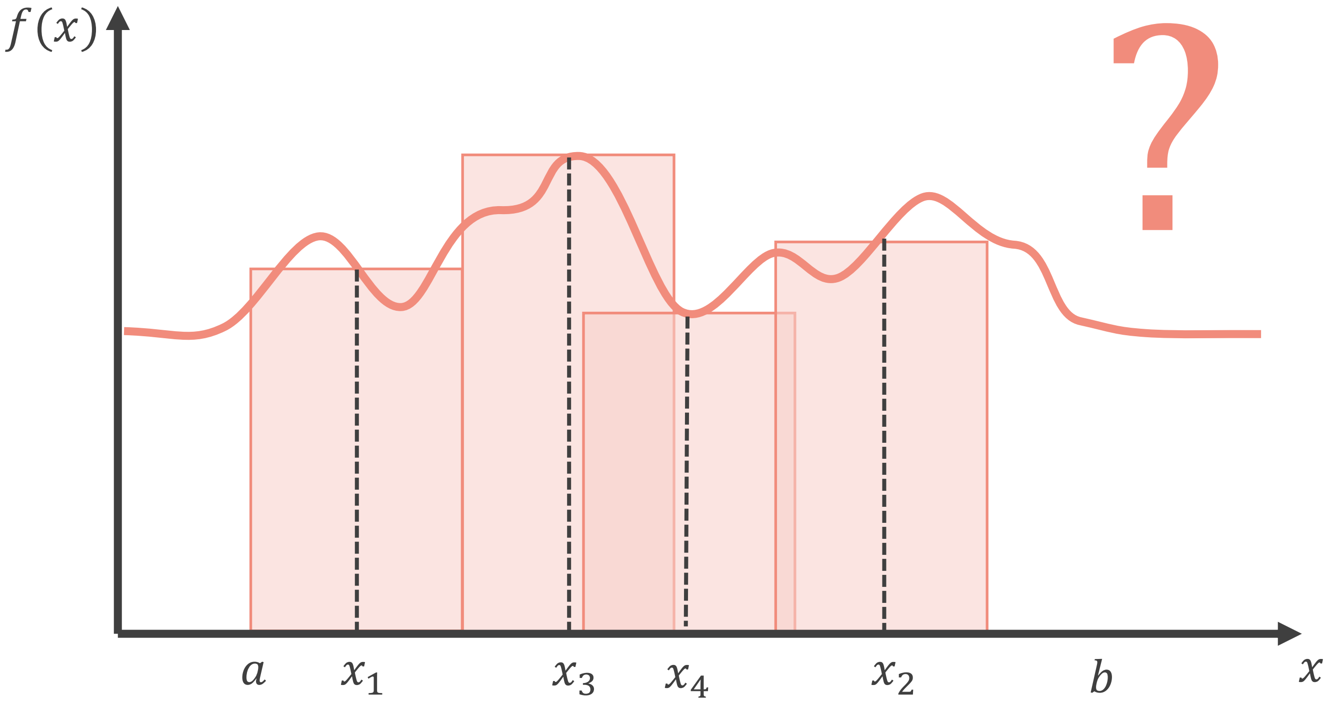
Monte Carlo Estimator

$$MCE(X) = \frac{1}{n}\sum_{i=1}^{n} f(X_i)(b-a)$$



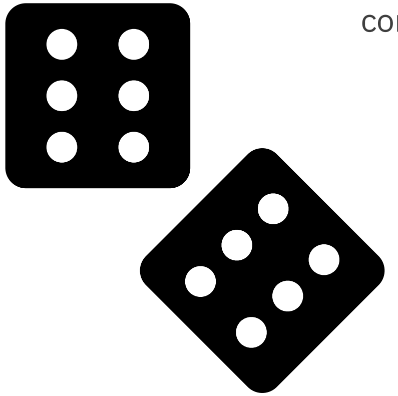
Monte Carlo Estimator

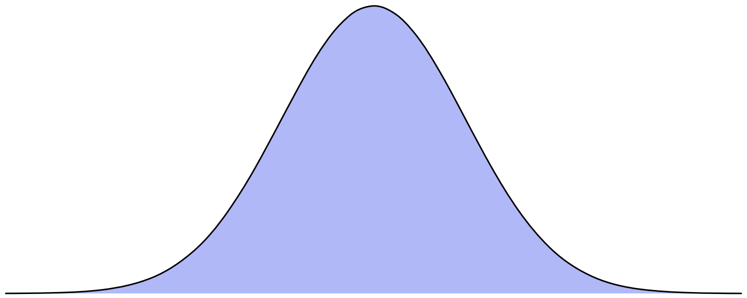
$$MCE(X) = \frac{1}{n}\sum_{i=1}^{n} f(X_i)(b-a) = \frac{b-a}{n}\sum_{i=1}^{n} f(X_i)$$



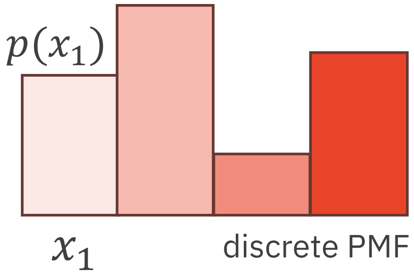
Random Variable

- **Intuition: Random variable X represents potential values for a random process**
  - X_i denotes i^th realization of the random variable X
  - x_i denotes the actual value of the X_i
- Probability mass/density function (PMF/PDF) $X \sim p(x)$ describes relative probability that a random process chooses value x
  - Example: unbiased die
    - All values are equally likely (uniform PMF)
    - $p(1) = p(2) = p(3) = p(4) = p(5) = p(6) = \frac{1}{6}$
  - Properties PMF:
    - $p(x_i) \geq 0 \; and \; \sum_{i=1}^{k} p(x_i) = 1$
  - Properties PDF:
    - $p(x) \geq 0 \; and \; \int_D p(x)dx = 1$





continuous PDF



Expected Value

- **Intuition: what value does a random variable take, on average?**
- Example:
  - coin with heads = 1, tails = 0
  - fair coin with probability ½ for each outcome
  - expected value: $½ * 1 + ½ * 0 = ½$

expected value of random variable X

number of possible outcomes

$$E[X] = \sum_{i=1}^{k} x_i p_i$$

value of ith outcome

probability of ith outcome

PDF

$$E[X] = \int_D x\, p(x) dx$$

values of the random variable over domain D

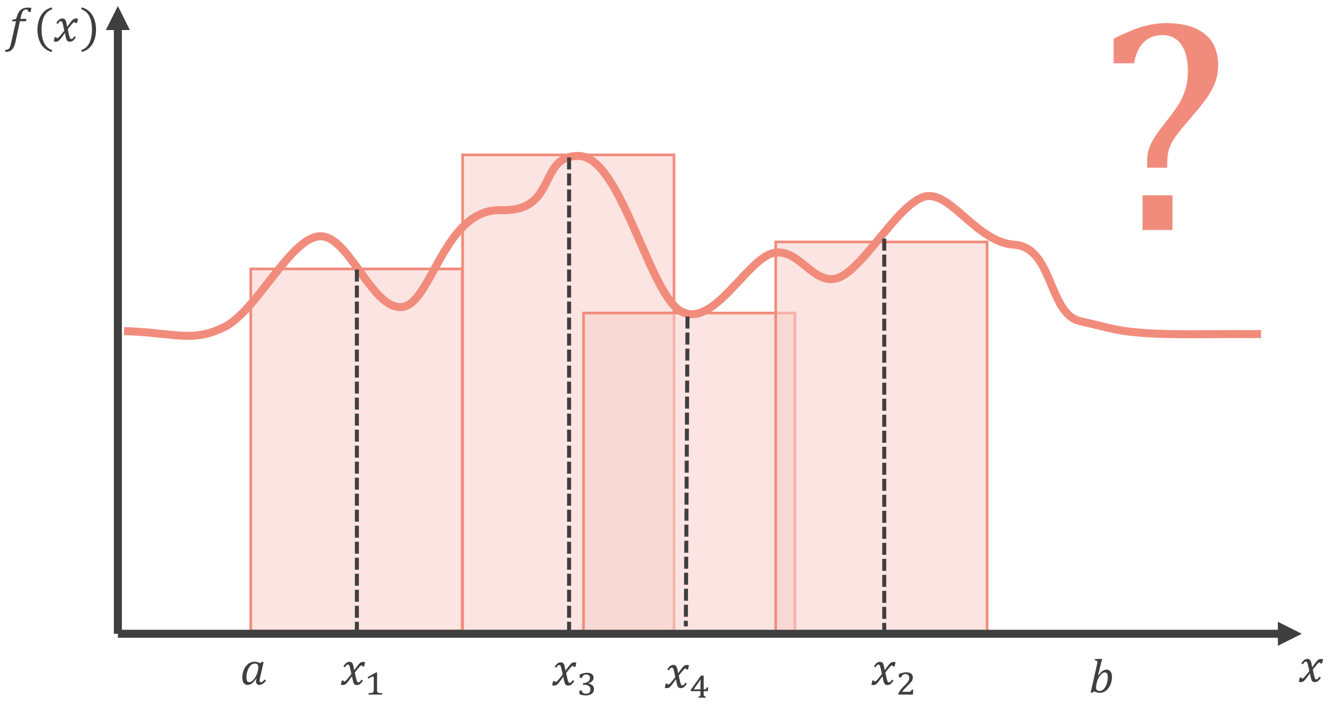
Expected Value of Function

- Applying a function $f\colon \mathbb{R} \to \mathbb{R}$ to a random variable results in a new random variable

$$E[X] = \int_D x\, p(x)\, dx \longrightarrow E[f(X)] = \int_D f(x)\, p(x)\, dx$$

Monte Carlo Estimator

$$MCE(X) = \frac{1}{n}\sum_{i=1}^{n} f(X_i)(b-a) = \frac{b-a}{n}\sum_{i=1}^{n} f(X_i)$$



Monte Carlo Integration

$$MCE(X) = \frac{1}{n}\sum_{i=1}^{n} f(X_i)(b-a) = \frac{b-a}{n}\sum_{i=1}^{n} f(X_i)$$

$E[aX] = aE[X]$

$E\left[\sum_i X_i\right] = \sum_i E[X_i]$

$$E[MCE(X)] = E\left[\frac{b-a}{n}\sum_{i=1}^{n} f(X_i)\right]$$

think about this as $E[f(X)]$ since in each step we are drawing from the same random variable

$$E[MCE(X)] = \frac{b-a}{n}\sum_{i=1}^{n} E[f(X_i)]$$

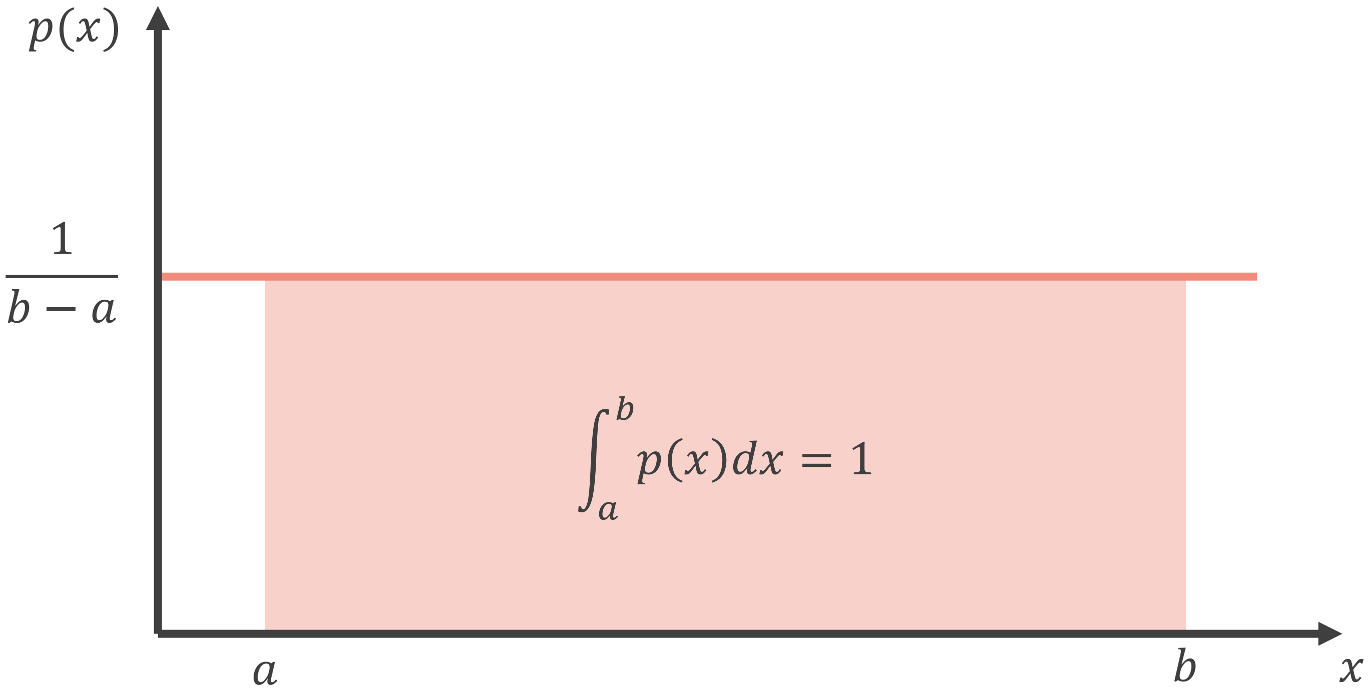
$E[f(X)] = \int_a^b f(x)\,p(x)dx$

$$E[MCE(X)] = \frac{b-a}{n}\sum_{i=1}^{n} \int_a^b f(x)\,p(x)\,dx$$

Monte Carlo Integration

Assuming uniform distribution $X \sim p(x)$

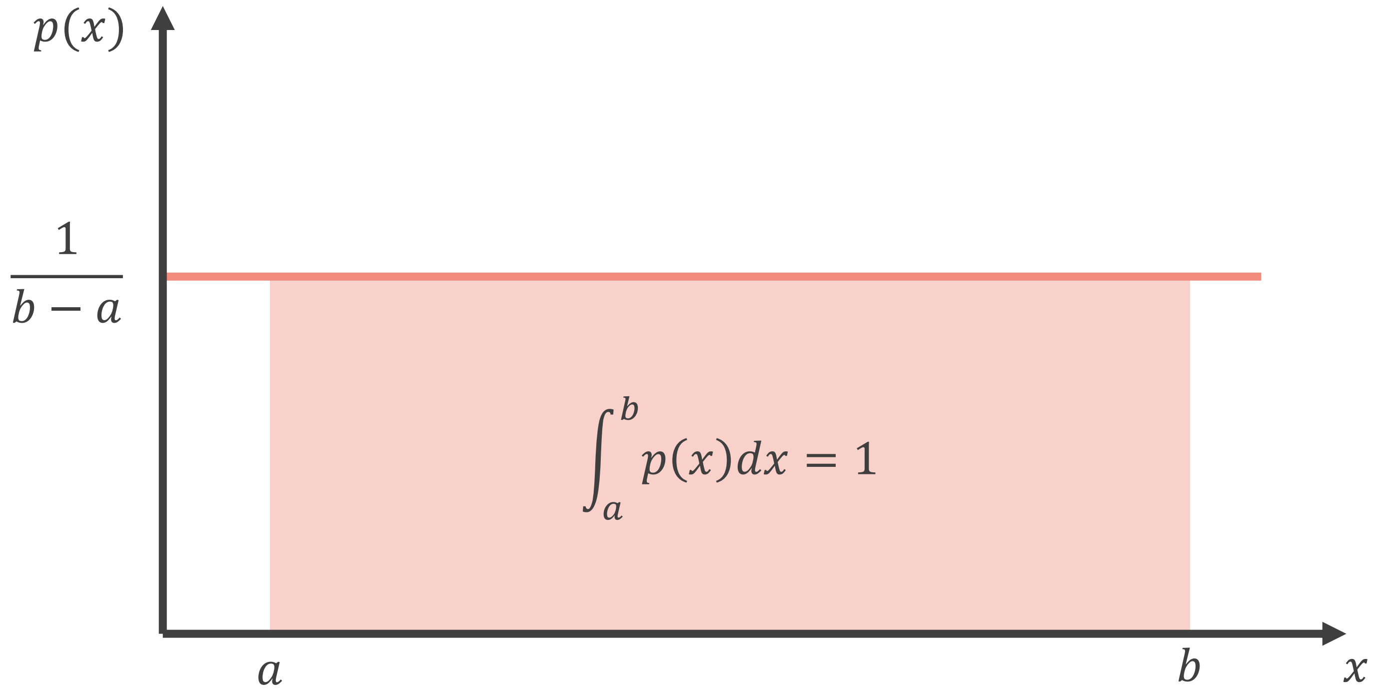
$$E[MCE(X)] = \frac{b-a}{n} \sum_{i=1}^{n} \int_a^b f(x)\, p(x)\, dx$$



Monte Carlo Integration

Assuming uniform distribution $X \sim p(x)$

$$E[MCE(X)] = \frac{b-a}{n} \sum_{i=1}^{n} \int_a^b f(x) \frac{1}{b-a} dx$$



Monte Carlo Integration

Assuming uniform distribution $X \sim p(x)$

$$E[MCE(X)] = \frac{b-a}{n}\sum_{i=1}^{n}\int_a^b f(x)\frac{1}{b-a}dx$$

$$E[MCE(X)] = \frac{1}{n}\sum_{i=1}^{n}\int_a^b f(x)\,dx$$

$$\frac{1}{n}\sum_{i=1}^{n} A = A$$

$$E[MCE(X)] = \int_a^b f(x)\,dx$$

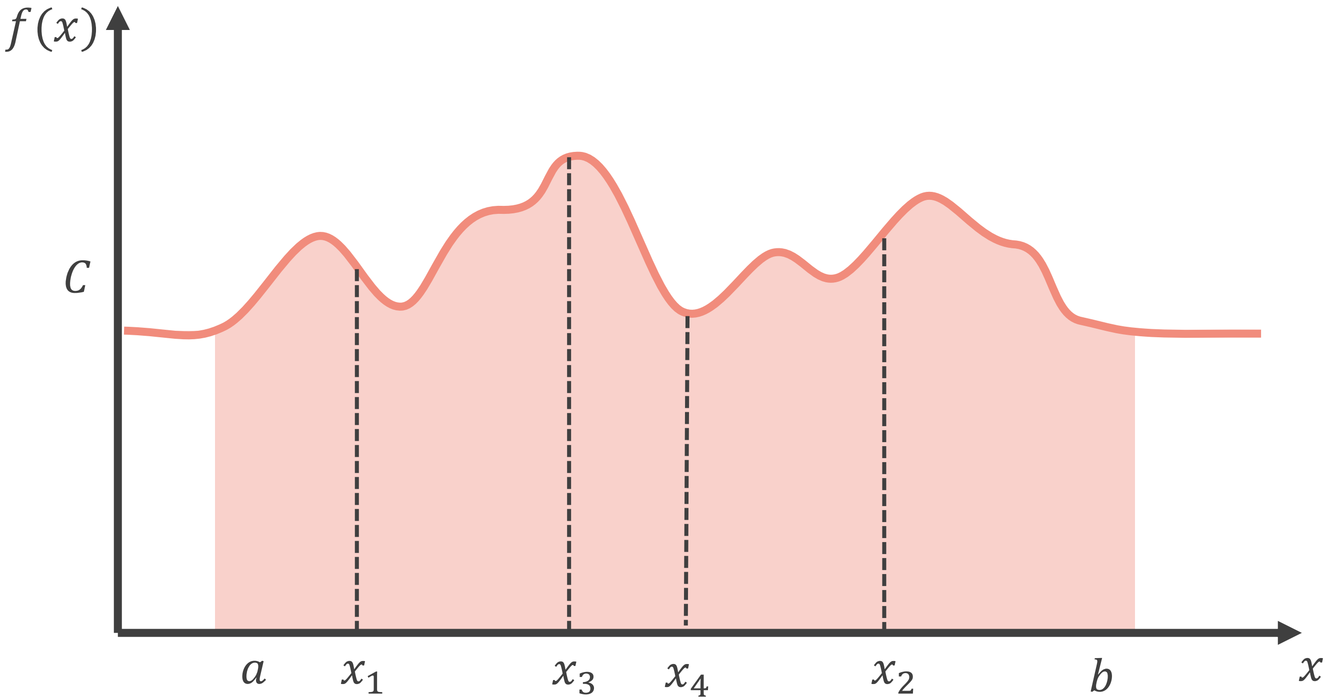
Monte Carlo Integration

Assuming uniform distribution $X \sim p(x)$

$$E[f(X)] = \int_a^b f(x)p(x)dx$$

does not really help us

$$\int_a^b f(x)dx = E[MCE(X)] = E\left[\frac{b-a}{n}\sum_{i=1}^{n} f(X_i)\right] = \frac{b-a}{n}\sum_{i=1}^{n} E[f(X_i)]$$



Law of Large Numbers

- For any random variable, the average value of n trials approaches the expected value as we increase n

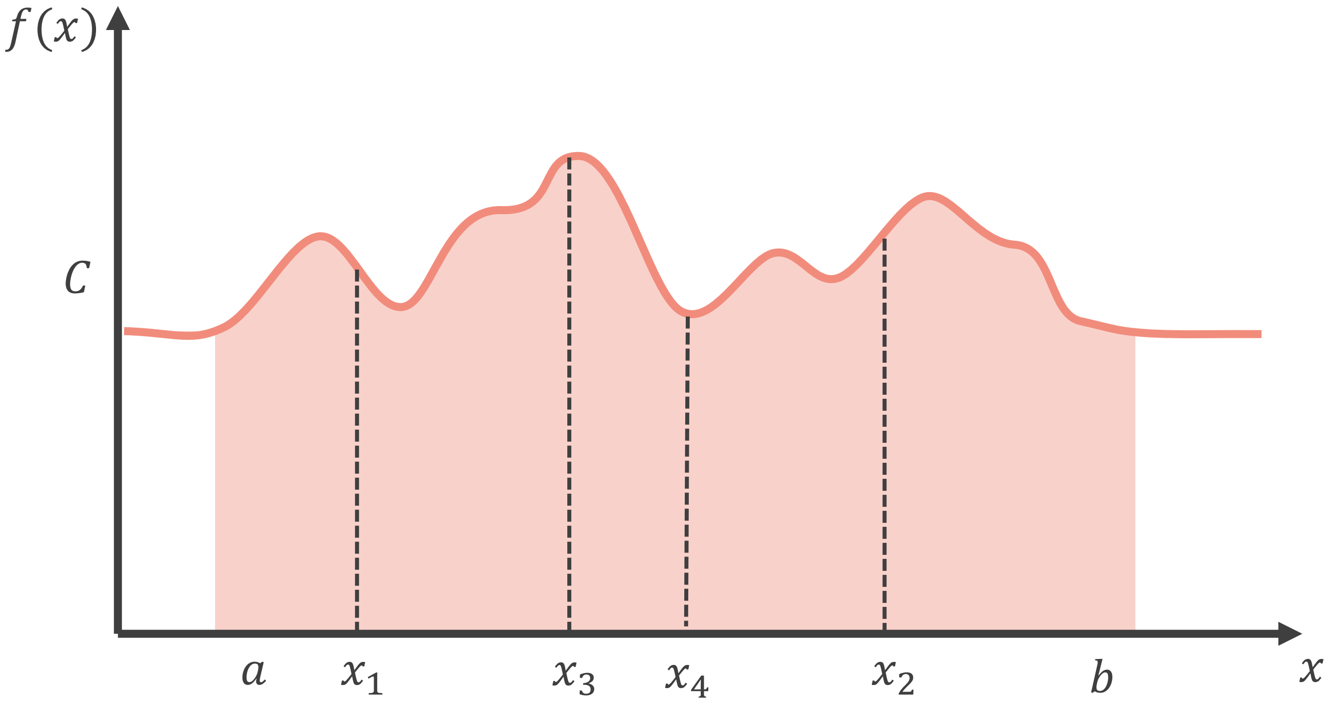
number of trials

$$E[X] = \lim_{n \to \infty} \left(\frac{1}{n} \sum_{i=1}^{n} X_i \right)$$

Monte Carlo Integration

Assuming uniform distribution $X \sim p(x)$

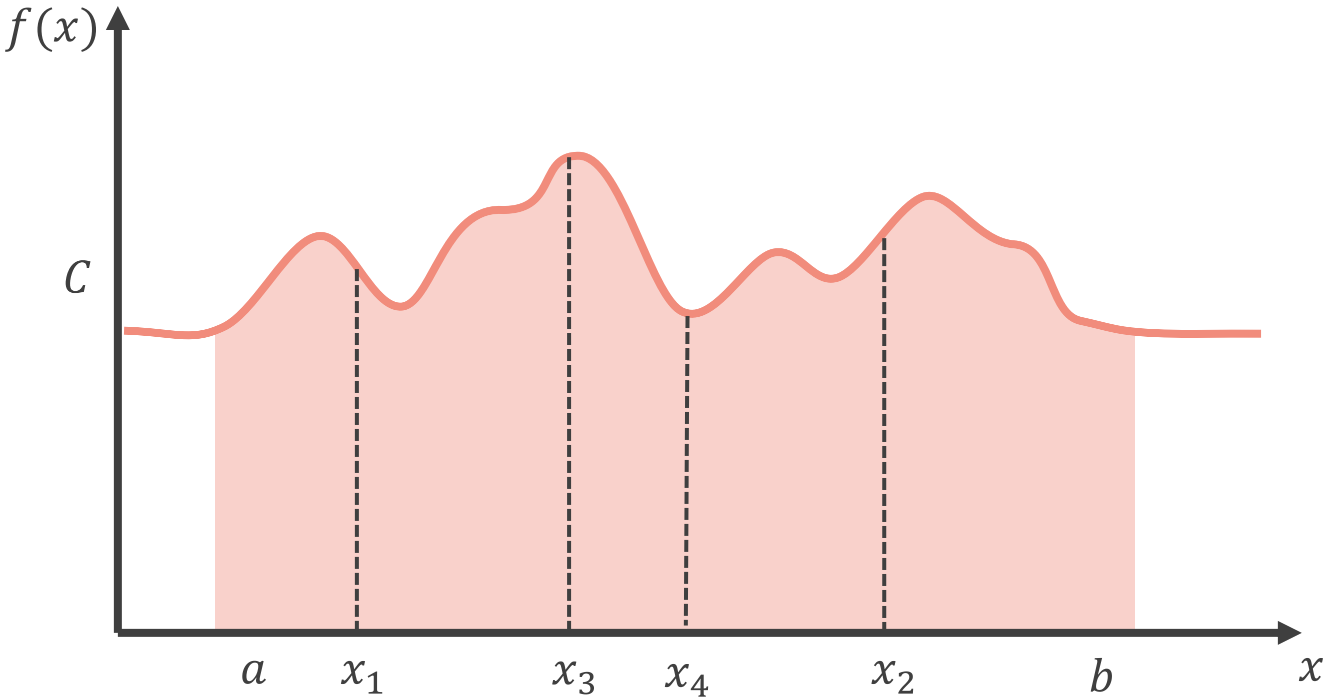
$$\int_a^b f(x)dx \approx \frac{b-a}{n}\sum_{i=1}^{n} f(X_i)$$



Monte Carlo Integration

Assuming **arbitrary** distribution $X \sim p(x)$

$$\int_a^b f(x)dx = \int_a^b \frac{f(x)}{p(x)} p(x)dx \; = ?$$



Monte Carlo Integration

Assuming **arbitrary** distribution $X \sim p(x)$

$$\int_a^b f(x)dx = \int_a^b \frac{f(x)}{p(x)} p(x)dx$$

$$g(x) = \frac{f(x)}{p(x)}$$

$$= \int_a^b g(x)p(x)dx$$

$$E[f(X)] = \int_a^b f(x)p(x)dx$$

$$= E[g(X)]$$

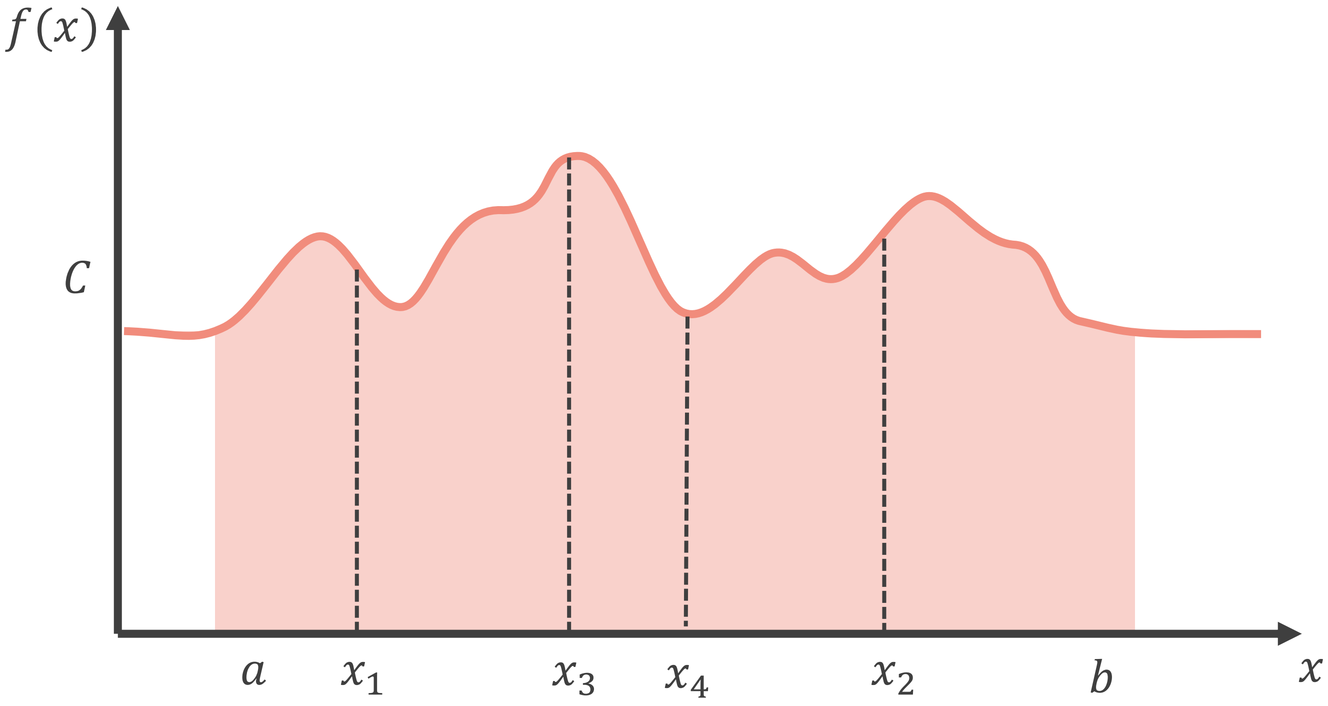
$$g(x) = \frac{f(x)}{p(x)}$$

$$= E\left[\frac{f(X)}{p(X)}\right]$$

Monte Carlo Integration

Assuming **arbitrary** distribution $X \sim p(x)$

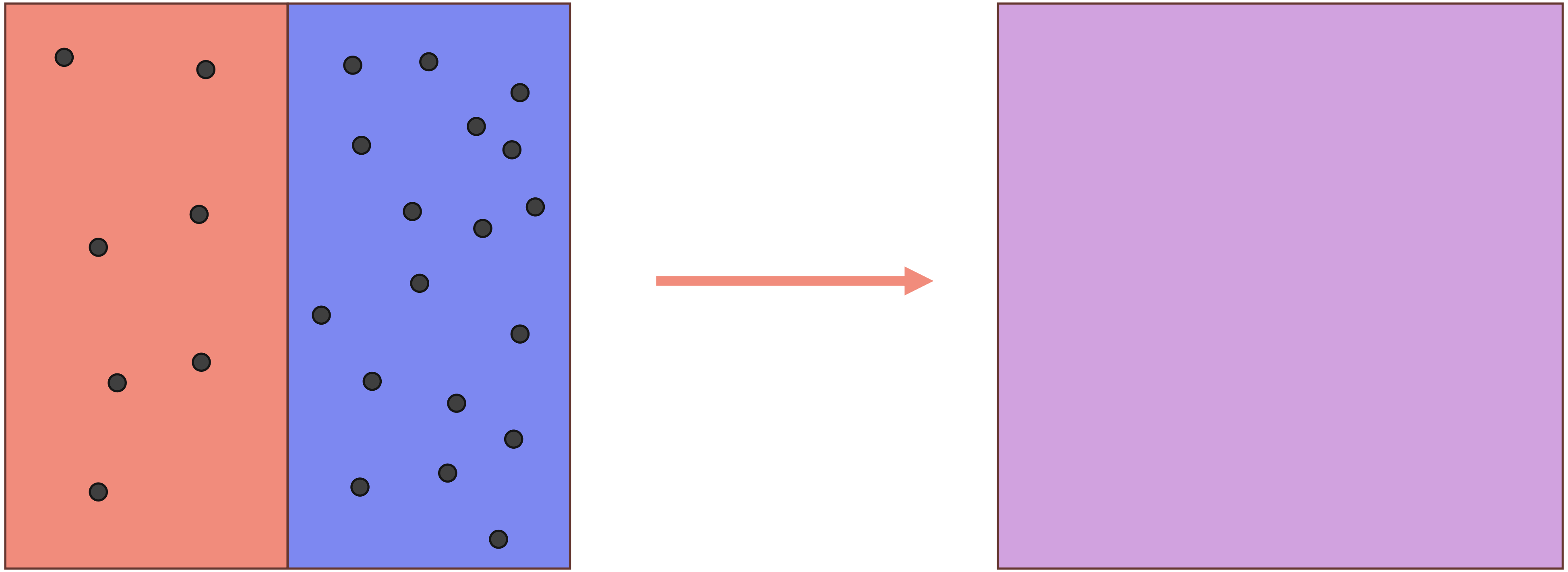
$$\int_a^b f(x)dx \approx \frac{1}{n}\sum_{i=1}^{n}\frac{f(X_i)}{p(X_i)}$$



Monte Carlo Integration

Assuming **arbitrary** distribution $X \sim p(x)$

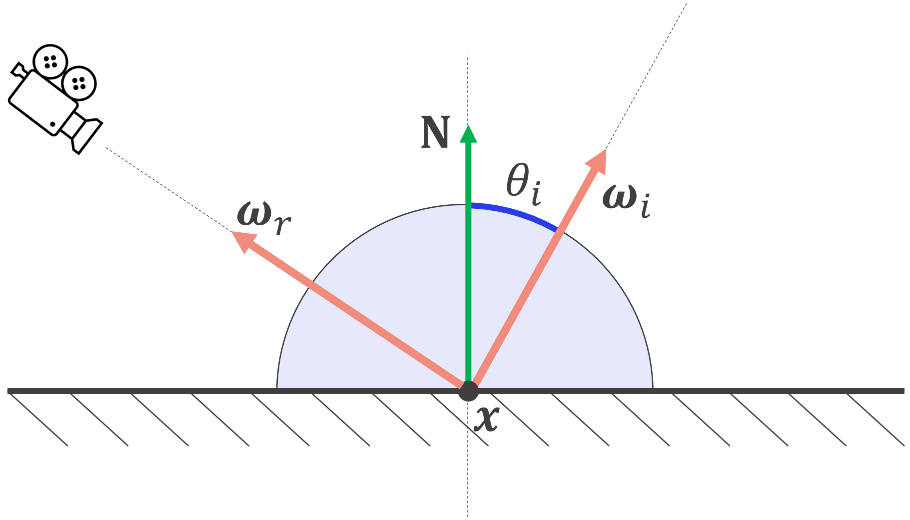
$$\int_a^b f(x)dx \approx \frac{1}{n}\sum_{i=1}^{n}\frac{f(X_i)}{p(X_i)}$$



Evaluating Rendering Function

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \int_\Omega f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \, d\boldsymbol{\omega}_i$$

$$\approx L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{2\pi}{n} \sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i$$

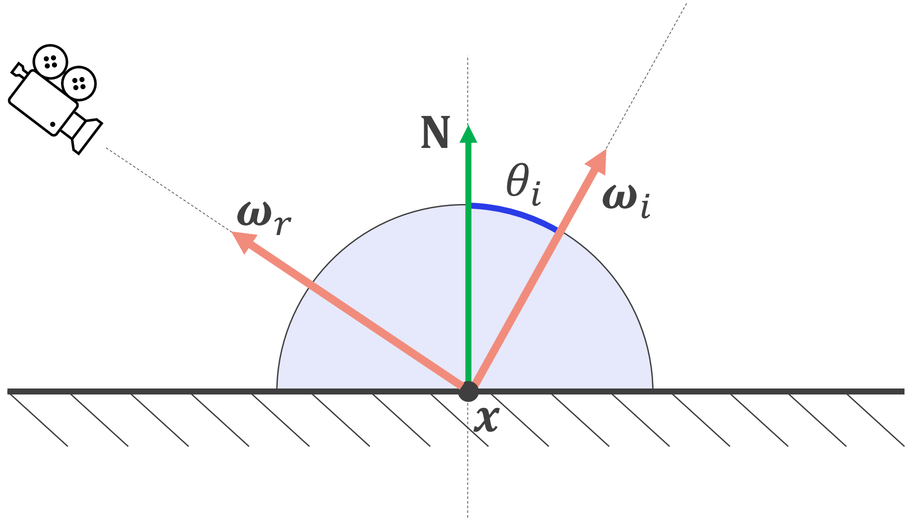


$$\int_a^b f(x)dx \approx \frac{b-a}{n} \sum_{i=1}^{n} f(X_i)$$

Evaluating Rendering Function

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \int_\Omega f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \, d\boldsymbol{\omega}_i$$

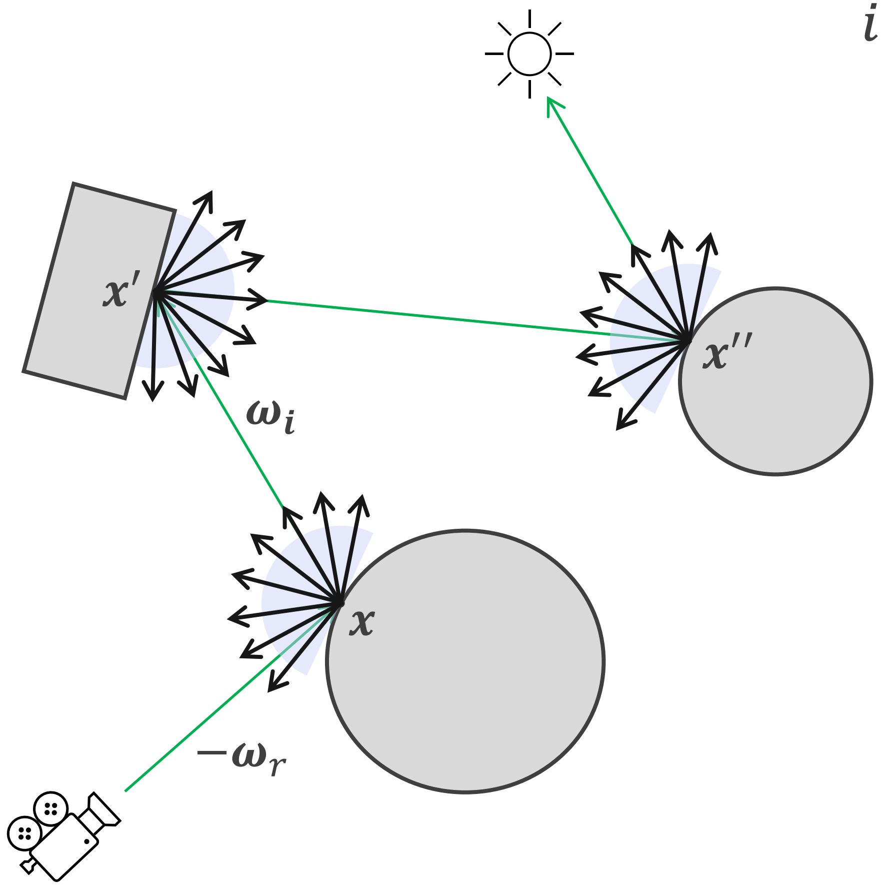
$$\approx L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} \frac{f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i}{p(\boldsymbol{\omega}_i)}$$



$$\int_a^b f(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} \frac{f(X_i)}{p(X_i)}$$

Evaluating Rendering Function

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$



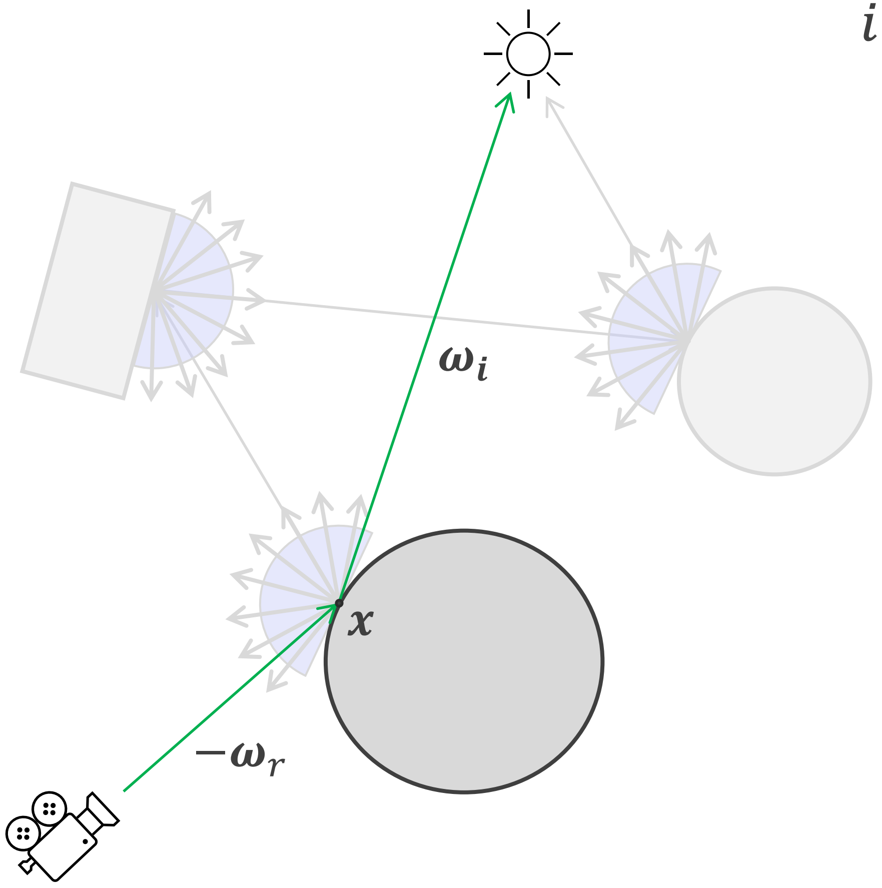
$$L_r(\boldsymbol{x}', \boldsymbol{\omega}_r') = L_e(\boldsymbol{x}', \boldsymbol{\omega}_r') + \frac{1}{n}\sum_{i=1}^{n} f_r'(\boldsymbol{x}', \boldsymbol{\omega}_i', \boldsymbol{\omega}_r')\, L_i(\boldsymbol{x}', \boldsymbol{\omega}_i') \cos\theta_i'\, d\boldsymbol{\omega}_i' \frac{1}{p(\boldsymbol{\omega}_i')}$$

$$L_r(\boldsymbol{x}'', \boldsymbol{\omega}_r'') = L_e(\boldsymbol{x}'', \boldsymbol{\omega}_r'') + \frac{1}{n}\sum_{i=1}^{n} f_r''(\boldsymbol{x}'', \boldsymbol{\omega}_i'', \boldsymbol{\omega}_r'')\, L_i(\boldsymbol{x}'', \boldsymbol{\omega}_i'') \cos\theta_i''\, d\boldsymbol{\omega}_i'' \frac{1}{p(\boldsymbol{\omega}_i'')}$$

$$L_r(\boldsymbol{x}''', \boldsymbol{\omega}_r''') = L_e(\boldsymbol{x}''', \boldsymbol{\omega}_r''') + \frac{1}{n}\sum_{i=1}^{n} f_r'''(\boldsymbol{x}''', \boldsymbol{\omega}_i''', \boldsymbol{\omega}_r''')\, L_i(\boldsymbol{x}''', \boldsymbol{\omega}_i''') \cos\theta_i'''\, d\boldsymbol{\omega}_i''' \frac{1}{p(\boldsymbol{\omega}_i''')}$$

Direct Illumination

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$



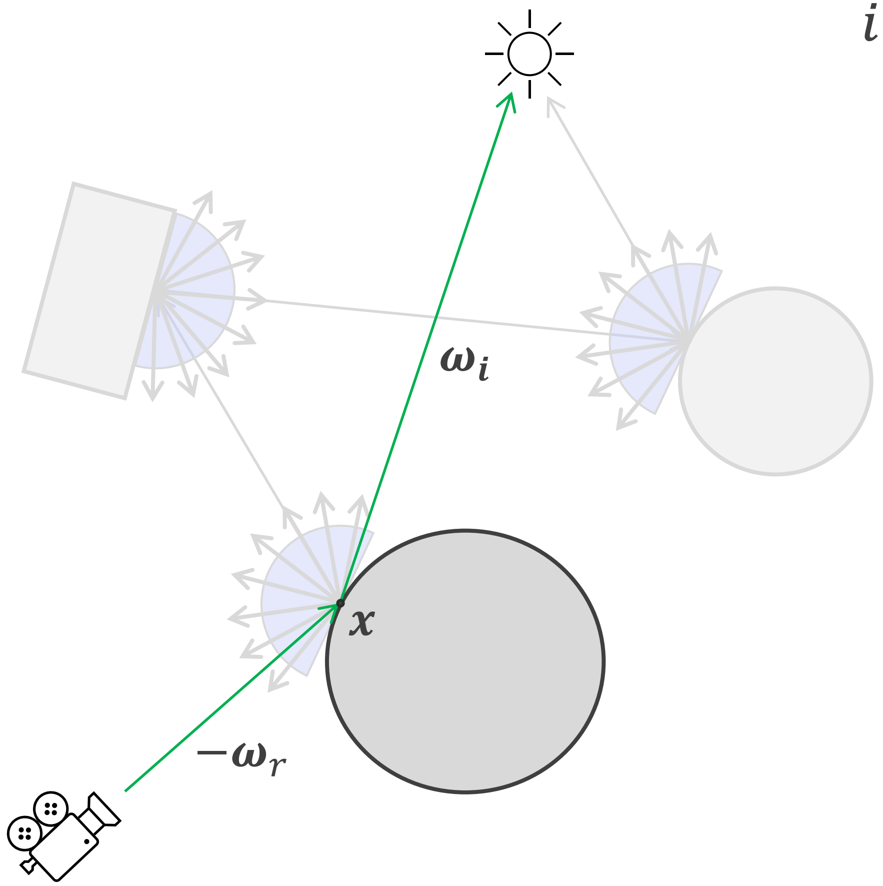
$$L_r(\boldsymbol{x}', \boldsymbol{\omega}_r') = L_e(\boldsymbol{x}', \boldsymbol{\omega}_r') + \frac{1}{n}\sum_{i=1}^{n} f_r'(\boldsymbol{x}', \boldsymbol{\omega}_i', \boldsymbol{\omega}_r') \, L_i(\boldsymbol{x}', \boldsymbol{\omega}_i') \cos\theta_i' \, d\boldsymbol{\omega}_i' \frac{1}{p(\boldsymbol{\omega}_i')}$$

$$L_r(\boldsymbol{x}'', \boldsymbol{\omega}_r'') = L_e(\boldsymbol{x}'', \boldsymbol{\omega}_r'') + \frac{1}{n}\sum_{i=1}^{n} f_r''(\boldsymbol{x}'', \boldsymbol{\omega}_i'', \boldsymbol{\omega}_r'') \, L_i(\boldsymbol{x}'', \boldsymbol{\omega}_i'') \cos\theta_i'' \, d\boldsymbol{\omega}_i'' \frac{1}{p(\boldsymbol{\omega}_i'')}$$

$$L_r(\boldsymbol{x}''', \boldsymbol{\omega}_r''') = L_e(\boldsymbol{x}''', \boldsymbol{\omega}_r''') + \frac{1}{n}\sum_{i=1}^{n} f_r'''(\boldsymbol{x}''', \boldsymbol{\omega}_i''', \boldsymbol{\omega}_r''') \, L_i(\boldsymbol{x}''', \boldsymbol{\omega}_i''') \cos\theta_i''' \, d\boldsymbol{\omega}_i''' \frac{1}{p(\boldsymbol{\omega}_i''')}$$

Direct Illumination

$$L_r(\boldsymbol{x},\boldsymbol{\omega}_r) = L_e(\boldsymbol{x},\boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x},\boldsymbol{\omega}_i,\boldsymbol{\omega}_r) L_i(\boldsymbol{x},\boldsymbol{\omega}_i)\cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$



$$L_r(\boldsymbol{x}',\boldsymbol{\omega}_r') = L_e(\boldsymbol{x}',\boldsymbol{\omega}_r') + \frac{1}{n}\sum_{i=1}^{n} f_r'(\boldsymbol{x}',\boldsymbol{\omega}_i',\boldsymbol{\omega}_r') L_i(\boldsymbol{x}',\boldsymbol{\omega}_i')\cos\theta_i'\, d\boldsymbol{\omega}_i' \frac{1}{p(\boldsymbol{\omega}_i')}$$

$$L_r(\boldsymbol{x}'',\boldsymbol{\omega}_r'') = L_e(\boldsymbol{x}'',\boldsymbol{\omega}_r'') + \frac{1}{n}\sum_{i=1}^{n} f_r''(\boldsymbol{x}'',\boldsymbol{\omega}_i'',\boldsymbol{\omega}_r'') L_i(\boldsymbol{x}'',\boldsymbol{\omega}_i'')\cos\theta_i''\, d\boldsymbol{\omega}_i'' \frac{1}{p(\boldsymbol{\omega}_i'')}$$

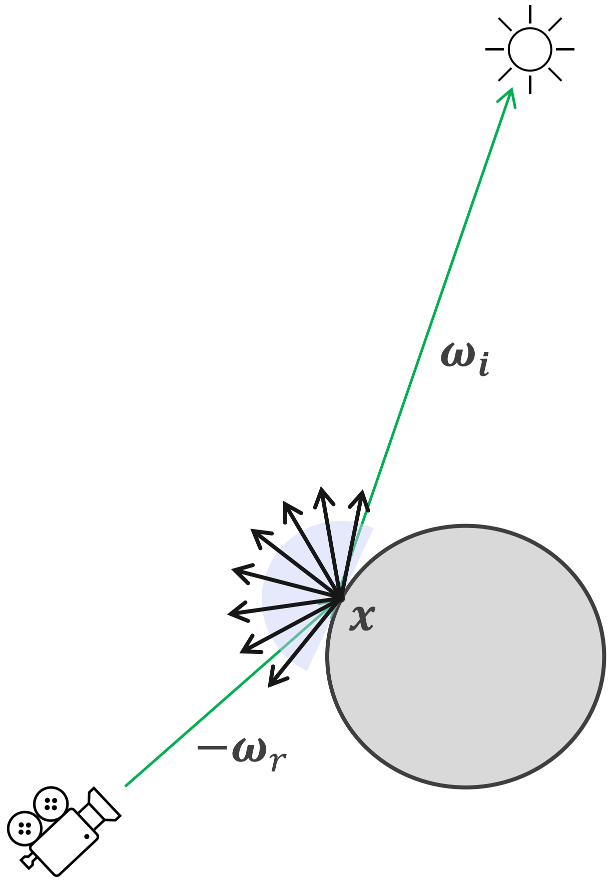
$$L_r(\boldsymbol{x}''',\boldsymbol{\omega}_r''') = L_e(\boldsymbol{x}''',\boldsymbol{\omega}_r''') + \frac{1}{n}\sum_{i=1}^{n} f_r'''(\boldsymbol{x}''',\boldsymbol{\omega}_i''',\boldsymbol{\omega}_r''') L_i(\boldsymbol{x}''',\boldsymbol{\omega}_i''')\cos\theta_i'''\, d\boldsymbol{\omega}_i''' \frac{1}{p(\boldsymbol{\omega}_i''')}$$

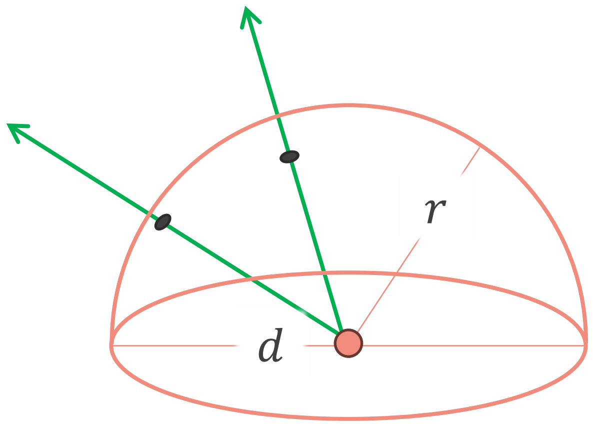
Direct Illumination

evaluates to 0 for non-emissive objects

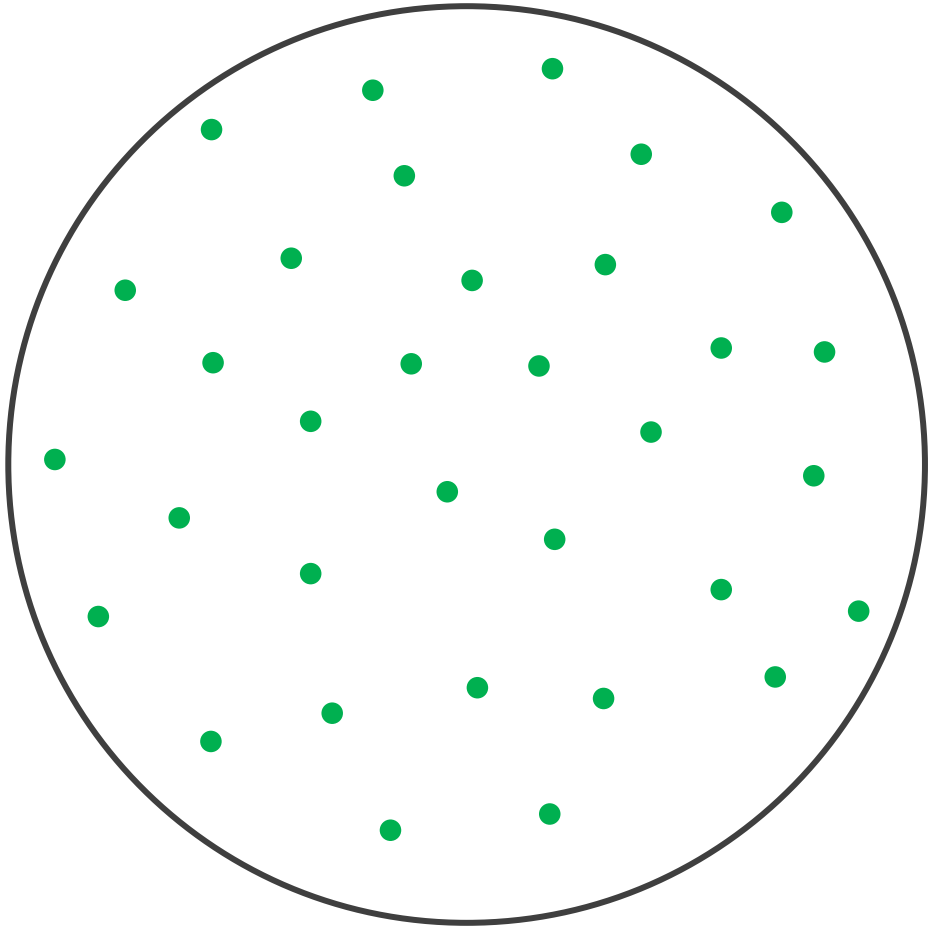
$\boldsymbol{\omega}'_r = -\boldsymbol{\omega}_i$

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_e(\boldsymbol{x}', \boldsymbol{\omega}'_r) \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$



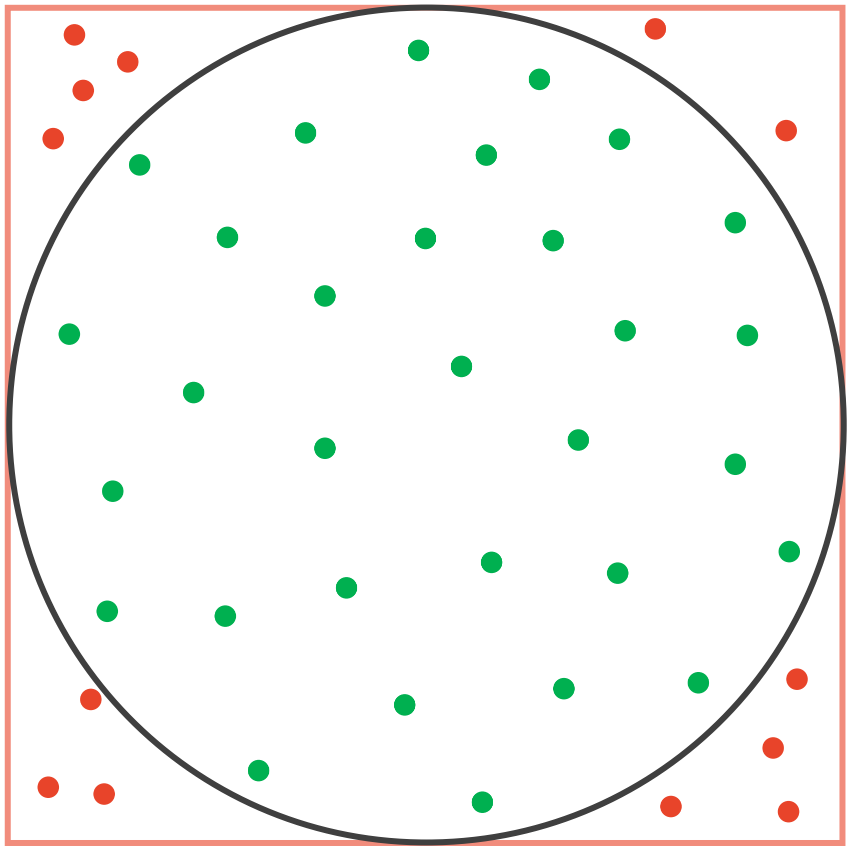


Uniformly Sample Unit Circle



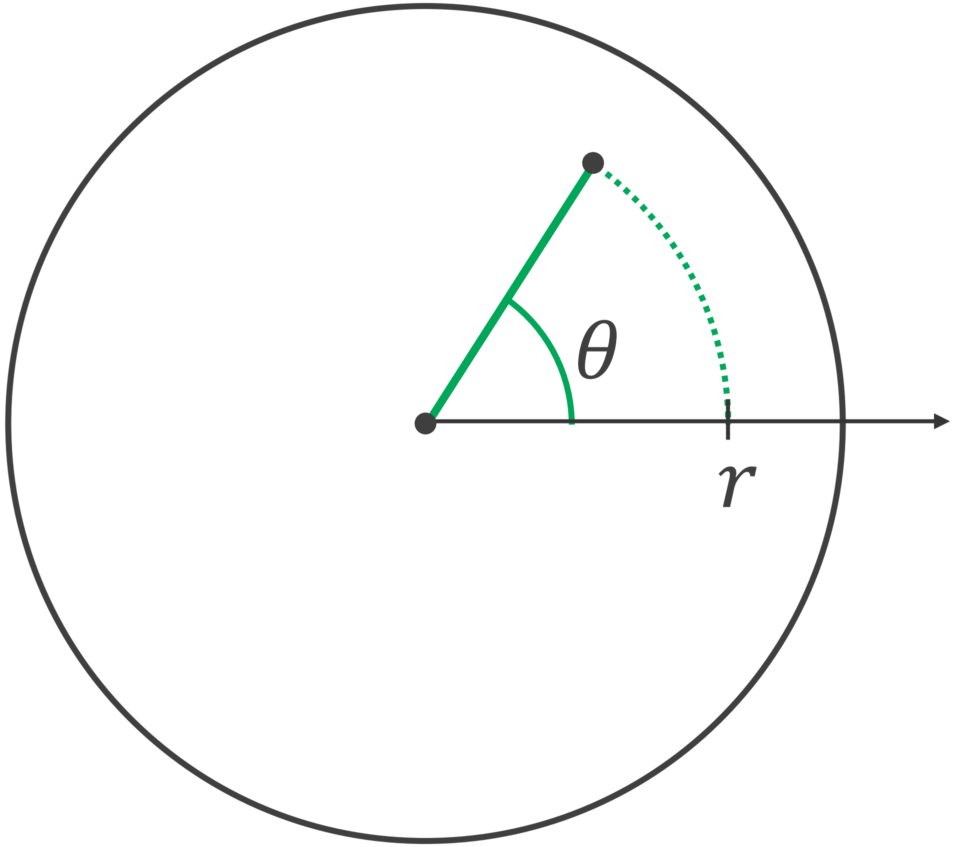
Rejection Technique

- Randomly generate (x, y)
  - $x \in [0,1]$
  - $y \in [0,1]$
  - keep when $x^2 + y^2 \leq 1$



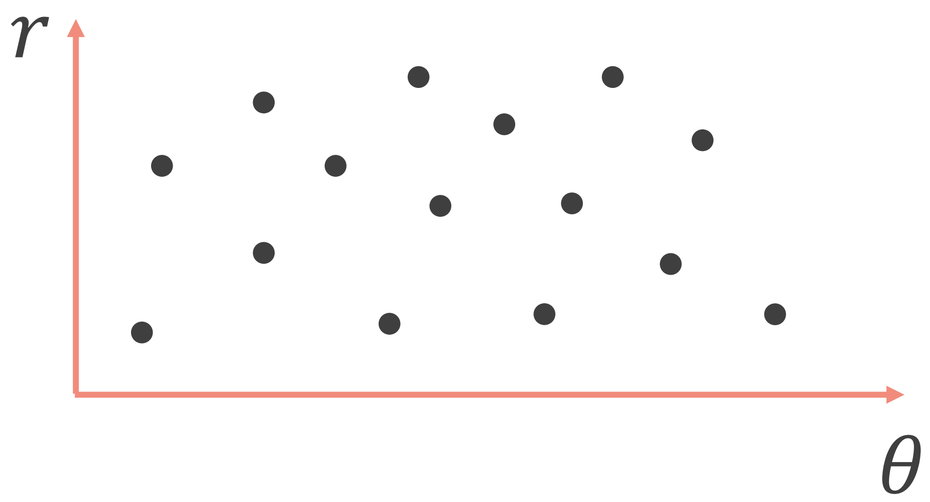
Polar Coordinates

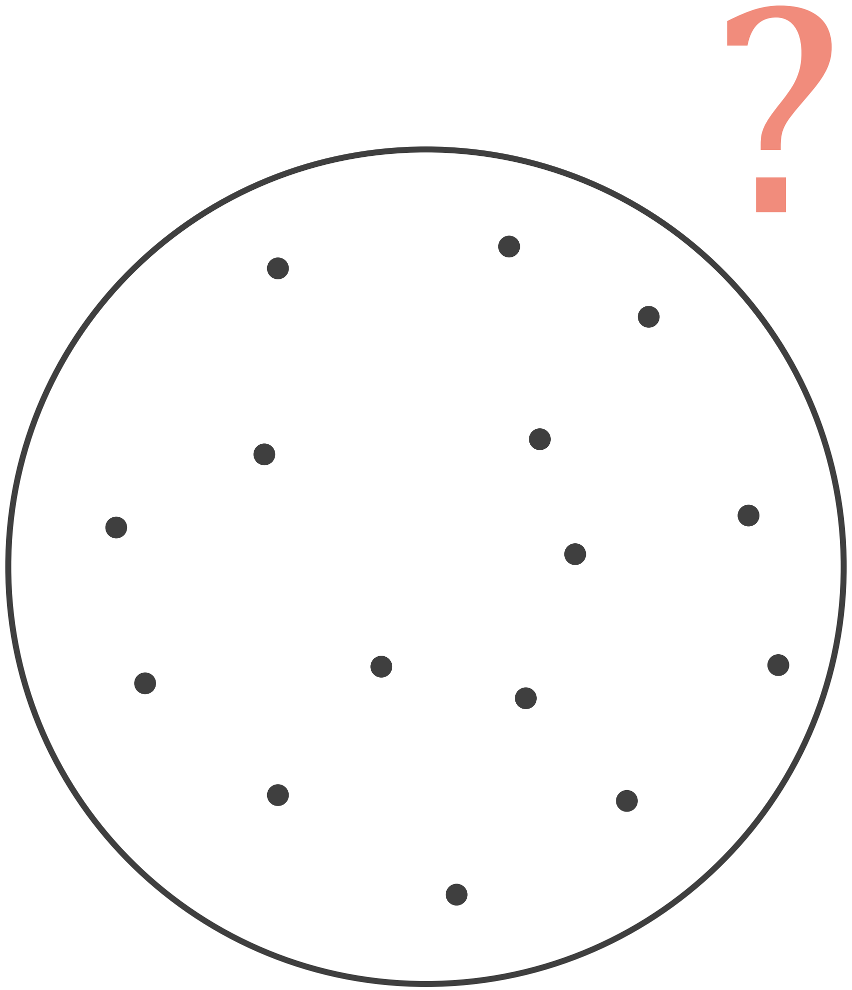
- θ uniform random angle between 0 and 2π
- r uniform random angle between 0 and 1



Polar Coordinates

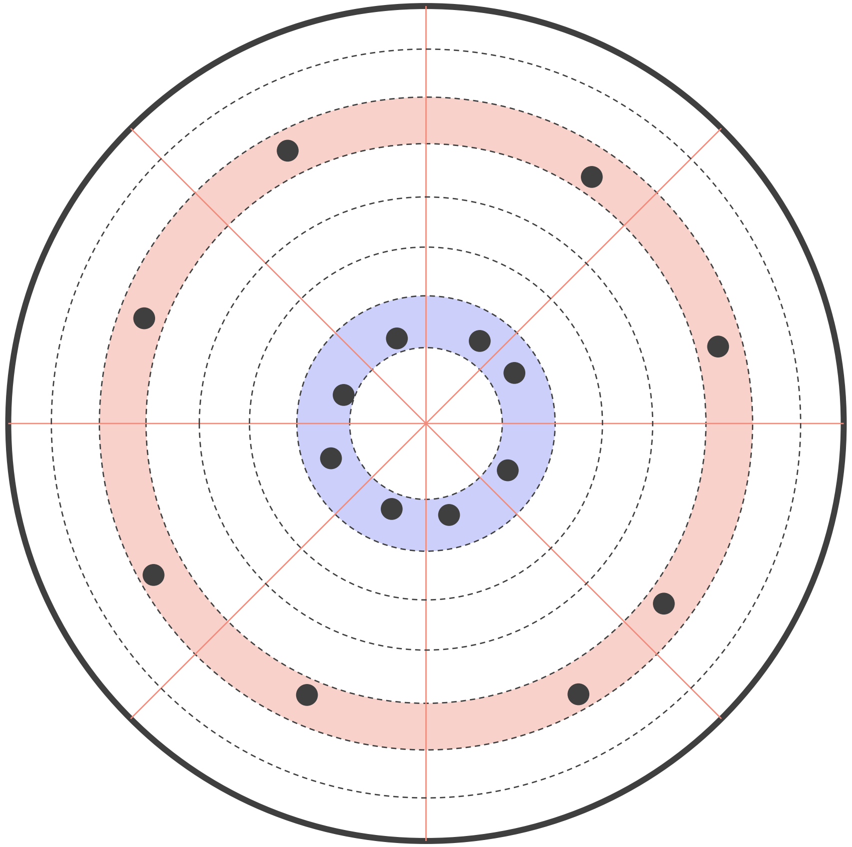
- θ uniform random angle between 0 and 2π
- r uniform random angle between 0 and 1





Sampling is NOT Uniform in Area!

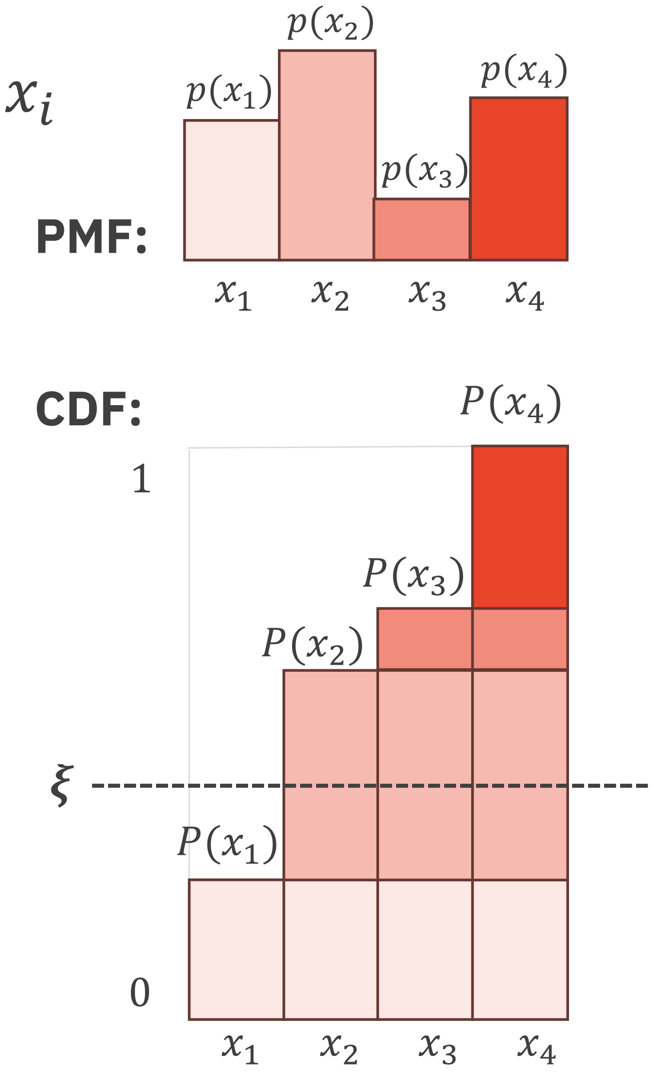
- Samples closer to the center are more likely to be chosen



Cumulative Distribution Function (CDF)

- Probability that random variable X will take a value $\leq x_i$
- **For PMF:** $\boldsymbol{P(x_i) = \sum_{j=1}^{i} p(x_j)}$
- For PDF: $P(x) = \int_0^x p(x)\,dx$
- Properties
  - $0 < P(x) < 1$
  - $P(n) = 1$
- Sampling probability distributions
  - Select x_i if $P(x_{i-1}) < \xi < P(x_i)$

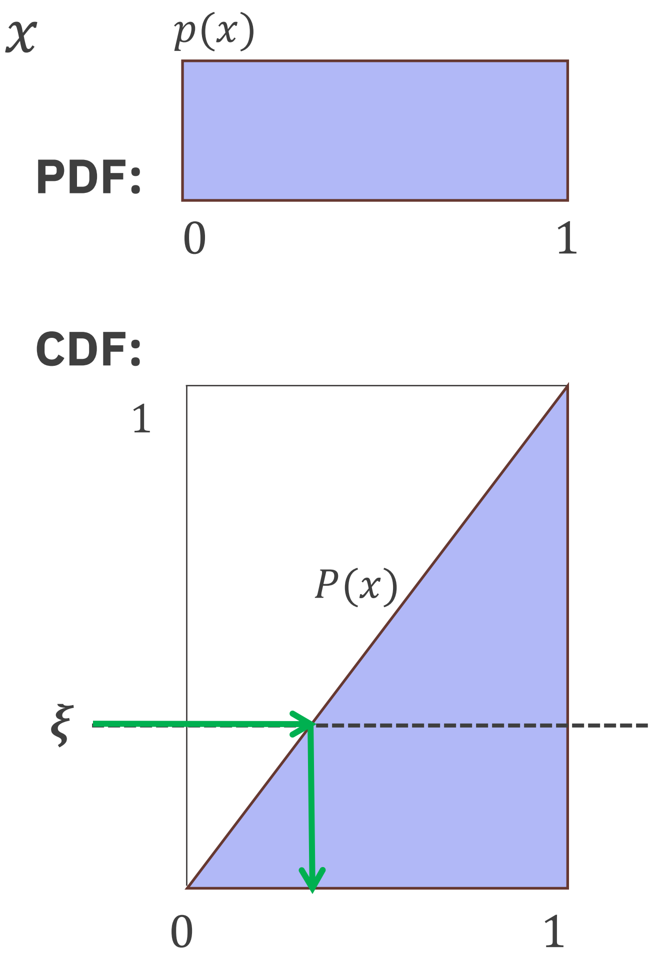
canonical uniform random variable ∊ [0,1)



Cumulative Distribution Function (CDF)

- Probability that random variable X will take a value $\leq x$
- For PMF: $P(x_i) = \sum_{j=1}^{i} p(x_j)$
- **For PDF:** $\boldsymbol{P(x) = \int_0^x p(x)\, dx}$
- Properties
  - $0 < P(x) < 1$
  - $P(n) = 1$
- Sampling probability distributions
  - **Select** $\boldsymbol{x = P^{-1}(\xi)}$

canonical uniform random variable ∈ [0,1)



Better Sampling Using Inversion Method

https://www.pbr-book.org/3ed-2018/Monte_Carlo_Integration/Transforming_between_Distributions
https://www.pbr-book.org/3ed-2018/Monte_Carlo_Integration/2D_Sampling_with_Multidimensional_Transformations

- Area of the **unit** disk: π

- Since we want uniform distribution over the disk and $\int_D p(x, y) = 1$

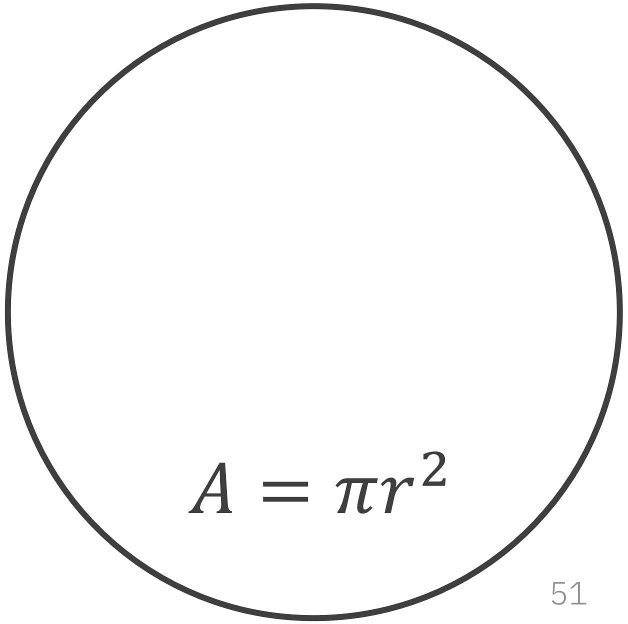
$$p(x,y) = \frac{1}{\pi} \longrightarrow p(r,\theta) = \frac{r}{\pi}$$

$$p(x,y) = \frac{p(r,\theta)}{r} \rightarrow p(r,\theta) = r\, p(x,y)$$

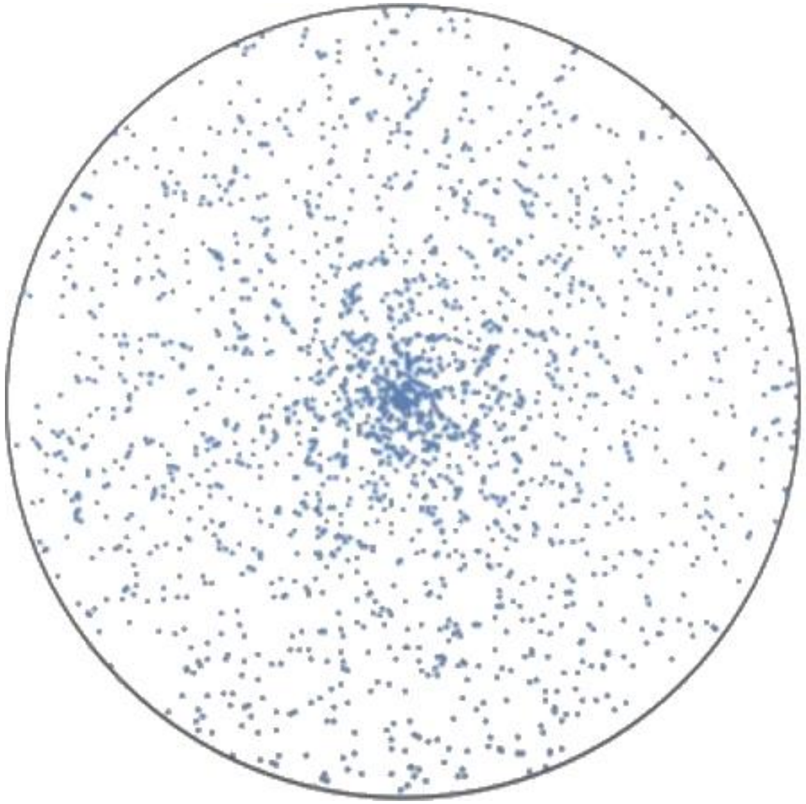
- r, θ are independent $\rightarrow p(r,\theta) = p(r)p(\theta)$

$$p(\theta) = \frac{1}{2\pi} \longrightarrow P(\theta) = \frac{\theta}{2\pi} \longrightarrow P^{-1}(\xi_1) = 2\pi\xi_1$$

$$p(r) = 2r \longrightarrow P(r) = r^2 \longrightarrow P^{-1}(\xi_2) = \sqrt{\xi_2}$$

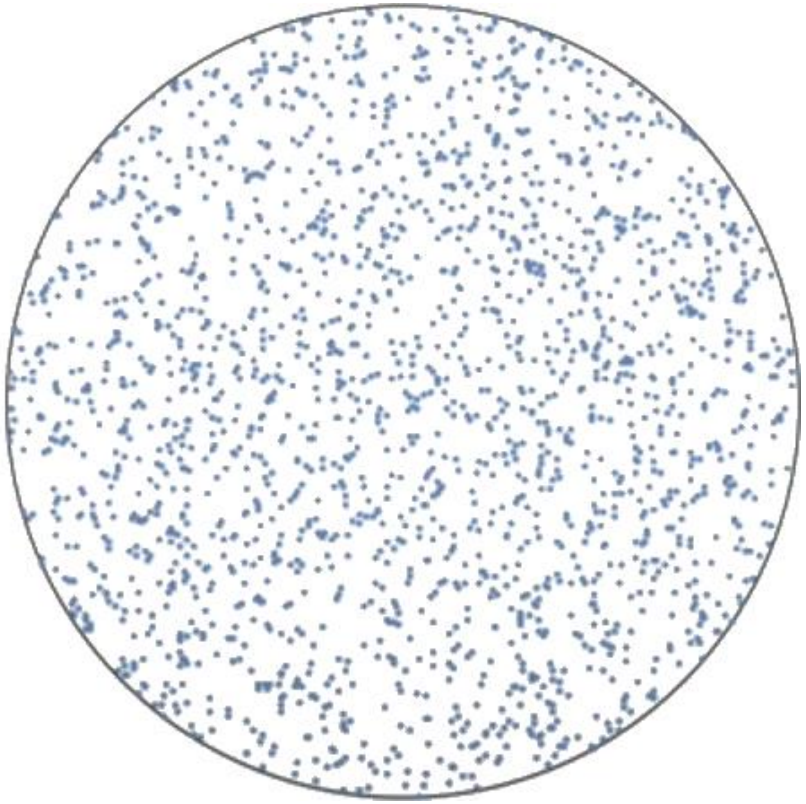


Better Sampling Using Inversion Method



$$\theta = 2\pi\xi_1$$

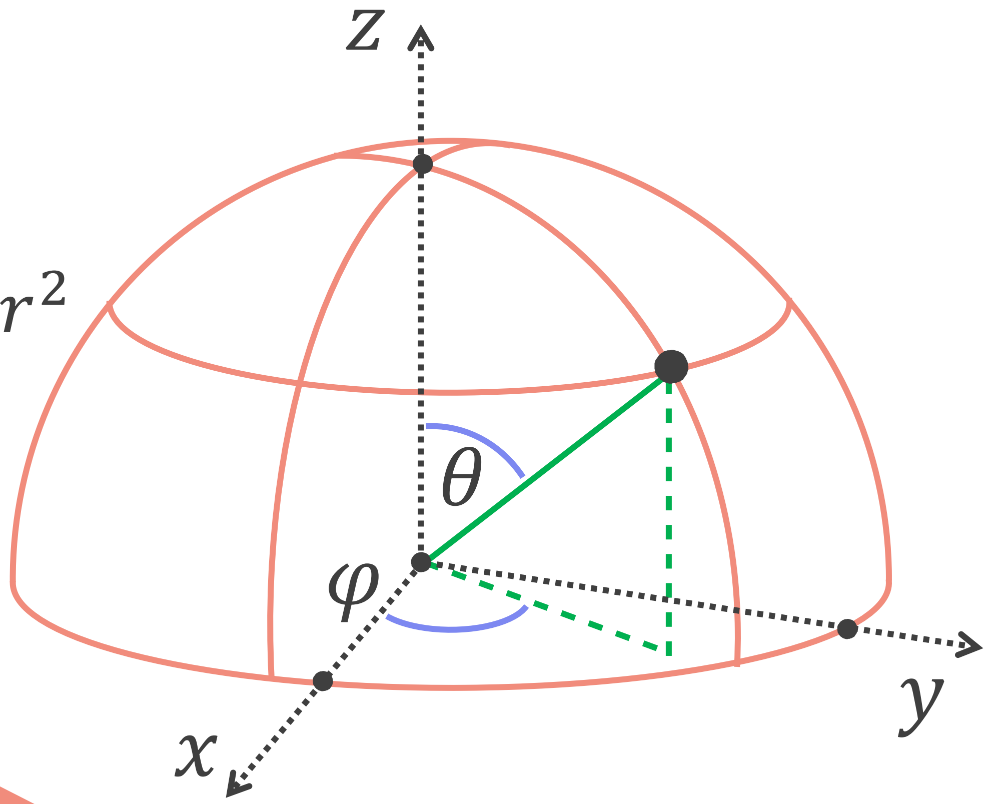
$$r = \xi_2$$



$$\theta = 2\pi\xi_1$$

$$r = \sqrt{\xi_2}$$

Uniform Hemisphere Sampling



$A = 2\pi r^2$

$$x = r \sin\theta \cos\varphi$$
$$y = r \sin\theta \sin\varphi$$
$$z = r \cos\theta$$

$p(r,\theta,\varphi) = r^2 sin\theta\ p(x,y,z)$

$$p(\omega) = \frac{1}{2\pi} \rightarrow p(\theta,\varphi) = \frac{\sin\theta}{2\pi}$$

note that we ignore r

$$p(\theta) = \sin\theta \rightarrow P(\theta) = 1 - \cos\theta \rightarrow P^{-1}(\xi_1) = \cos^{-1}(1-\xi_1) \rightarrow \cos^{-1}(\xi_1)$$

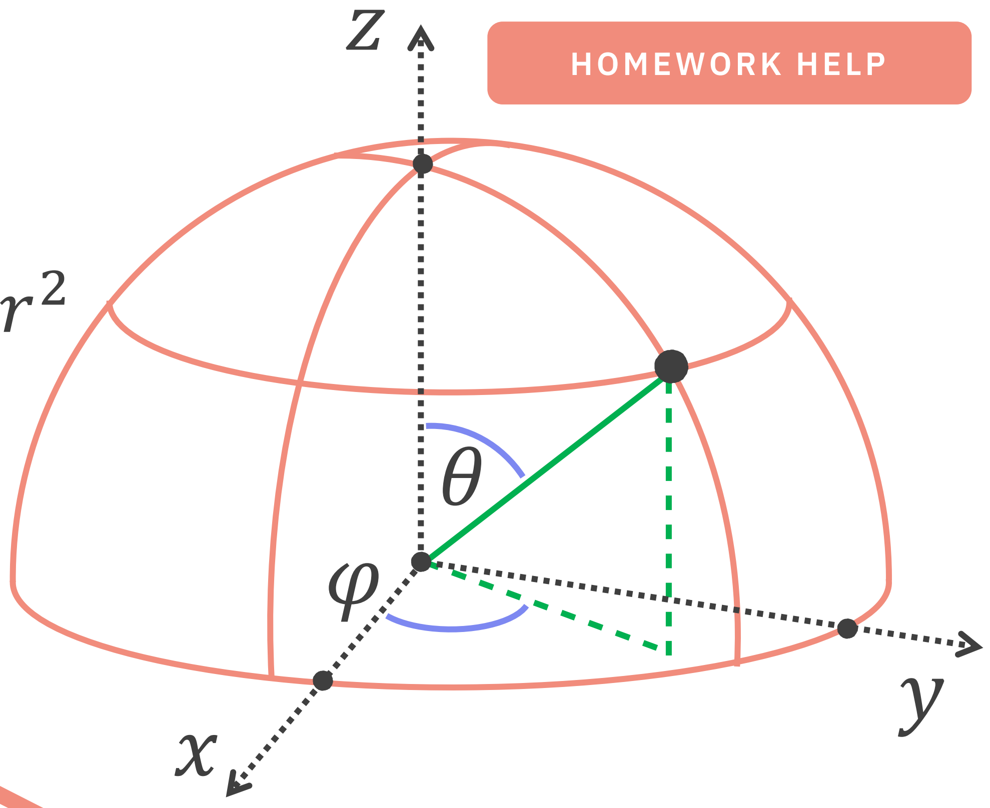
$$p(\varphi) = \frac{1}{2\pi} \rightarrow P(\varphi) = \frac{\varphi}{2\pi} \rightarrow P^{-1}(\xi_2) = 2\pi\xi_2$$

Uniform Hemisphere Sampling

HOMEWORK HELP

$$x = r\sin\theta\cos\varphi = \sqrt{1-\xi_1^2}\cos(2\pi\xi_2)$$
$$y = r\sin\theta\sin\varphi = \sqrt{1-\xi_1^2}\sin(2\pi\xi_2)$$
$$z = r\cos\theta = \xi_1$$

$A = 2\pi r^2$



$p(r,\theta,\varphi) = r^2 sin\theta\, p(x,y,z)$

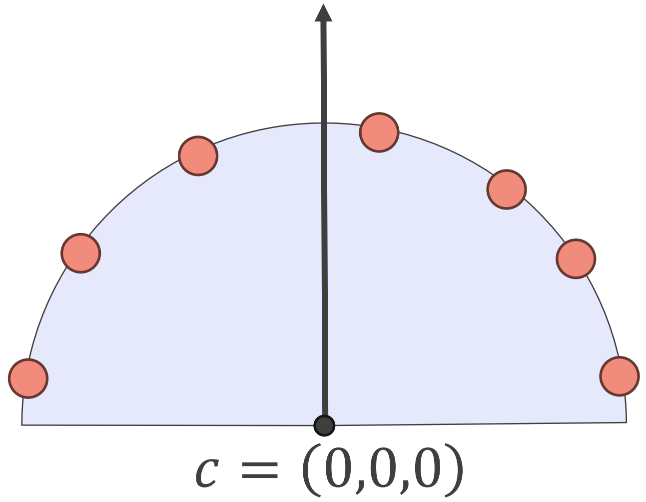
$$p(\omega) = \frac{1}{2\pi} \rightarrow p(\theta,\varphi) = \frac{\sin\theta}{2\pi}$$

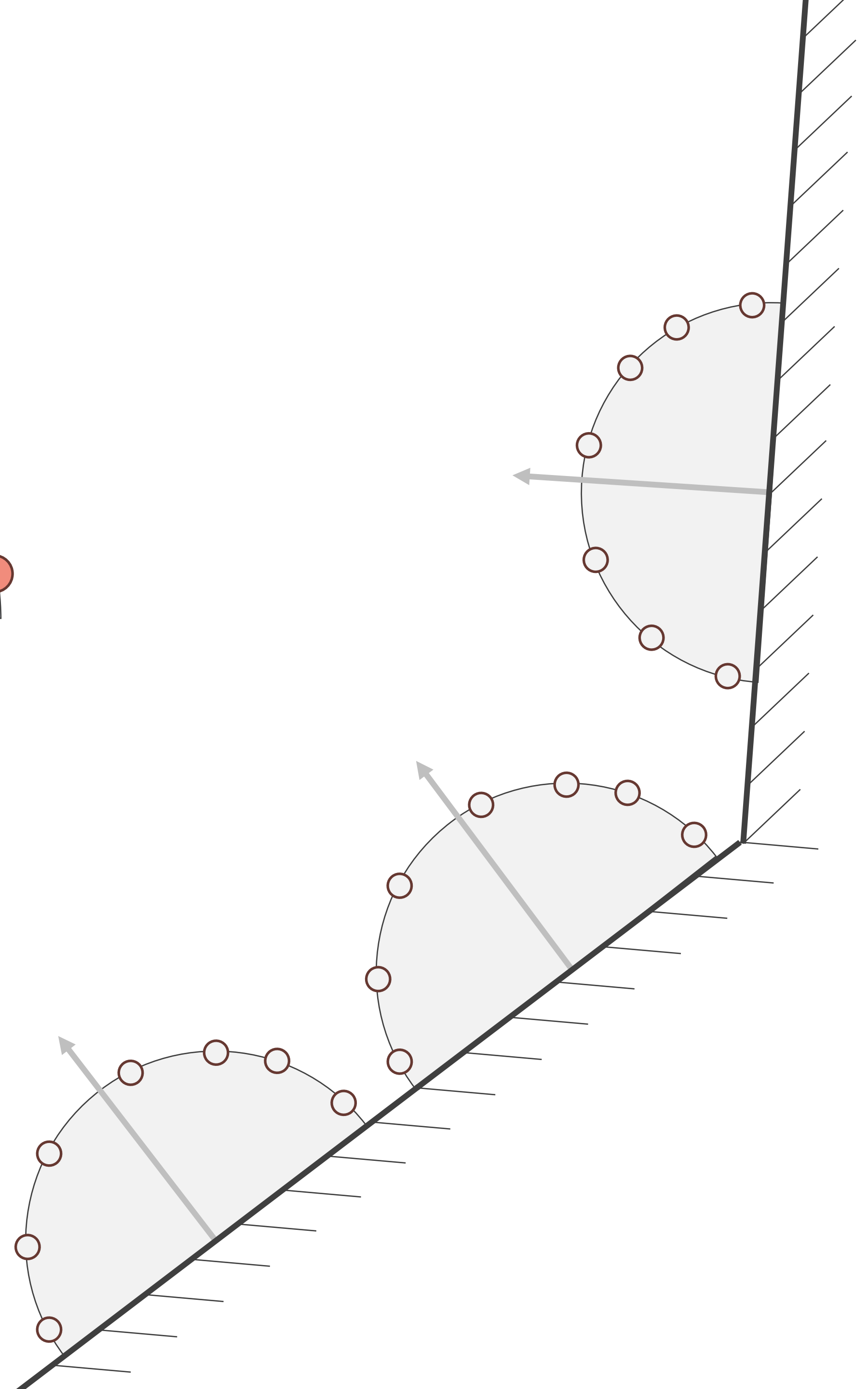
note that we ignore r

$$p(\theta) = \sin\theta \rightarrow P(\theta) = 1-\cos\theta \rightarrow P^{-1}(\xi_1) = \cos^{-1}(1-\xi_1) \rightarrow \cos^{-1}(\xi_1)$$

$$p(\varphi) = \frac{1}{2\pi} \rightarrow P(\varphi) = \frac{\varphi}{2\pi} \rightarrow P^{-1}(\xi_2) = 2\pi\xi_2$$

Transforming Samples





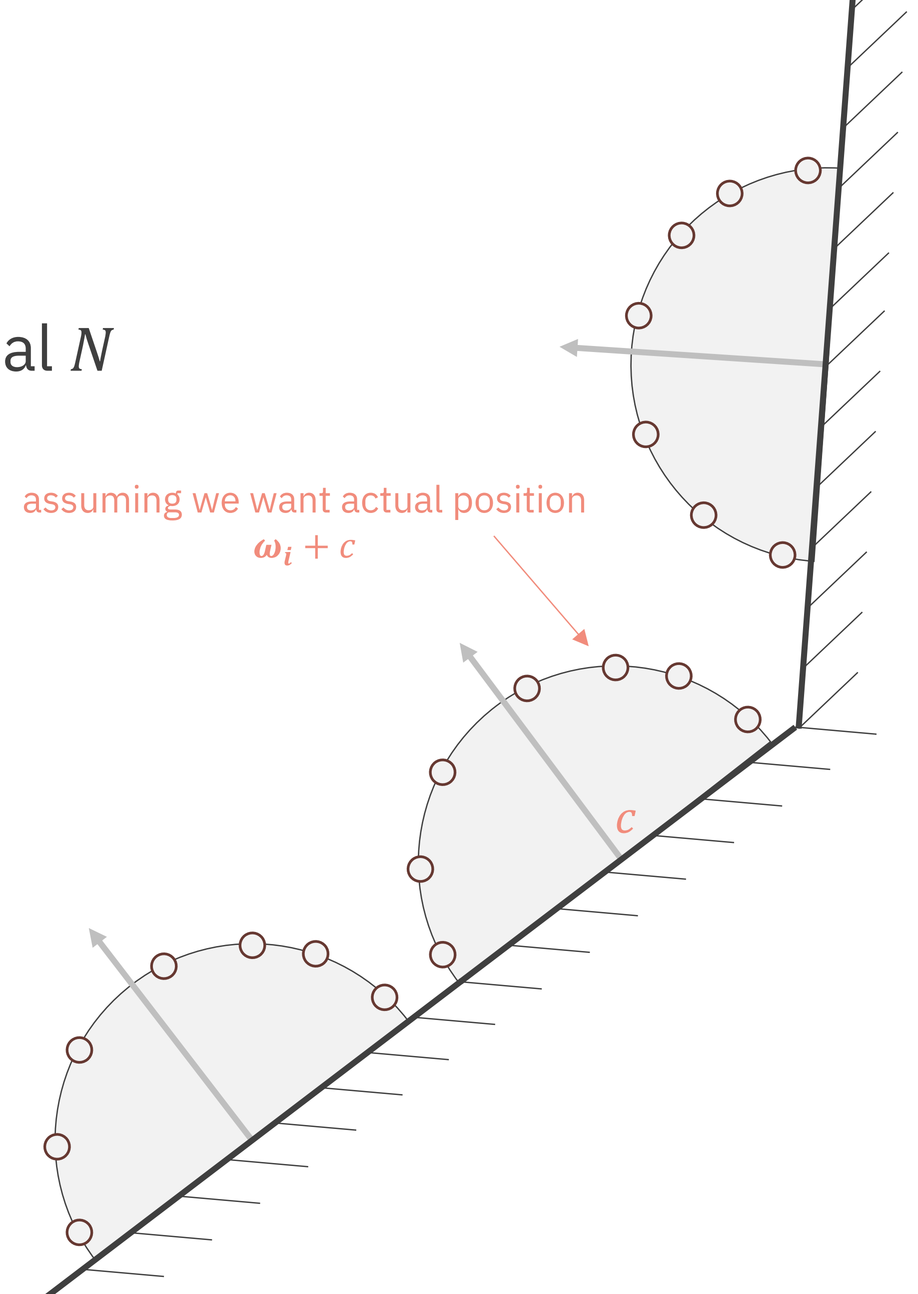
HOMEWORK HELP

Transforming Samples

- We do not care about hemisphere rotation
- So for a local sample (x, y, z) and surface normal N
  - $\boldsymbol{U} = \begin{cases} (0,0,1) & when\ |\boldsymbol{N}.z| < 0.999 \\ (0,1,0) & otherwise \end{cases}$
  - $\boldsymbol{T} = \boldsymbol{U} \times \boldsymbol{N}$
  - $\boldsymbol{B} = \boldsymbol{N} \times \boldsymbol{T}$
  - $\boldsymbol{\omega_i} = \boldsymbol{T}x + \boldsymbol{B}y + \boldsymbol{N}z$

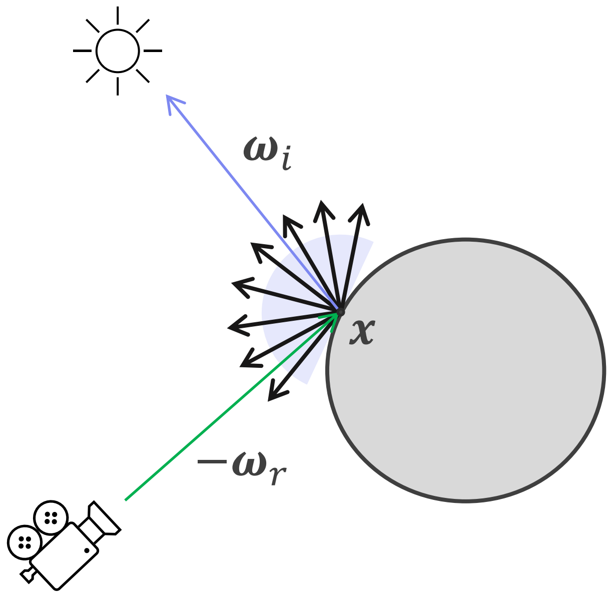
cross products, order is important, do not forget to normalize

sample local coordinates



Direct Illumination

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_e(\boldsymbol{x}', \boldsymbol{\omega}'_r) \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$

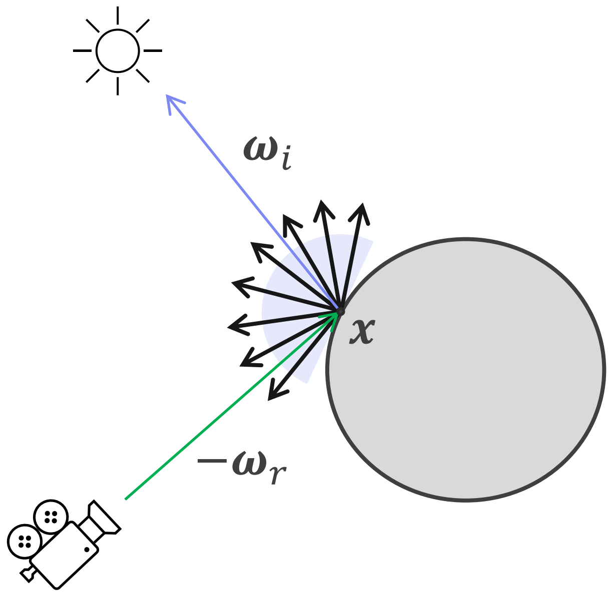


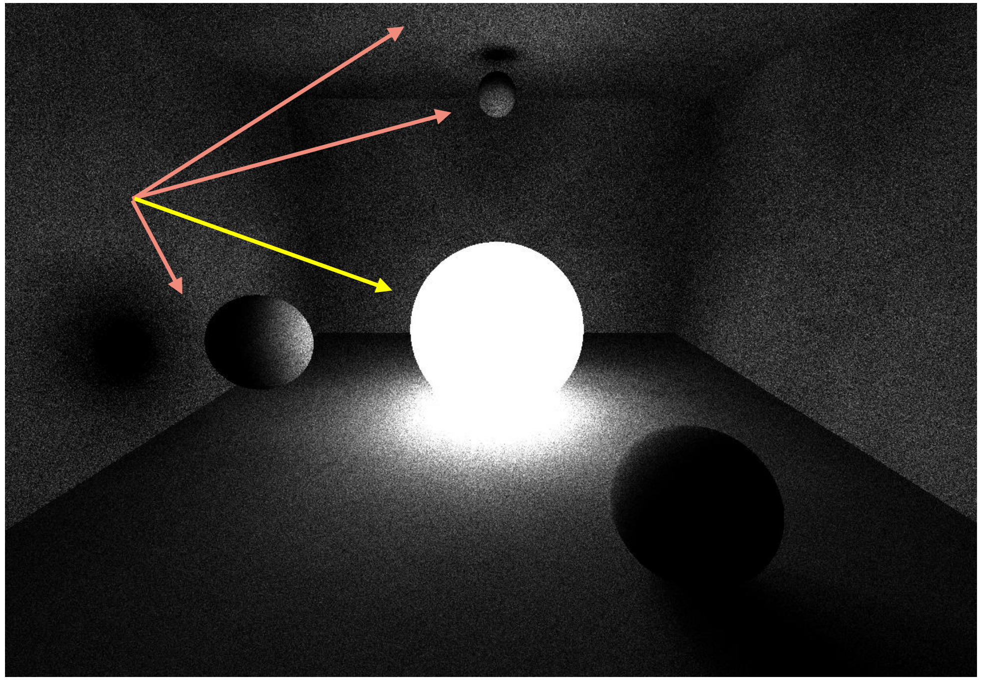

```
Trace(ray) {
    radiance ← (0,0,0)
    hit ← ClosestHit(ray)
    if(hit == miss) return radiance

    Le = hit.material.emission

    for(i=0; i < samples; i++) {
        sample ← GetNextSample(hit, ray)
        next_ray ← Ray(hit.intersection + epsilon * hit.normal, sample.direction)
        next_hit ← ClosestHit(next_ray);
        if(next_hit == miss || sample.pdf == 0) continue //i.e., skip this sample

        brdf ← ComputeBRDF(hit, next_ray.direction, -ray.direction)
        emission ← next_hit.material.emission
        radiance += brdf * emission * dot(hit.normal, next_ray.direction) / sample.pdf
    }
    return Le + radiance / samples
}
```

Direct Illumination

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_e(\boldsymbol{x}', \boldsymbol{\omega}_r') \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$

$$f_r = 1.0, n = 100, p(\boldsymbol{\omega}_i) = \frac{1}{2\pi}$$

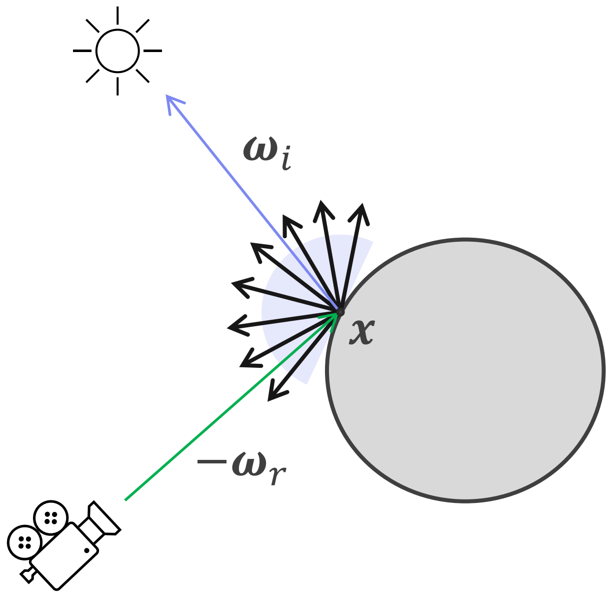


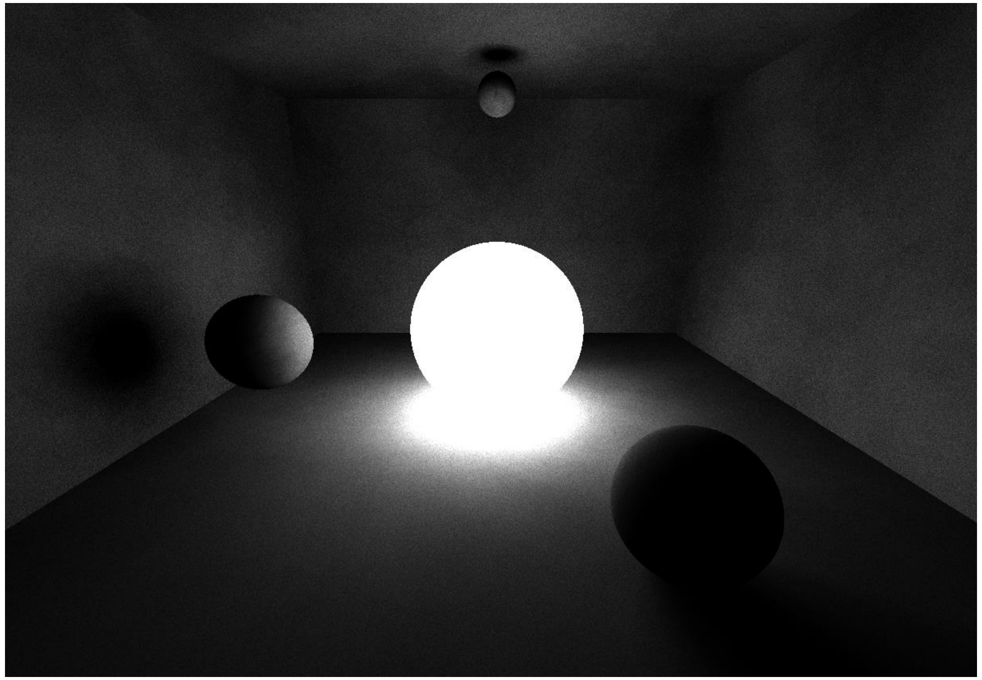


Direct Illumination

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_e(\boldsymbol{x}', \boldsymbol{\omega}_r') \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$

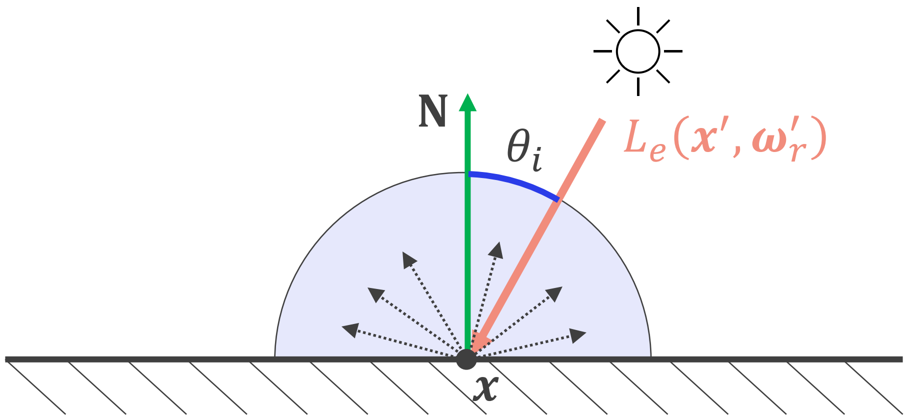
$$f_r = 1.0, n = 1000, p(\boldsymbol{\omega}_i) = \frac{1}{2\pi}$$

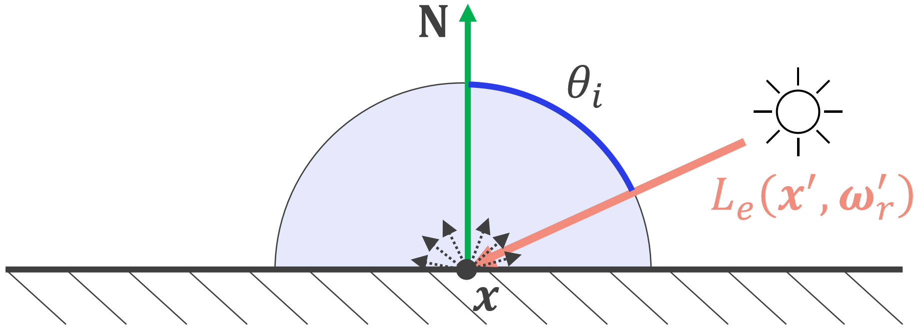




Lambert's Cosine Law

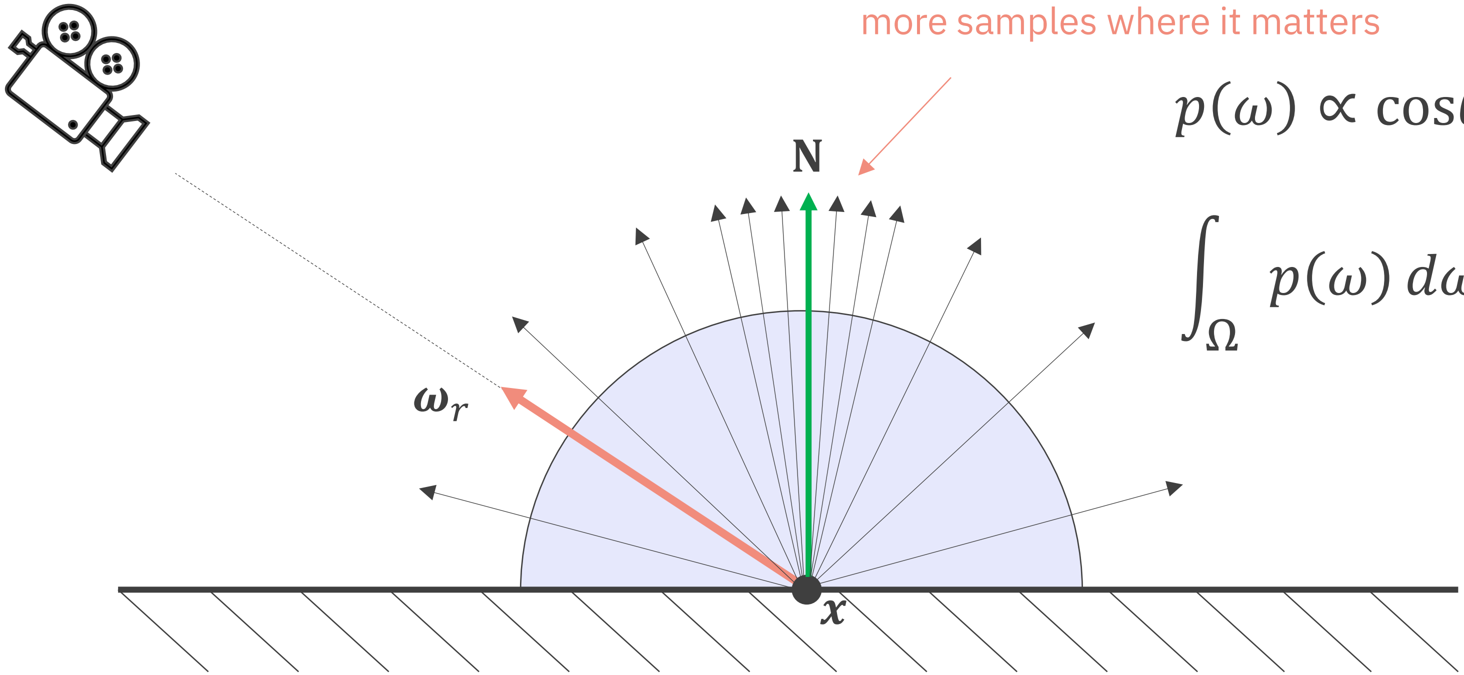
$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_e(\boldsymbol{x}', \boldsymbol{\omega}_r') \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$





Cosine-Weighted Sampling

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_e(\boldsymbol{x}', \boldsymbol{\omega}_r') \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$

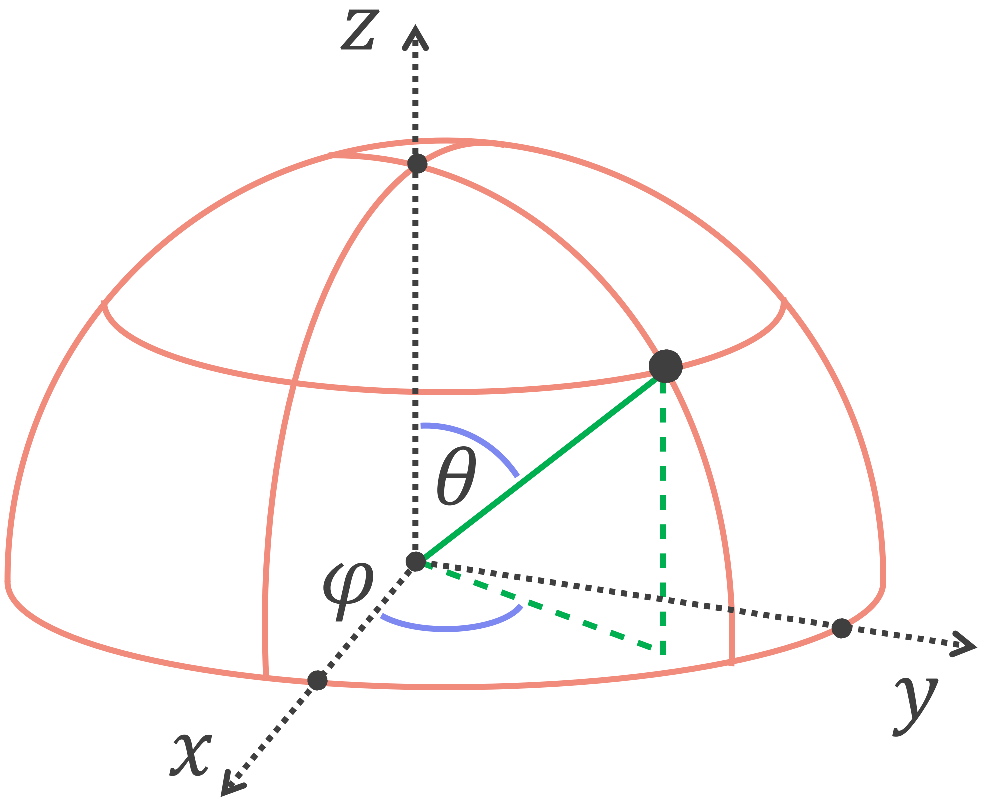


$$p(\omega) \propto \cos\theta$$

$$\int_{\Omega} p(\omega)\, d\omega = 1 \rightarrow p(\omega) = \frac{\cos\theta}{\pi}$$

Cosine-Weighted Sampling

$$x = r\sin\theta\cos\varphi$$
$$y = r\sin\theta\sin\varphi$$
$$z = r\cos\theta$$



$p(r,\theta,\varphi) = r^2 sin\theta\, p(x,y,z)$

$$p(\omega) = \frac{\cos\theta}{\pi} \rightarrow p(\theta,\varphi) = \frac{\cos\theta\sin\theta}{\pi}$$

note that we ignore r

$$p(\theta) = 2\cos\theta\sin\theta \rightarrow P(\theta) = 1 - \cos^2\theta \rightarrow P^{-1}(\xi_1) = \cos^{-1}\sqrt{1-\xi_1}$$

$$p(\varphi) = \frac{1}{2\pi} \rightarrow P(\varphi) = \frac{\varphi}{2\pi} \rightarrow P^{-1}(\xi_2) = 2\pi\xi_2$$

Cosine-Weighted Sampling

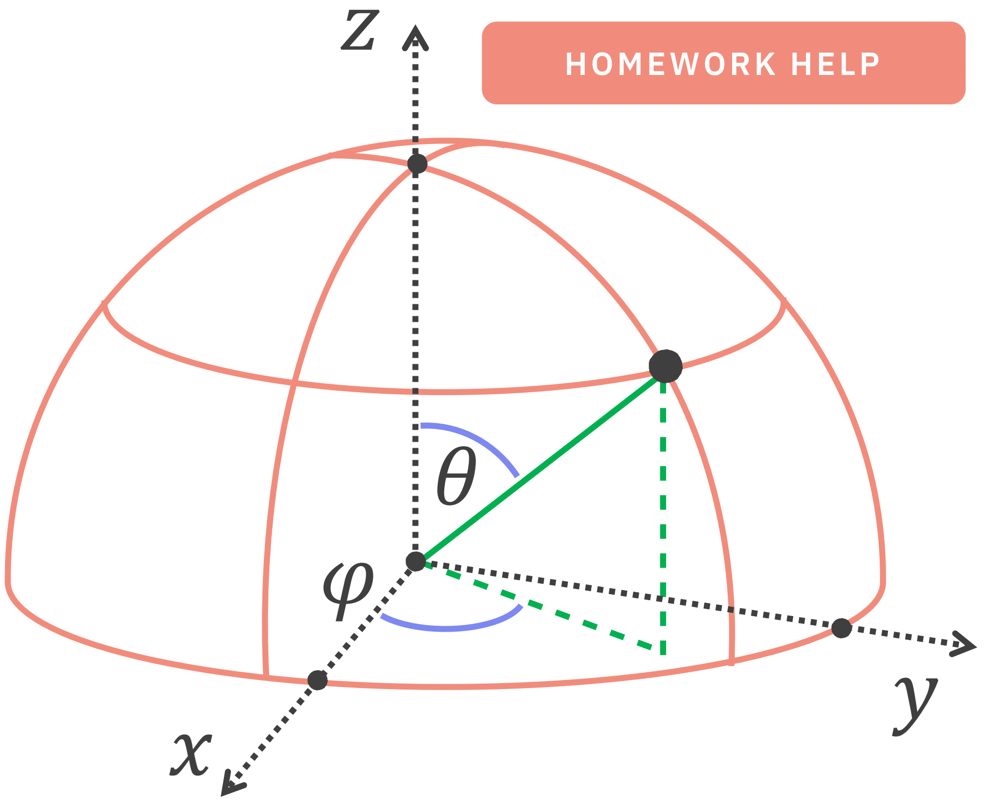
HOMEWORK HELP

$$x = r \sin\theta \cos\varphi = \sqrt{\xi_1}\cos(2\pi\xi_2)$$

$$y = r \sin\theta \sin\varphi = \sqrt{\xi_1}\sin(2\pi\xi_2)$$

$$z = r\cos\theta = \sqrt{1-\xi_1} = \sqrt{1-x^2-y^2}$$

Malley's method



$p(r,\theta,\varphi) = r^2 sin\theta \; p(x,y,z)$

$$p(\omega) = \frac{\cos\theta}{\pi} \;\rightarrow\; p(\theta,\varphi) = \frac{\cos\theta \sin\theta}{\pi}$$

note that we ignore r

$$p(\theta) = 2\cos\theta\sin\theta \;\rightarrow\; P(\theta) = 1-\cos^2\theta \;\rightarrow\; P^{-1}(\xi_1) = \cos^{-1}\sqrt{1-\xi_1}$$

$$p(\varphi) = \frac{1}{2\pi} \;\rightarrow\; P(\varphi) = \frac{\varphi}{2\pi} \;\rightarrow\; P^{-1}(\xi_2) = 2\pi\xi_2$$

Cosine-Weighted Sampling

$$p(\boldsymbol{\omega}_i) = \frac{\cos\theta_i}{\pi}$$

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_e(\boldsymbol{x}', \boldsymbol{\omega}_r') \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$

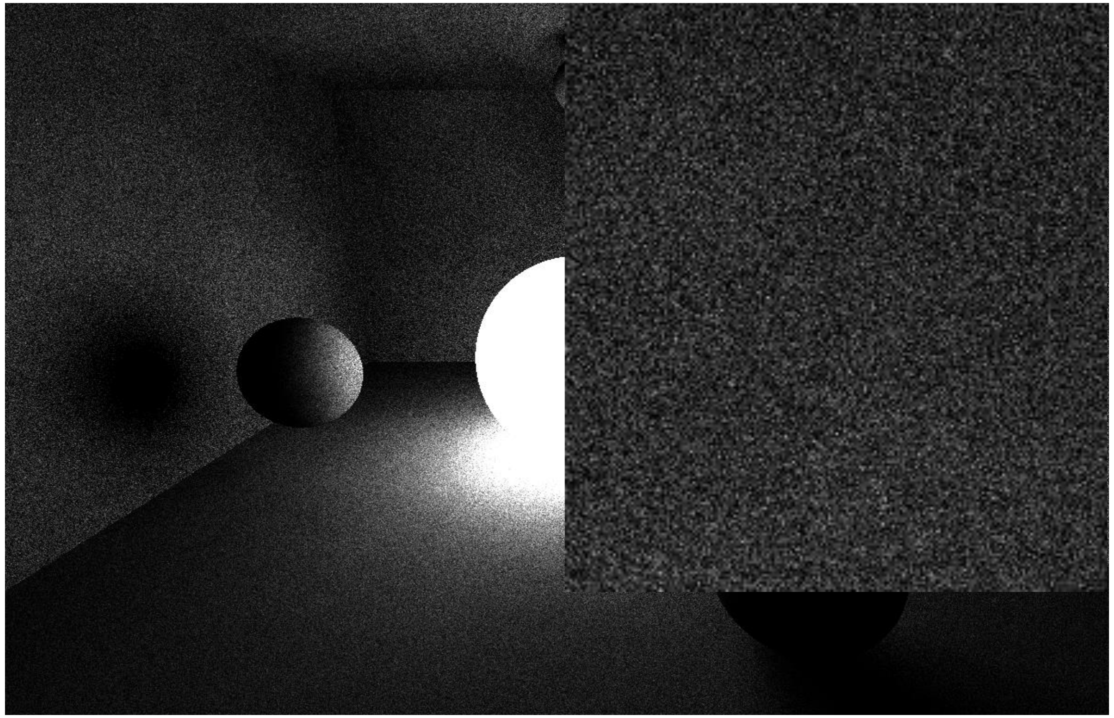
$$= L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_e(\boldsymbol{x}', \boldsymbol{\omega}_r') \cos\theta_i \frac{\pi}{\cos\theta_i}$$

$$= L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_e(\boldsymbol{x}', \boldsymbol{\omega}_r') \pi$$

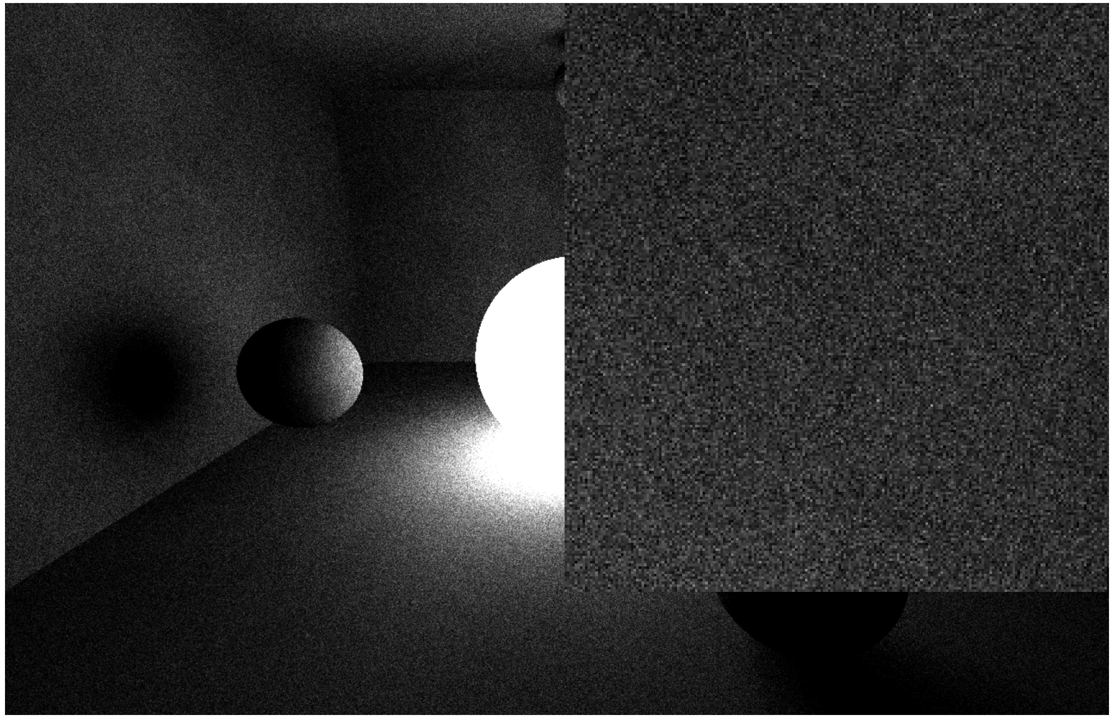
Observation: We no longer care about the light direction, because we have more samples around the normal, thus the contribution will be stronger from there

Comparison

$f_r = 1.0, n = 100, p(\boldsymbol{\omega}_i) = \frac{1}{2\pi}$



$f_r = 1.0, n = 100, p(\boldsymbol{\omega}_i) = \frac{\cos\theta}{\pi}$



Energy Non-Conserving BRDF

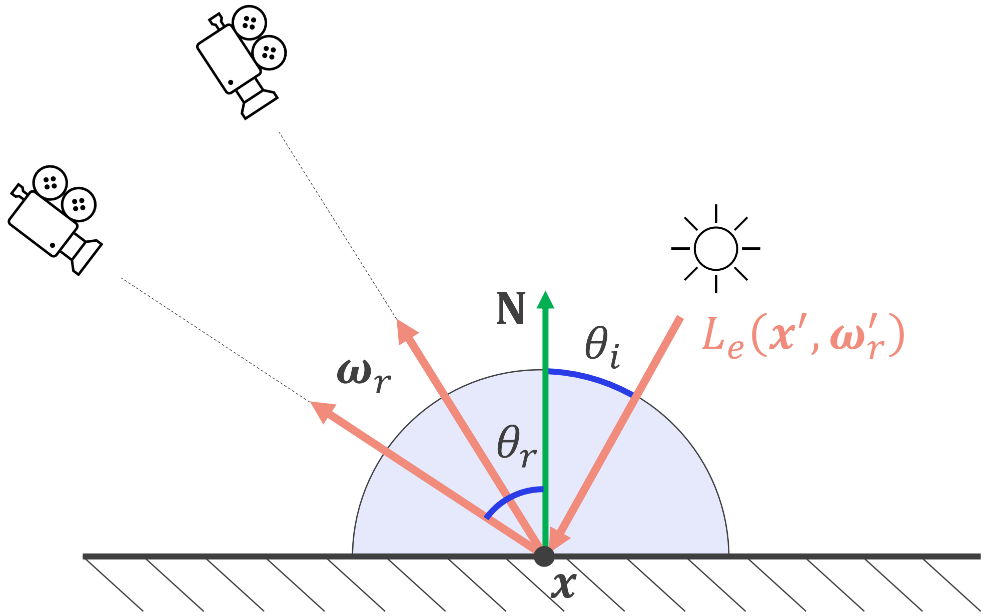
HOMEWORK HELP

$f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) = 1$

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} 1 \, L_e(\boldsymbol{x}', \boldsymbol{\omega}_r') \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$

- Sum of energy reflected in all directions:

$$\int_{\Omega} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) \cos\theta_r \, d\boldsymbol{\omega}_r > 1$$



Lambertian BRDF

HOMEWORK HELP

$$f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) = \frac{\rho}{\pi}$$

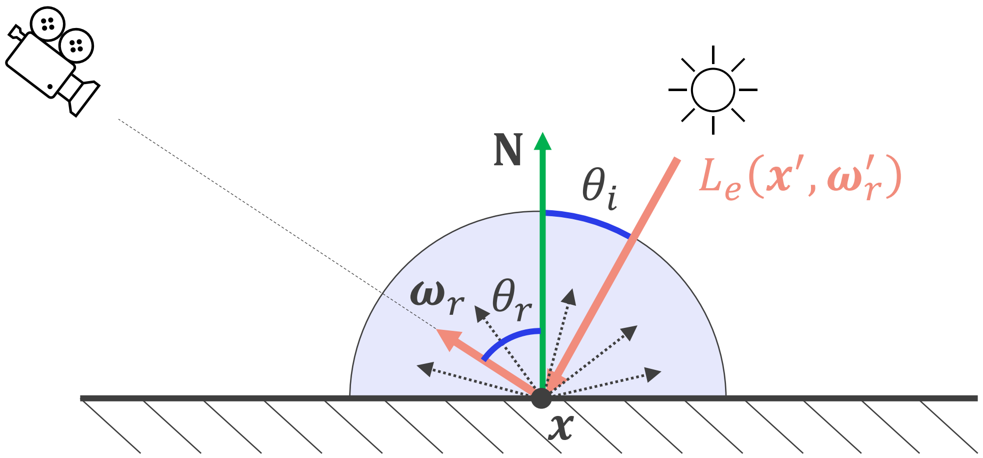
$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} \frac{\rho}{\pi} L_e(\boldsymbol{x}', \boldsymbol{\omega}_r') \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$

- Sum of energy reflected in all directions:

$$\int_\Omega f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) \cos\theta_r \, d\boldsymbol{\omega}_r \leq 1$$

$$f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) \leq \frac{1}{\int_\Omega \cos\theta_r \, d\boldsymbol{\omega}_r}$$

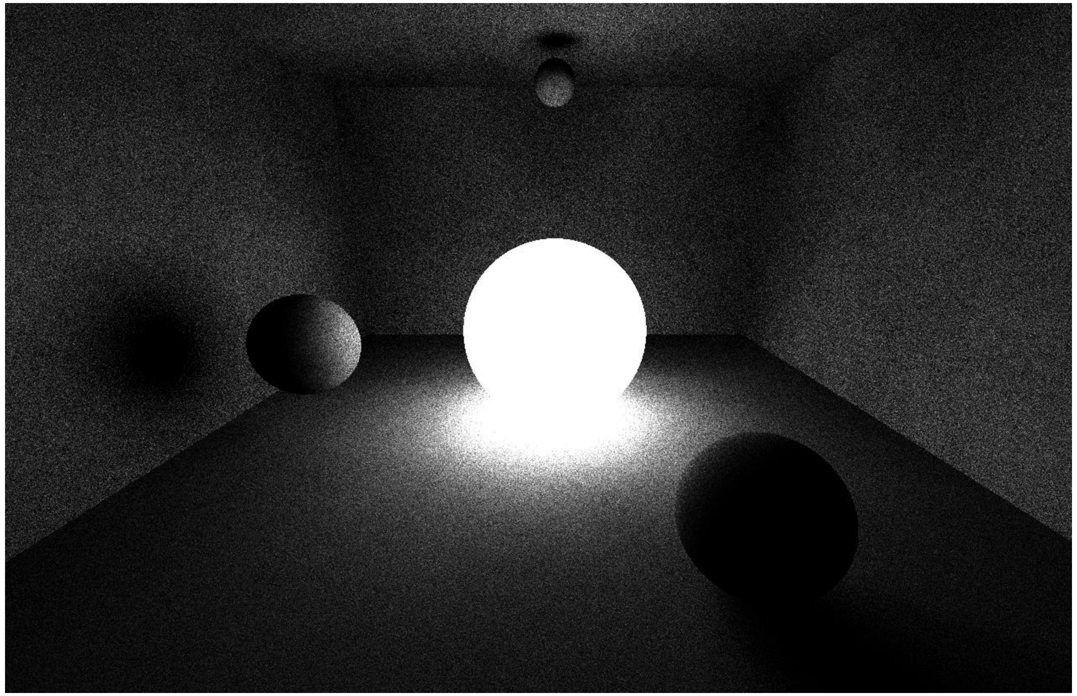
$$f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) \leq \frac{1}{\pi}$$



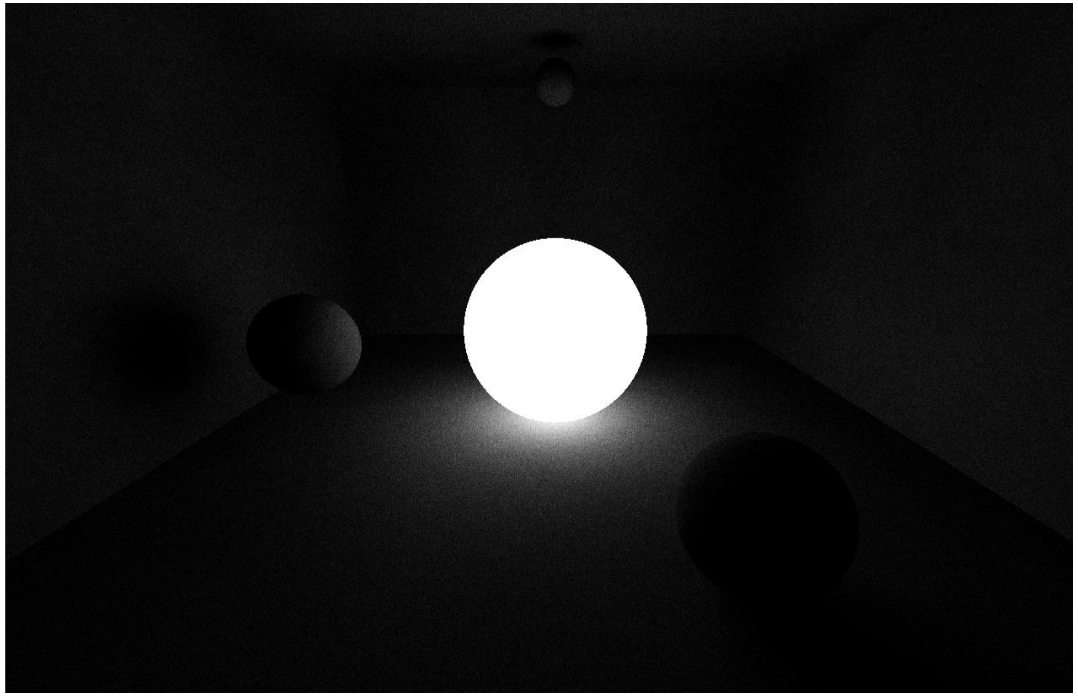
ρ – albedo [0,1] – defines how much light is absorbed by the surface

Lambertian BRDF Comparison

$f_r = 1, n = 100, p(\boldsymbol{\omega}_i) = \frac{cos\theta}{\pi}$

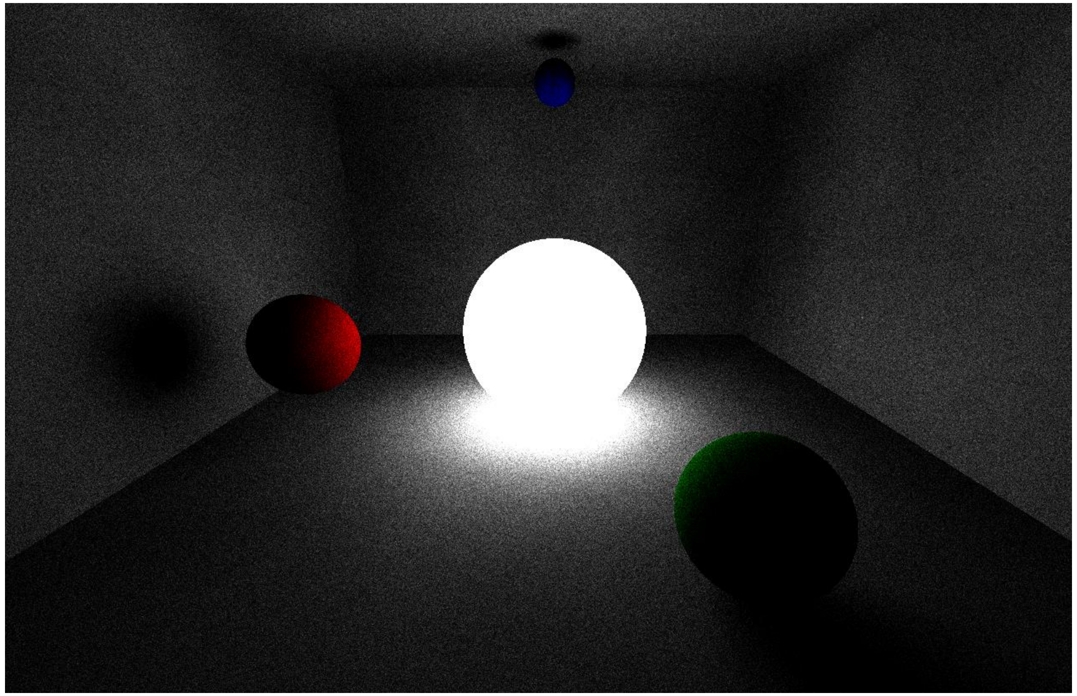


$f_r = \frac{1}{\pi}, n = 100, p(\boldsymbol{\omega}_i) = \frac{cos\theta}{\pi}$

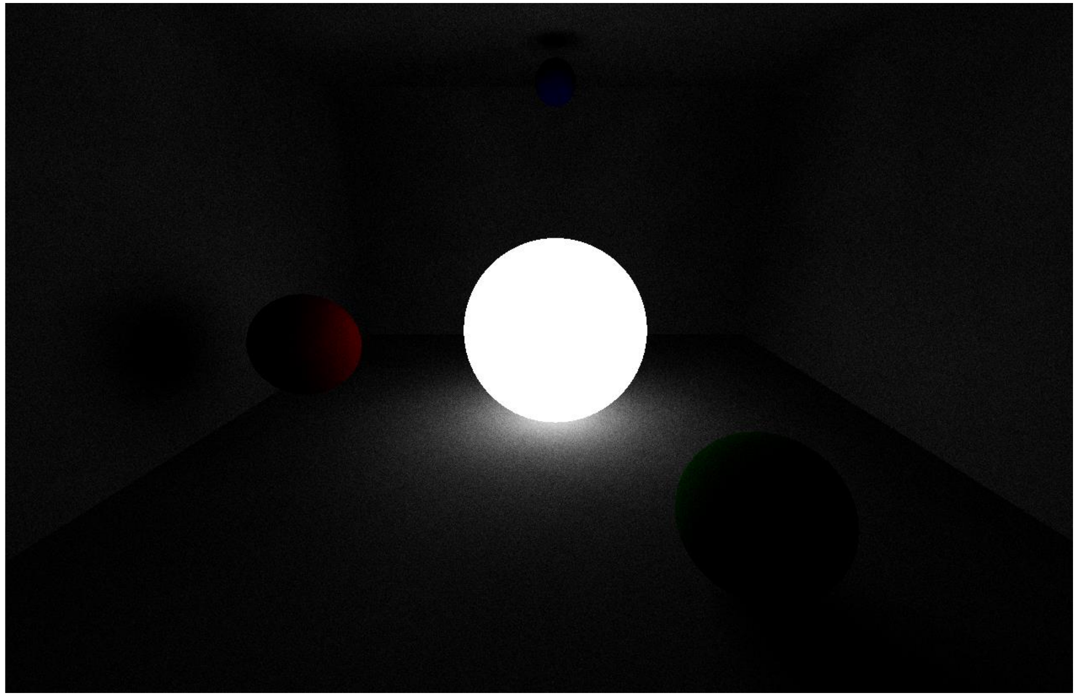


Lambertian BRDF Comparison

$f_r = \rho, n = 100, p(\boldsymbol{\omega}_i) = \frac{cos\theta}{\pi}$

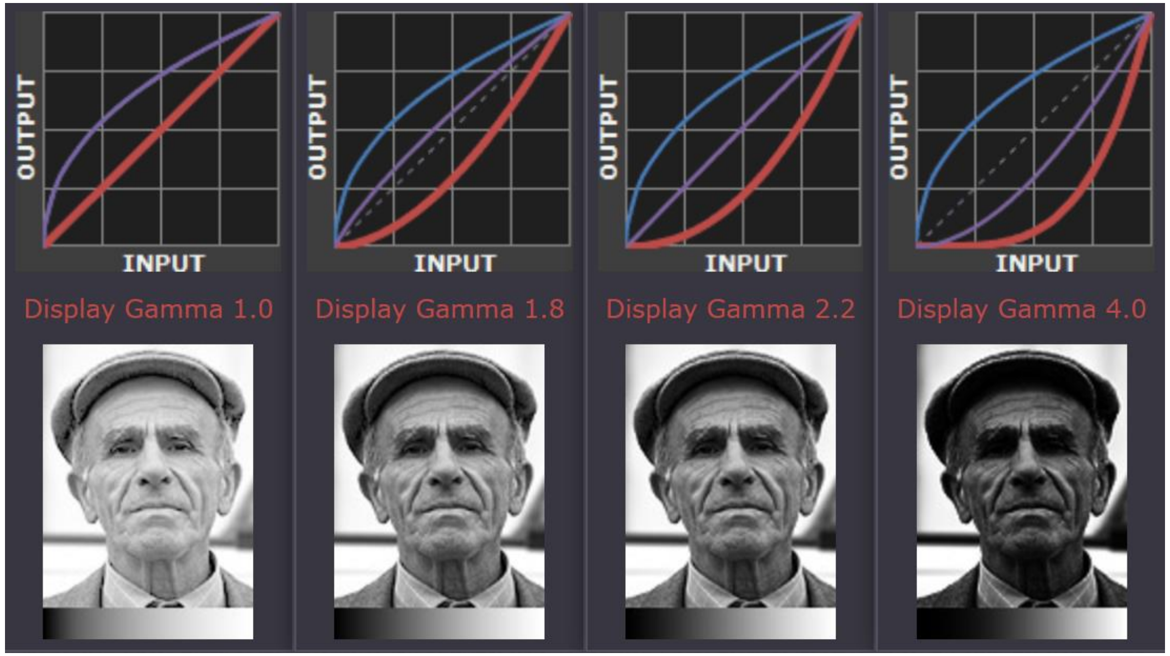


$f_r = \frac{\rho}{\pi}, n = 100, p(\boldsymbol{\omega}_i) = \frac{cos\theta}{\pi}$



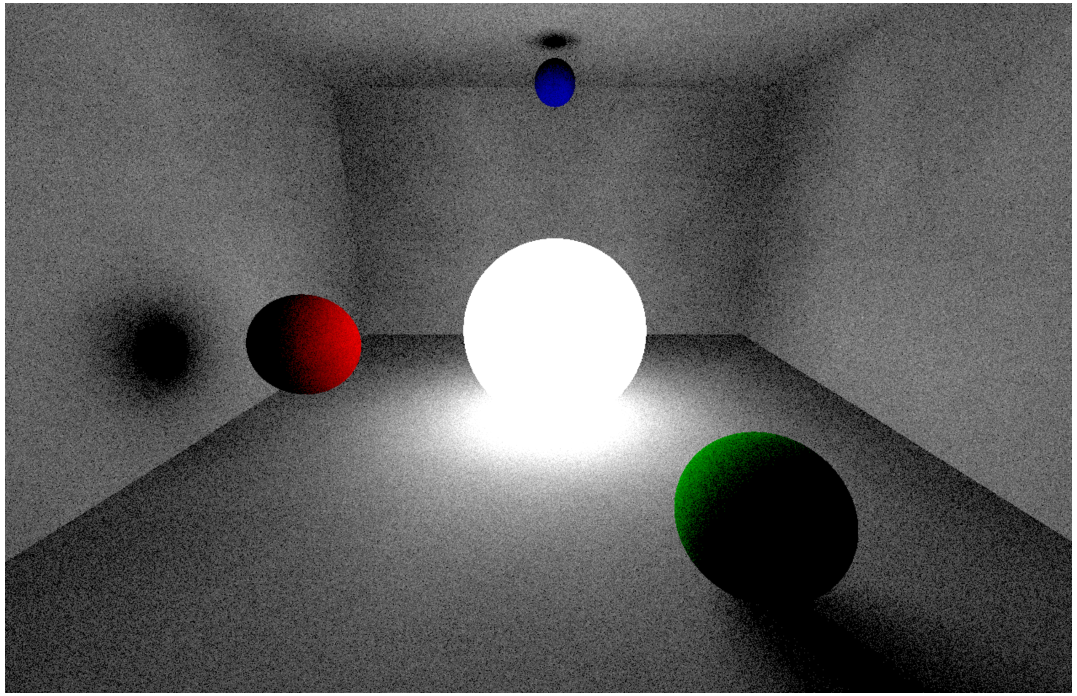
Gamma Correction

- Humans do not perceive luminance linearly
  - much more sensitive to changes in dark tones
- We can use gamma correction to fix that
  - Nonlinear operation used to encode and decode luminance
  - $output = input^{\gamma}$

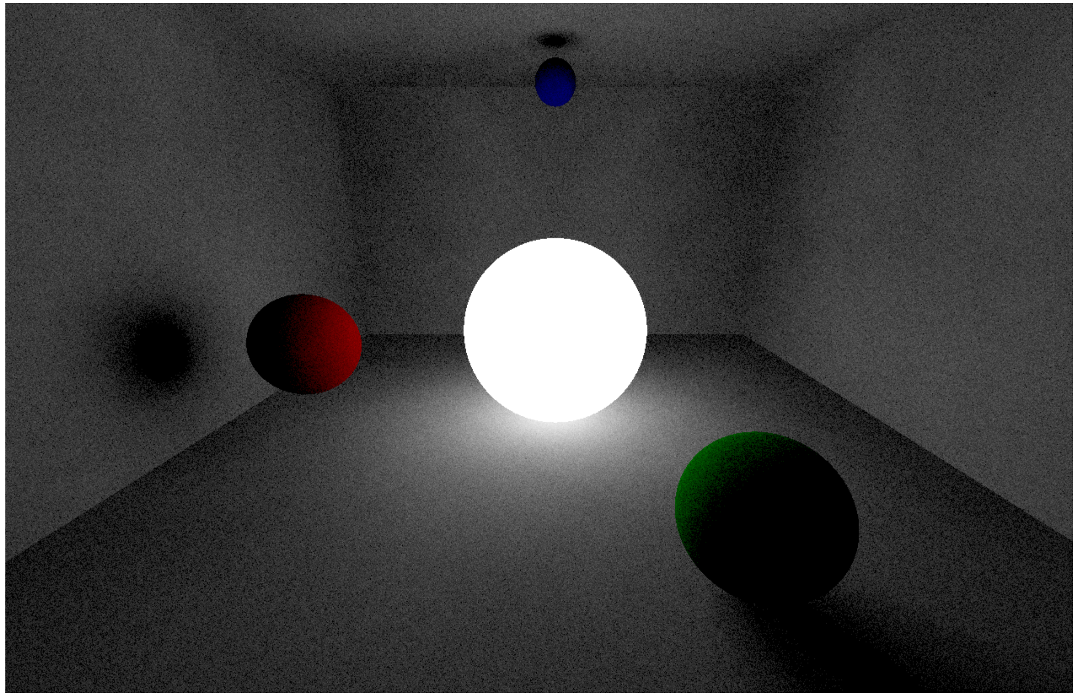


Lambertian BRDF Comparison (with Gamma Correction)

$f_r = \rho, n = 100, p(\boldsymbol{\omega}_i) = \frac{cos\theta}{\pi}$

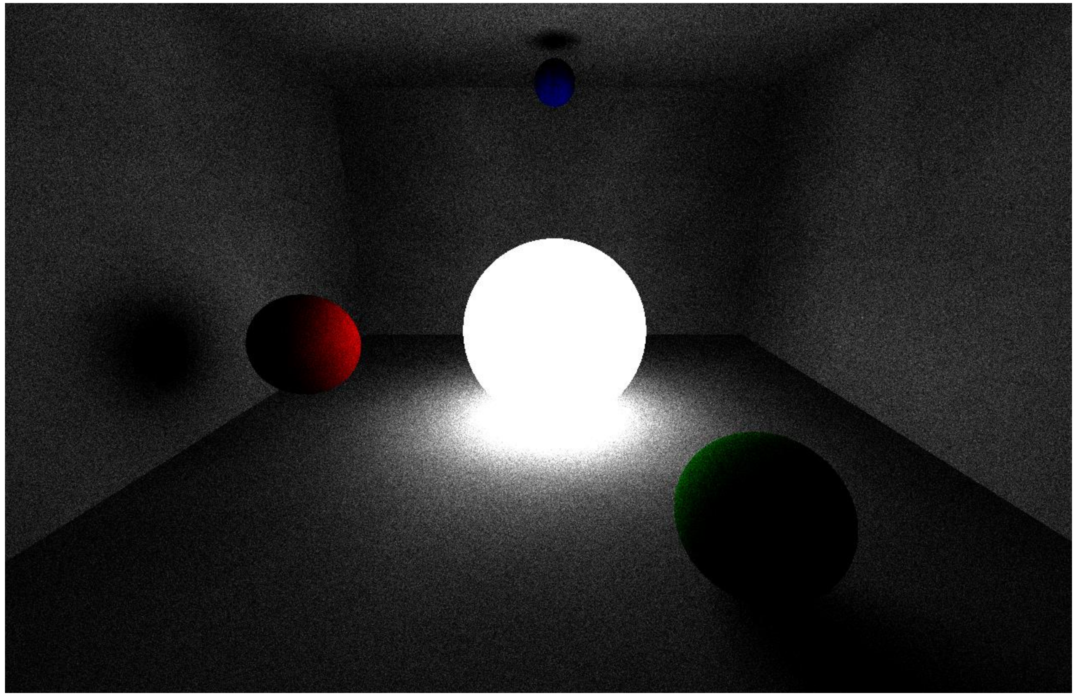


$f_r = \frac{\rho}{\pi}, n = 100, p(\boldsymbol{\omega}_i) = \frac{cos\theta}{\pi}$

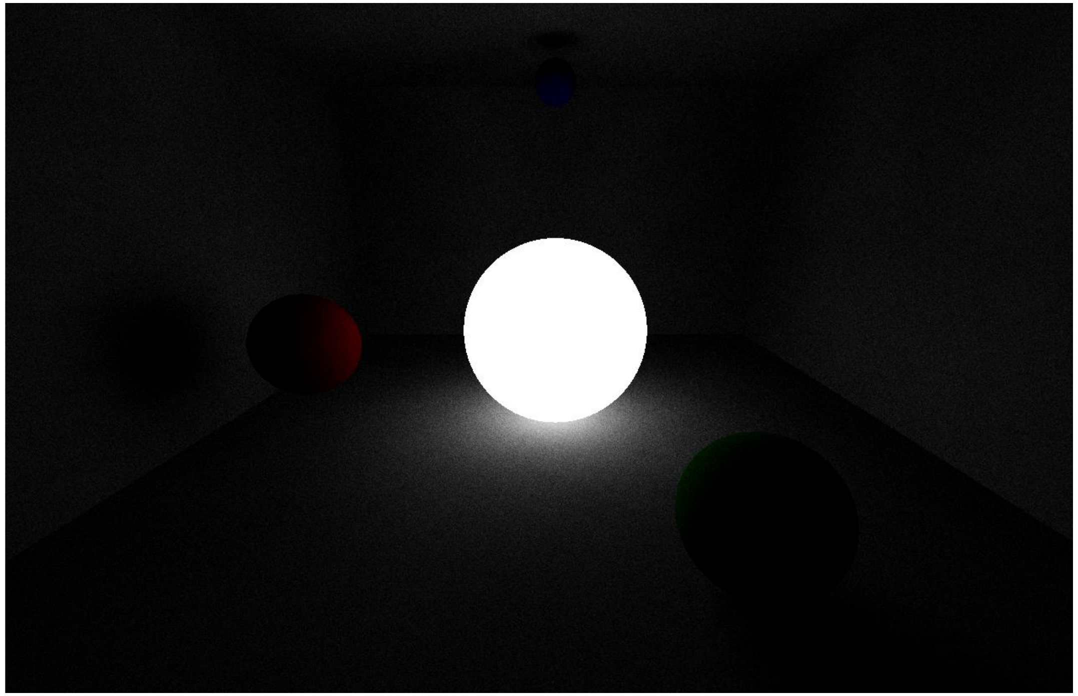


Lambertian BRDF Comparison

$$f_r = \rho, n = 100, p(\boldsymbol{\omega}_i) = \frac{cos\theta}{\pi}$$

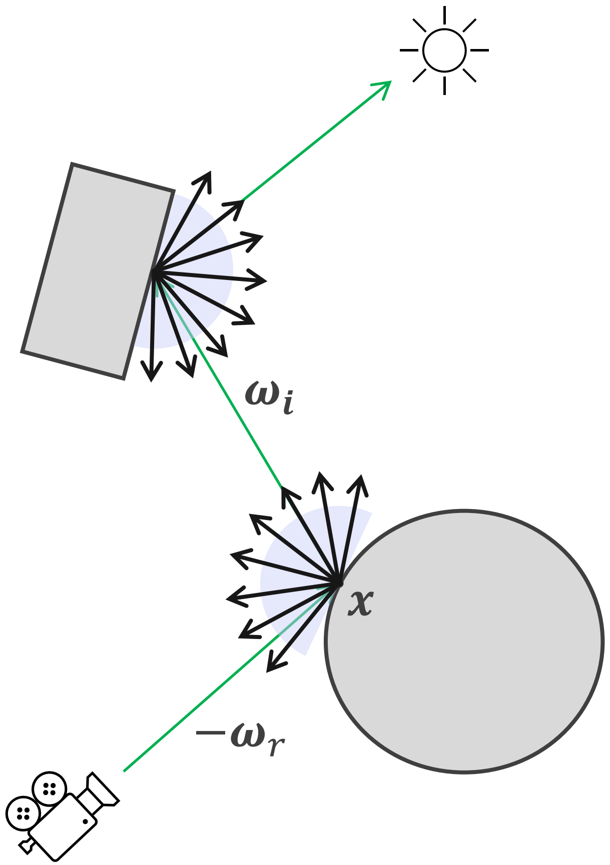


$$f_r = \frac{\rho}{\pi}, n = 100, p(\boldsymbol{\omega}_i) = \frac{cos\theta}{\pi}$$



Evaluating Rendering Function

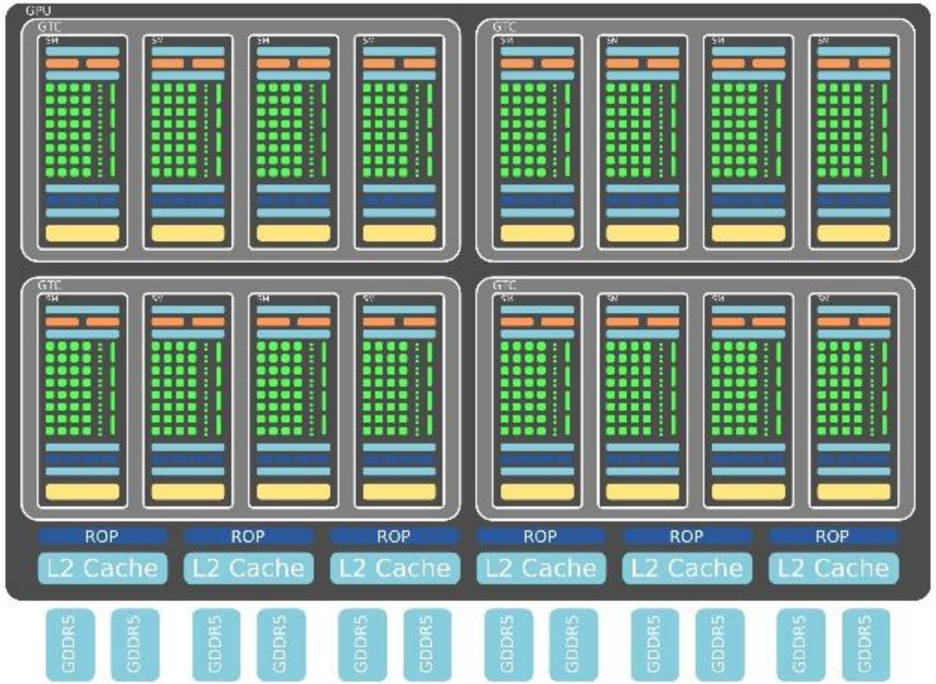
$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$




```
Trace(ray) {
    radiance ← (0,0,0)
    hit ← ClosestHit(ray)
    if(hit == miss) return radiance

    Le = hit.material.emission

    for(i=0; i < samples; i++) {
        sample ← GetNextSample(hit, ray)
        next_ray ← Ray(hit.intersection + epsilon * hit.normal, sample.direction)
        next_hit ← ClosestHit(next_ray);
        if(next_hit == miss || sample.pdf == 0) continue //i.e., skip this sample

        brdf ← ComputeBRDF(hit, next_ray.direction, -ray.direction)
        emission ← next_hit.material.emission
        radiance += brdf * Trace(next_ray) * dot(hit.normal, next_ray.direction) / sample.pdf
    }
    return Le + radiance / samples
}
```

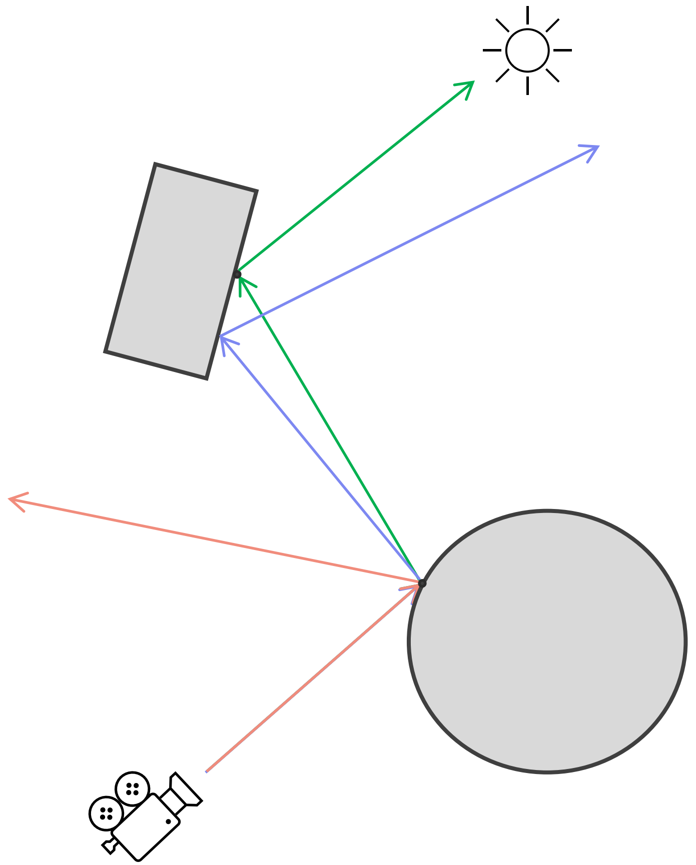
Recursion on GPU

- Not a good idea as GPU is heavy SIMD architecture
  - Cache divergence
  - Flow divergence
- OpenGL/GLSL does not support/allow it
- Solution:
  - Rewrite using custom stack/queue
    - Pain but it works (no dynamic allocation)
  - Utilize properties of Monte Carlo integration
    - Random samples can be accumulated over time



Path Tracing

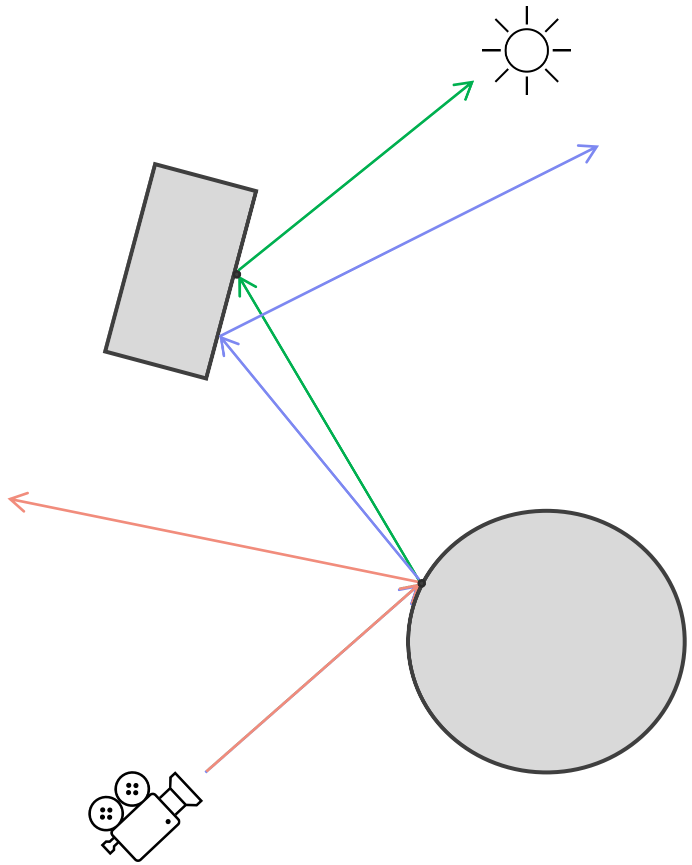
$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$



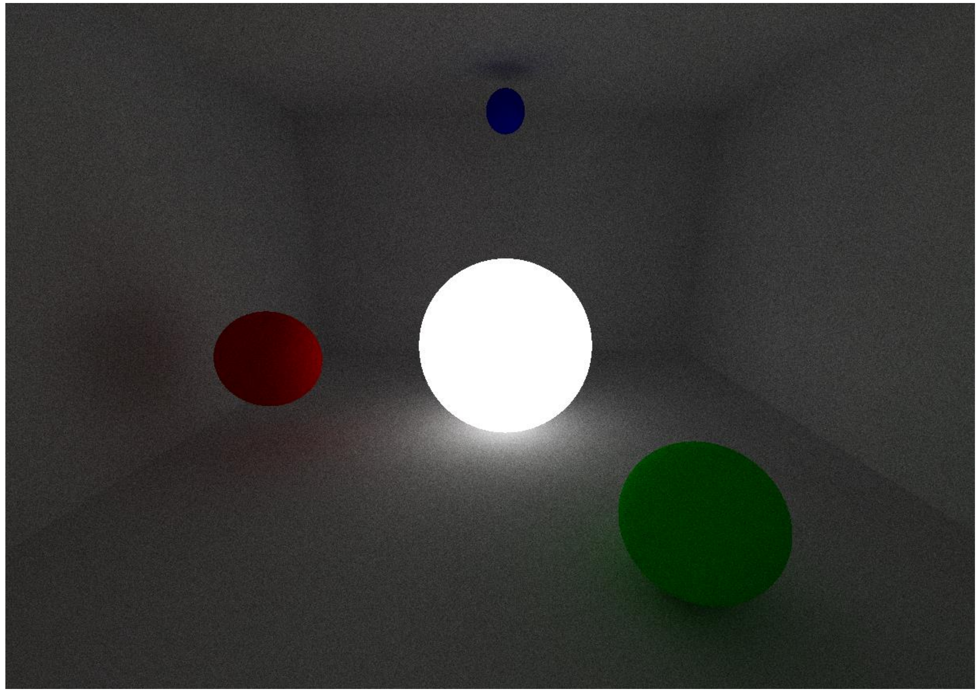
```
Trace(ray) {
    radiance ← (0,0,0)
    throughput ← (1,1,1)

    for(i=0; i < bounces; i++) {
        hit ← ClosestHit(ray)
        if(hit == miss) return radiance
        if(hit.emissive){
            radiance += hit.material.emission * throughput
            return radiance
        }
        sample ← GetNextSample(hit, ray)
        brdf ← ComputeBRDF(hit, sample.direction, -ray.direction)

        throughput *= brdf * dot(hit.normal, sample.direction) / sample.pdf

        ray ← Ray(hit.intersection + epsilon * hit.normal, sample.direction)
    }
    return radiance
}
```

Path Tracing

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$

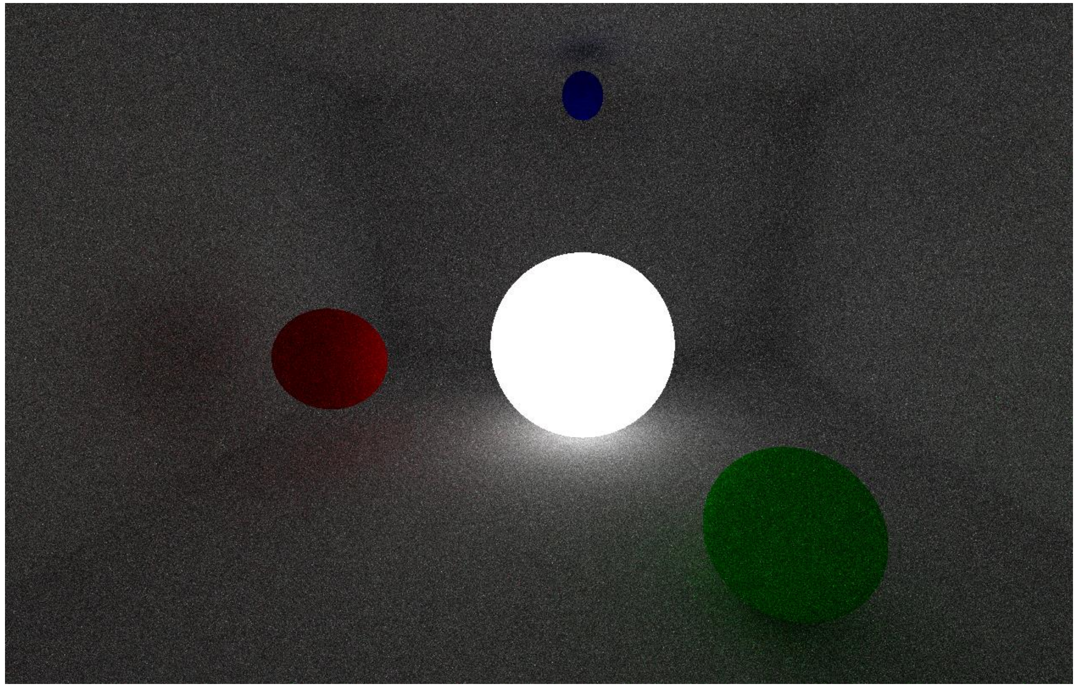


$$f_r = \frac{\rho}{\pi}, n = 100, p(\boldsymbol{\omega}_i) = \frac{cos\theta}{\pi}, bounces = 10$$

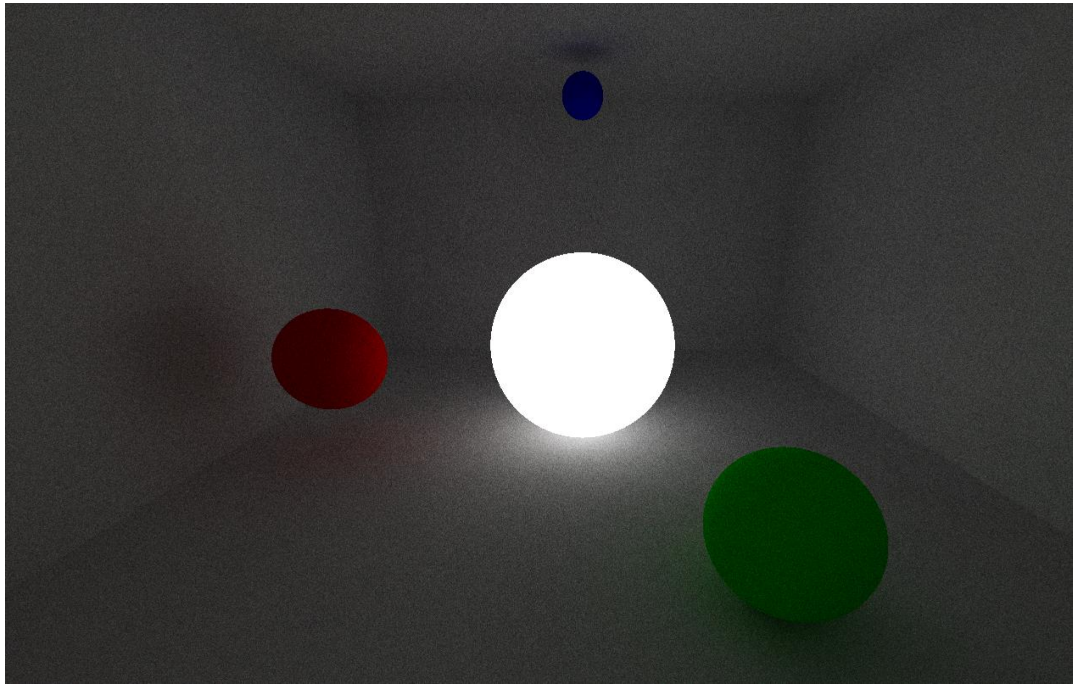


Comparision

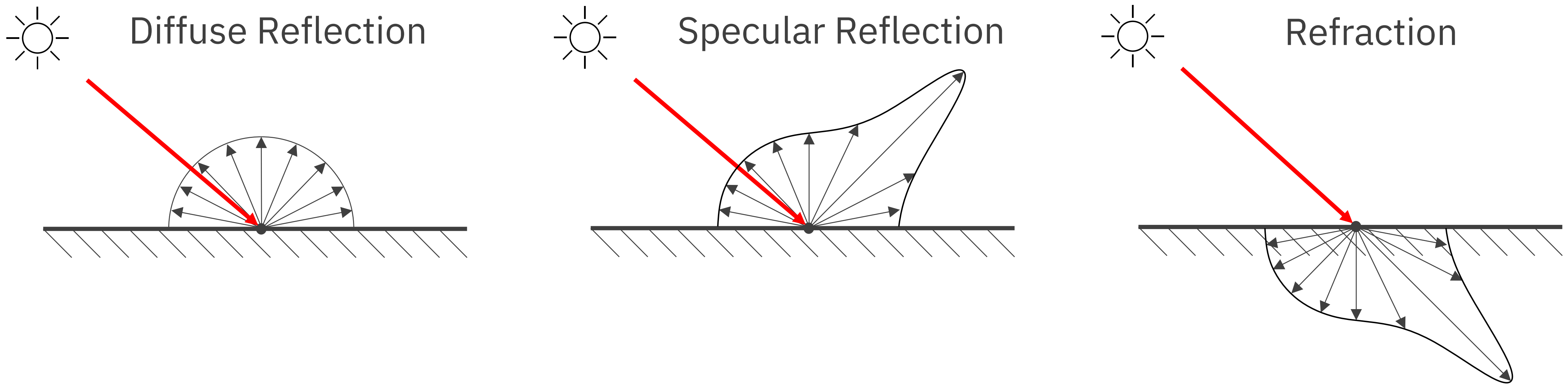
$f_r = \frac{\rho}{\pi}, n = 100, p(\boldsymbol{\omega}_i) = \frac{1}{2\pi}, bounces = 10$



$f_r = \frac{\rho}{\pi}, n = 100, p(\boldsymbol{\omega}_i) = \frac{cos\theta}{\pi}, bounces = 10$



Surface Lighting Effects



HOMEWORK HELP

Reflection & Refraction

- Described by Fresnel equations (S,P polarizations)

$$R_s = \left| \frac{(\eta_1 * \cos(\theta_i) - \eta_2 * \cos(\theta_t))}{(\eta_1 * \cos(\theta_i) + \eta_2 * \cos(\theta_t))} \right|^2$$

$$R_p = \left| \frac{(\eta_1 * \cos(\theta_t) - \eta_2 * \cos(\theta_i))}{(\eta_1 * \cos(\theta_t) + \eta_2 * \cos(\theta_i))} \right|^2$$

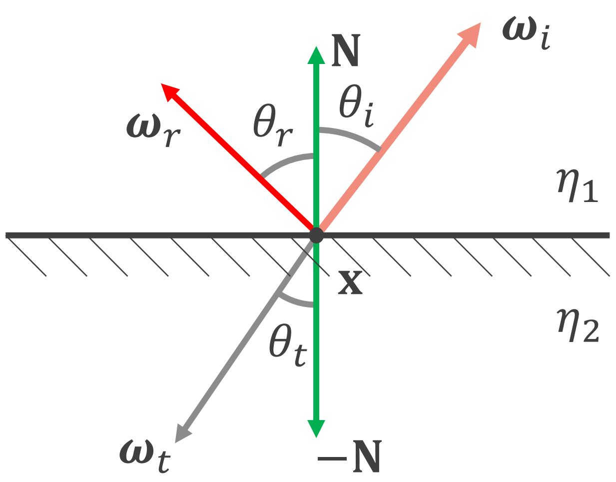
$$R_{eff} = \frac{1}{2}(R_s + R_p)$$

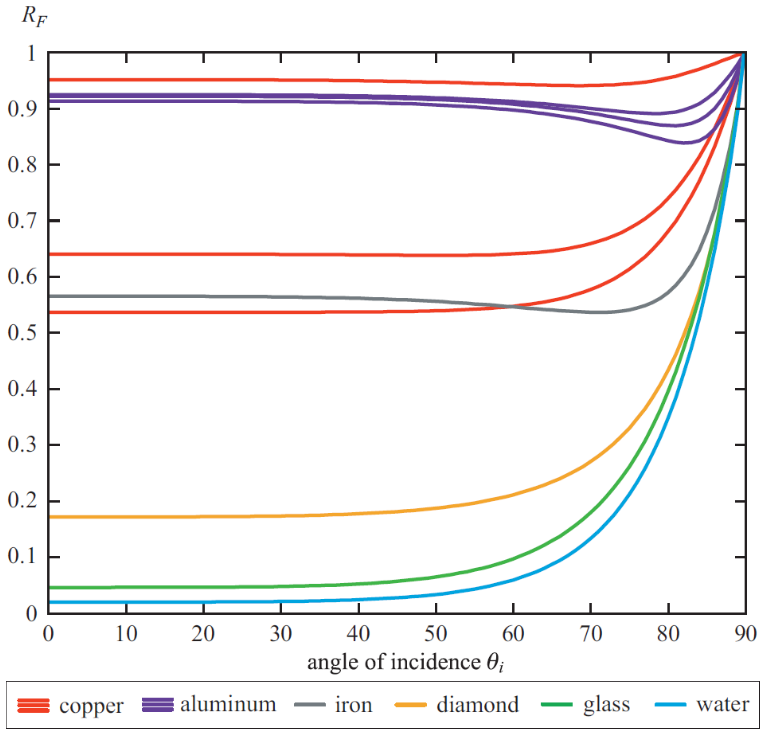
η_1, η_2 – Indices of refraction

- Schlick's approximation:

$$F_{Schlick}(R_0, \boldsymbol{\omega}_i, N) = R_0 + (1 - R_0)(1 - \boldsymbol{\omega}_i \cdot N)^5$$

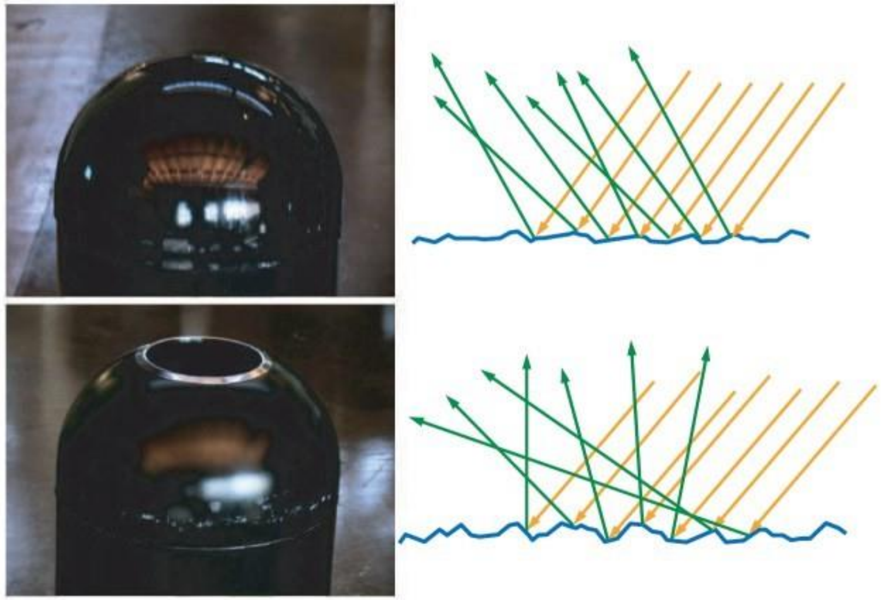
in GLSL you may want to **clamp** this to [0,1]

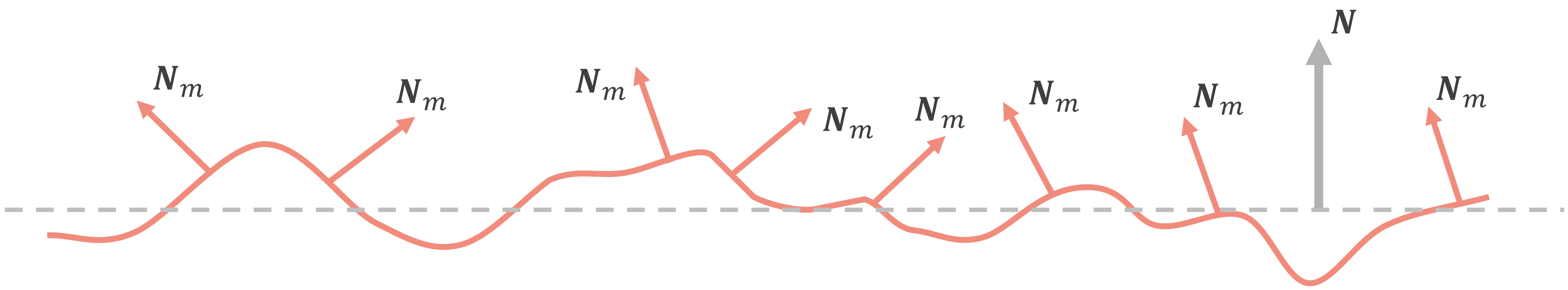




Microfacet Model

- Macrosurface (actual geometry)
- Microsurface (used for illumination)
  - Mirrors smaller than ‘pixel’
  - Not for displacement in geometry
  - Not for normal mapping
- Multiple models exist





Microfacet Model BRDF

Torrance–Sparrow Model

Geometric Attenuation

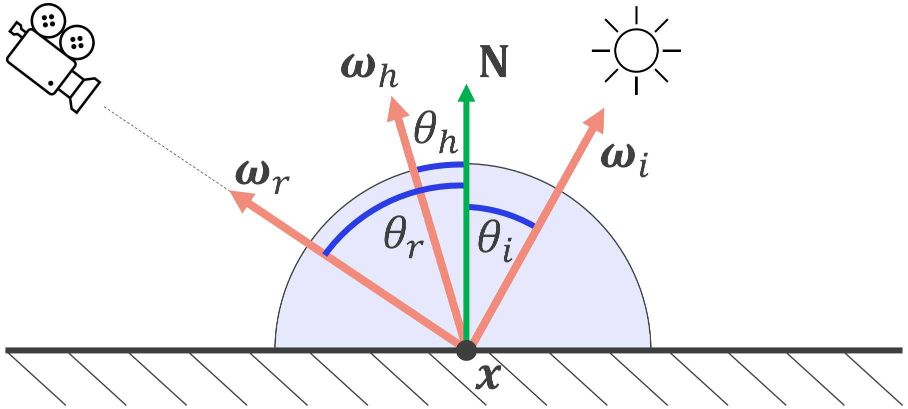
$$\omega_h = \frac{\omega_r + \omega_i}{|\omega_r + \omega_i|}$$

Fresnel

Normal Distribution Function

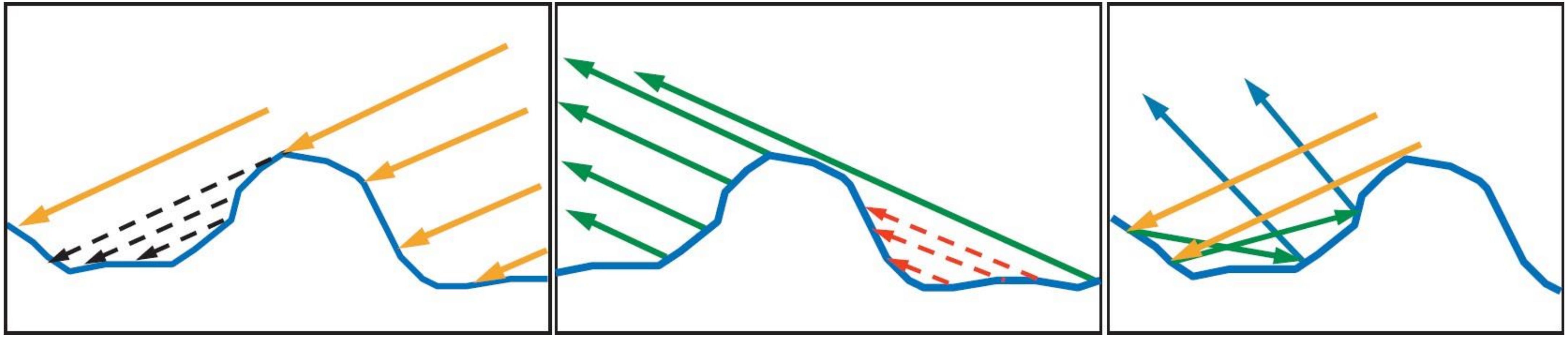
$$f_r(x, \omega_i, \omega_r) = \frac{F(\omega_i, \omega_h)G(\omega_i, \omega_r)D(\omega_h)}{4\cos\theta_i\cos\theta_r}$$

Normalization



Geometric Attenuation

- Shadowing: Facets occlude the light for other facets
- Masking: Facets cannot be seen due to other facets
- Interreflection: Facets reflect the light to other facets, and then the light is reflected to the viewer



Geometric Attenuation

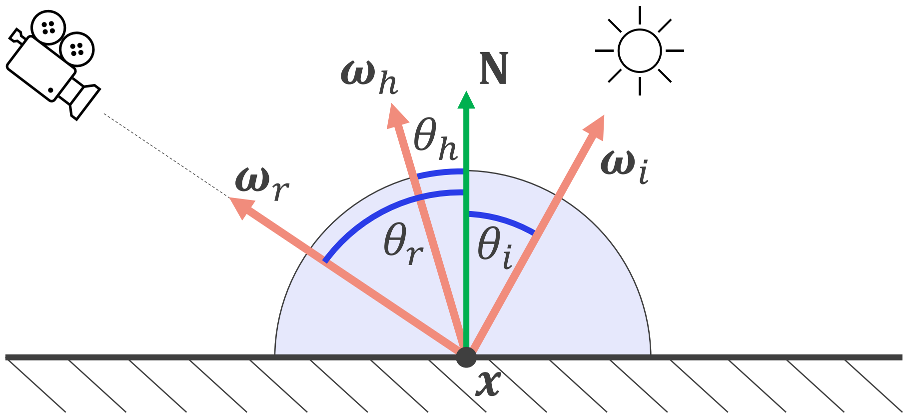
HOMEWORK HELP

- Multiple models for geometric attenuation exist (α stands for roughness)
  - E.g., Smith GGX
    - $\mathrm{G}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_r) = \frac{2\cos\theta_i \cos\theta_r}{\left(\cos\theta_r \sqrt{\alpha^2 + (1-\alpha^2)\cos^2\theta_i}\right) + \left(\cos\theta_i \sqrt{\alpha^2 + (1-\alpha^2)\cos^2\theta_r}\right)}$

Normal Distribution Function

HOMEWORK HELP

- Multiple NDFs exist α stands for roughness of the material
  - Smith GGX
    - $D(w_h) = \frac{\alpha^2}{\pi\left((\alpha^2-1)\cos^2\theta_h+1\right)^2}$
  - Beckmann
    - $D(w_h) = \frac{sin\theta_h}{\pi\alpha^2\cos^3\theta_h} e^{-\frac{\tan^2\theta_h}{\alpha^2}}$
  - Blinn
    - $D(w_h) = \frac{\alpha+2}{2\pi}\cos^{\alpha+1}\theta_h \sin\theta_h$



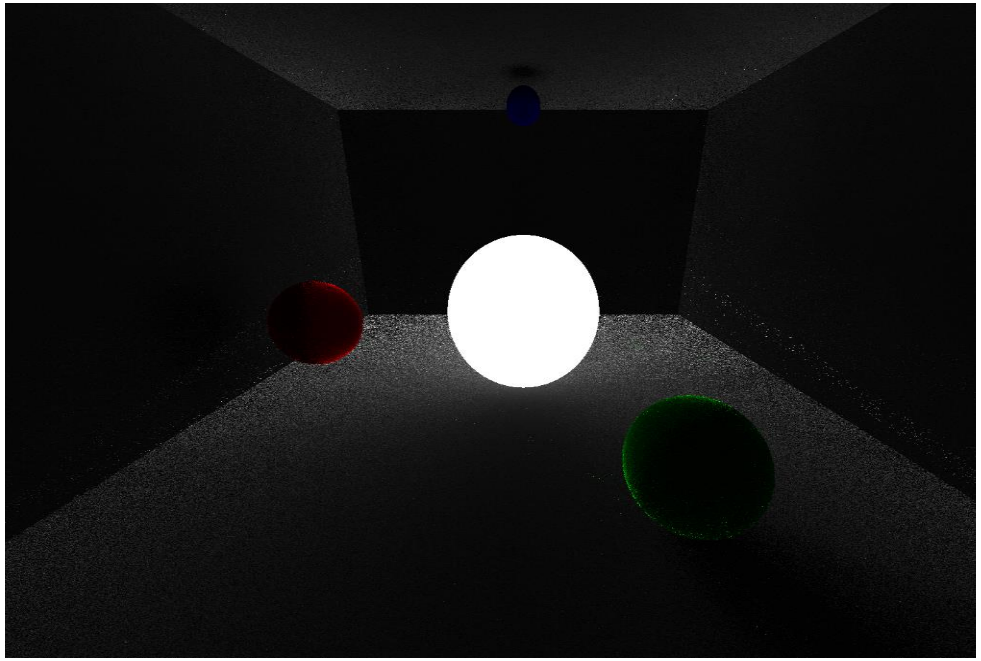
Microfacet Model BRDF

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$

$$f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) = \frac{F(\omega_i, \omega_h) G(\omega_i, \omega_r) D(\omega_h)}{4\cos\theta_i \cos\theta_r}$$

$$p(\boldsymbol{\omega}_i) = \frac{\cos\theta_i}{\pi}$$

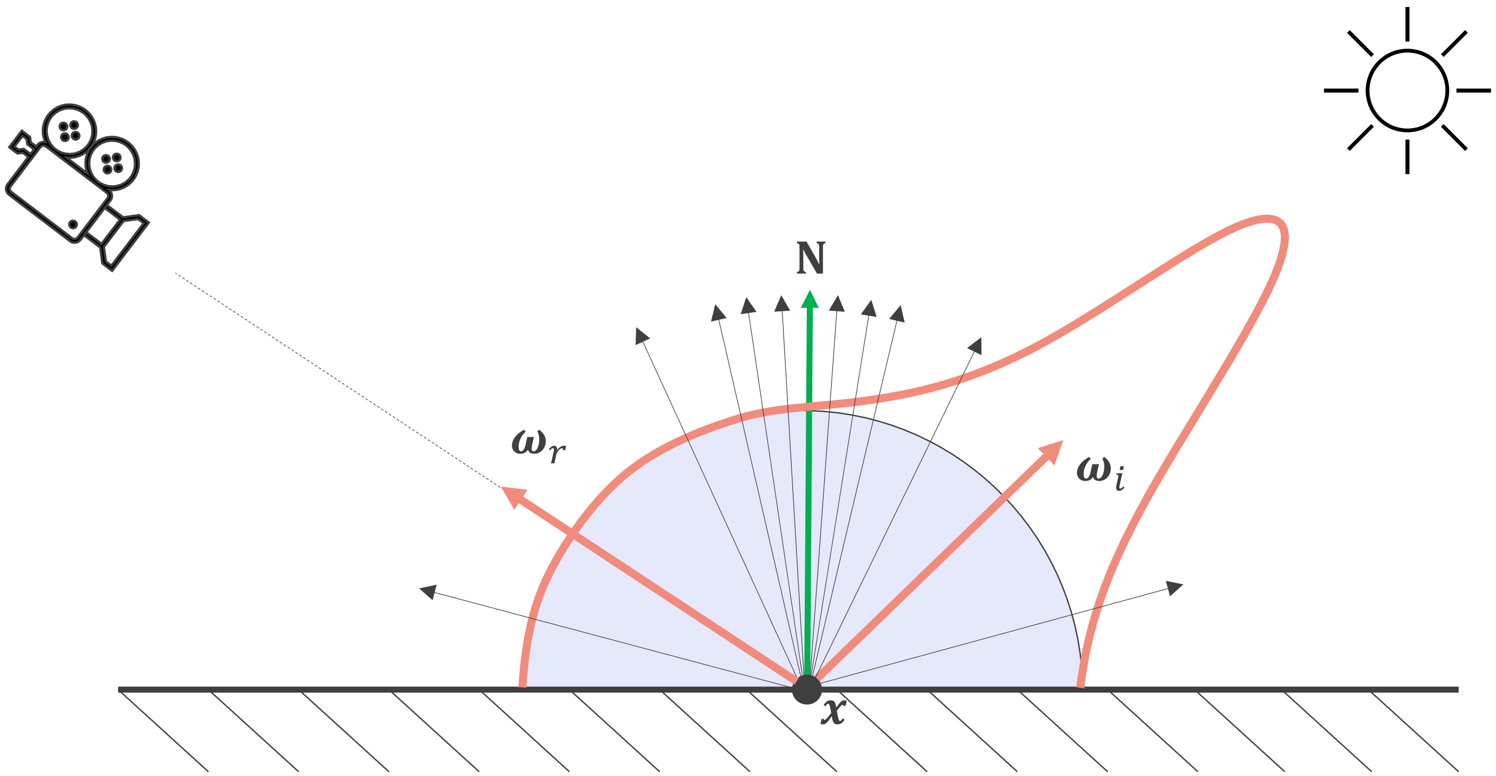
$n = 100, rougness = 0.2, gama = ON$



Sampling BRDF

$$f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) = \frac{F(\omega_i, \omega_h) G(\omega_i, \omega_r) D(\omega_h)}{4 \cos\theta_i \cos\theta_r}$$

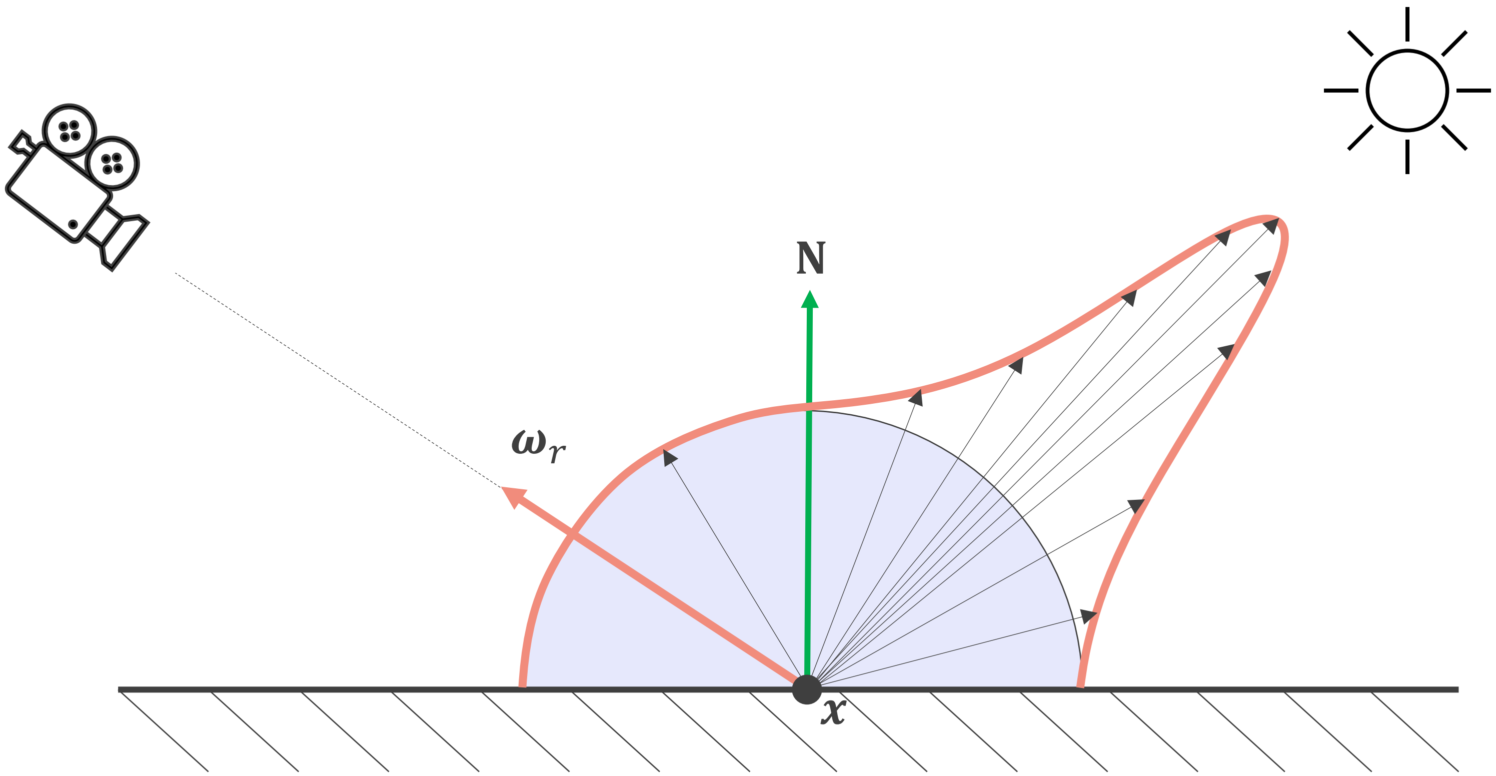
$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n} \sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$



Sampling BRDF

$$f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) = \frac{F(\omega_i, \omega_h) G(\omega_i, \omega_r) D(\omega_h)}{4 \cos\theta_i \cos\theta_r}$$

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$



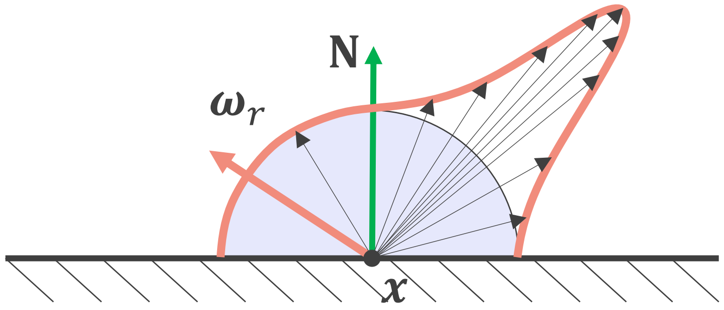
$$p(\omega) \propto f_r(\ldots)$$

Sampling Microfacet BRDF

- The $D(\boldsymbol{\omega}_h)$ has the largest influence on the result

$$f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) = \frac{F(\boldsymbol{\omega}_i, \boldsymbol{\omega}_h)G(\boldsymbol{\omega_i}, \boldsymbol{\omega_r})D(\boldsymbol{\omega}_h)}{4\cos\theta_i \cos\theta_r}$$

- We can use the inversion method to compute sampling according to $D(\boldsymbol{\omega}_h)$
  1. Express $p(\boldsymbol{\omega}) \propto D(\boldsymbol{\omega}_h)$
  2. Convert $p(\boldsymbol{\omega})$ to $p(\theta, \varphi)$ and separate to $p(\theta)\ p(\varphi)$
  3. Compute $\mathrm{P}(\theta)$ and $\mathrm{P}(\varphi)$ by integrating $p(\theta)$ an $p(\varphi)$, respectively
  4. Compute $\mathrm{P}^{-1}(\theta)$ and $\mathrm{P}^{-1}(\varphi)$
  5. Use canonical uniform distribution to sample hemisphere
  6. Convert back to Cartesian coordinates



Example GGX

HOMEWORK HELP

https://agraphicsguynotes.com/posts/sample_microfacet_brdf

$$D(\omega_h) = \frac{\alpha^2}{\pi\left((\alpha^2 - 1)\cos^2\theta_h + 1\right)^2}$$

PDF respecting solid angle

$$p(\omega_h) = \frac{\alpha^2 \cos\ \theta_h}{\pi\left((\alpha^2 - 1)\cos^2\theta_h + 1\right)^2} \rightarrow p(\theta_h, \varphi_h) = \frac{\alpha^2 \cos\ \theta_h \sin\ \theta_h}{\pi\left((\alpha^2 - 1)\cos^2\theta_h + 1\right)^2}$$

$$\boldsymbol{\theta_h = \arccos\sqrt{\frac{1 - \xi_1}{\xi_1(\alpha^2 - 1) + 1}}}$$

$$\boldsymbol{\varphi_h = 2\pi\xi_2}$$

$$\rightarrow \begin{aligned} x_h &= r \sin\theta_h \cos\varphi_h \\ y_h &= r \sin\theta_h\, sin\,\varphi_h \\ z_h &= r \cos\theta_h \end{aligned}$$

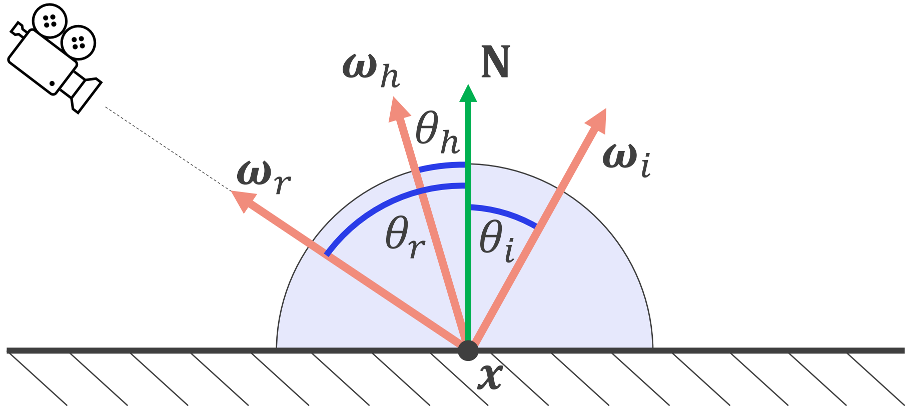
do not forget to convert the sample to world coordinates

Example GGX

HOMEWORK HELP

https://agraphicsguynotes.com/posts/sample_microfacet_brdf

- To compute $\boldsymbol{\omega}_i$ we need to reflect $-\boldsymbol{\omega}_r$ along $\boldsymbol{\omega}_h$
- Assuming $\boldsymbol{\omega}_r' = -\boldsymbol{\omega}_r$
  - $\boldsymbol{\omega}_i = \boldsymbol{\omega}_r' - 2\boldsymbol{\omega}_h(\boldsymbol{\omega}_r' \cdot \boldsymbol{\omega}_h)$ ← in GLSL you may want to use **reflect ($\boldsymbol{\omega}_r'$,$\boldsymbol{\omega}_h$)**
- We need to express the probability with respect to $\boldsymbol{\omega}_i$
  - $p(\boldsymbol{\omega}_i) = \frac{D(\boldsymbol{\omega}_h) \cos \theta_h}{4(\boldsymbol{\omega}_r \cdot \boldsymbol{\omega}_h)}$



Microfacet Model BRDF

HOMEWORK HELP

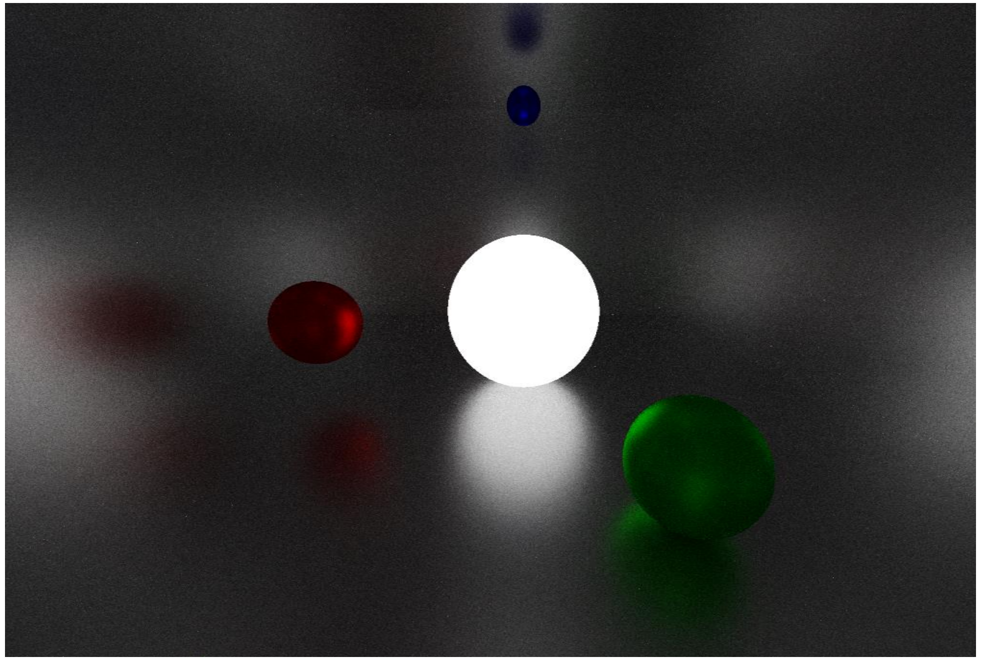
https://agraphicsguynotes.com/posts/sample_microfacet_brdf

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$

$$f_r = \frac{F(\omega_i, \omega_h) G(\omega_i, \omega_r) D(\omega_h)}{4 \cos\theta_i \cos\theta_r}$$

$$p(\boldsymbol{\omega}_i) = \frac{D(\omega_h) \cos\theta_h}{4(\boldsymbol{\omega}_r \cdot \omega_h)}$$

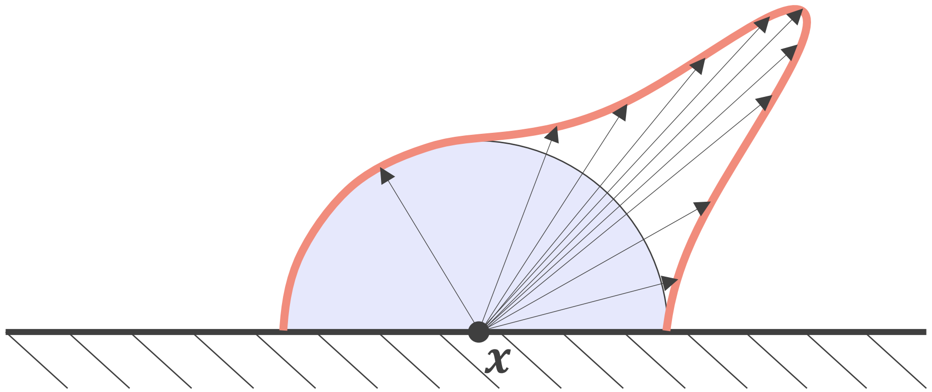
$n = 100, rougness = 0.2, gama = OFF$

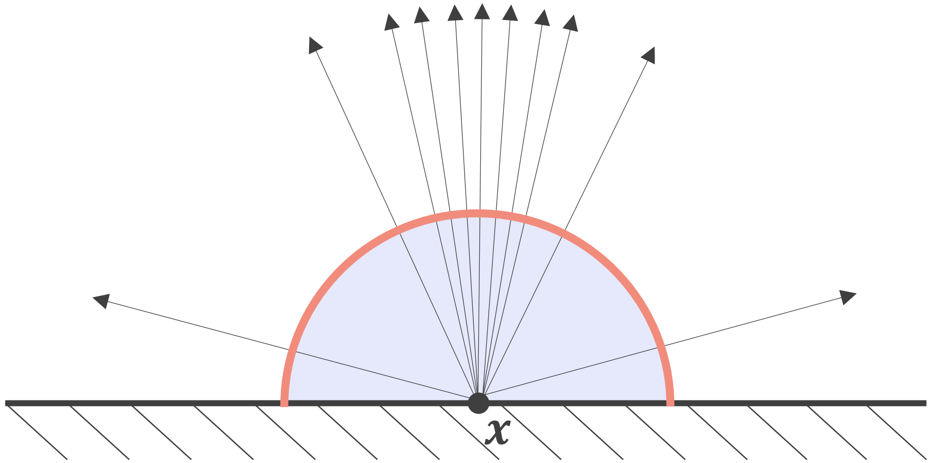


Combining BRDFs

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$

- Sampling BRDF converges faster for specular surfaces
- Sampling according to $\cos\theta_i$ converges faster for diffuse surfaces



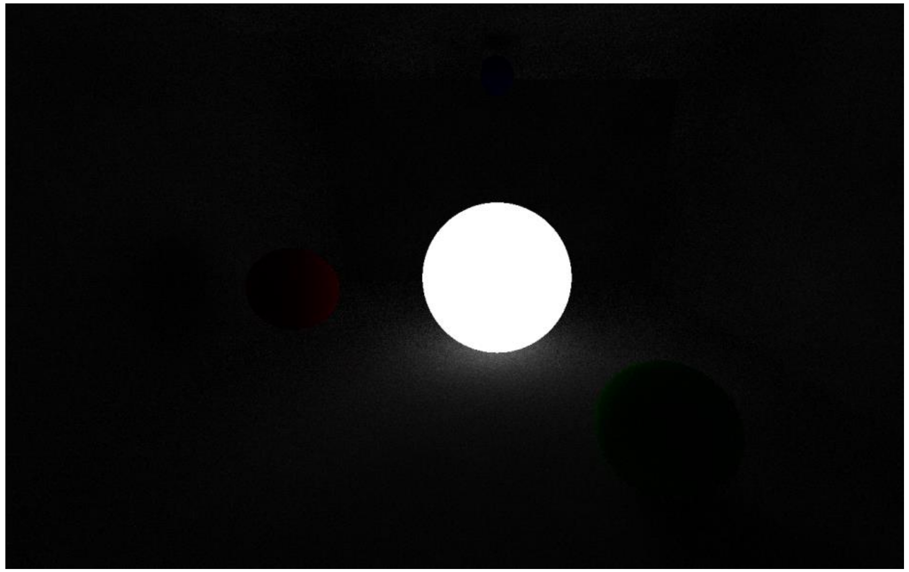


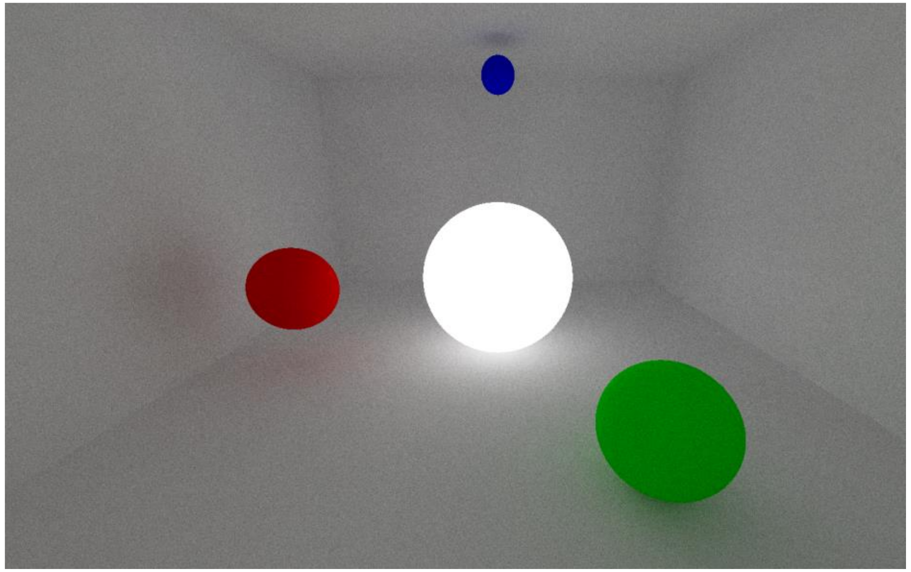
Comparison

$$f_r = \frac{F(\boldsymbol{\omega}_i, \boldsymbol{\omega}_h)G(\boldsymbol{\omega}_i, \boldsymbol{\omega}_r)D(\boldsymbol{\omega}_h)}{4\cos\theta_i \cos\theta_r}$$

$$p(\boldsymbol{\omega}_i) = \frac{D(\boldsymbol{\omega}_h)\cos\theta_h}{4(\boldsymbol{\omega}_r \cdot \boldsymbol{\omega}_h)}$$

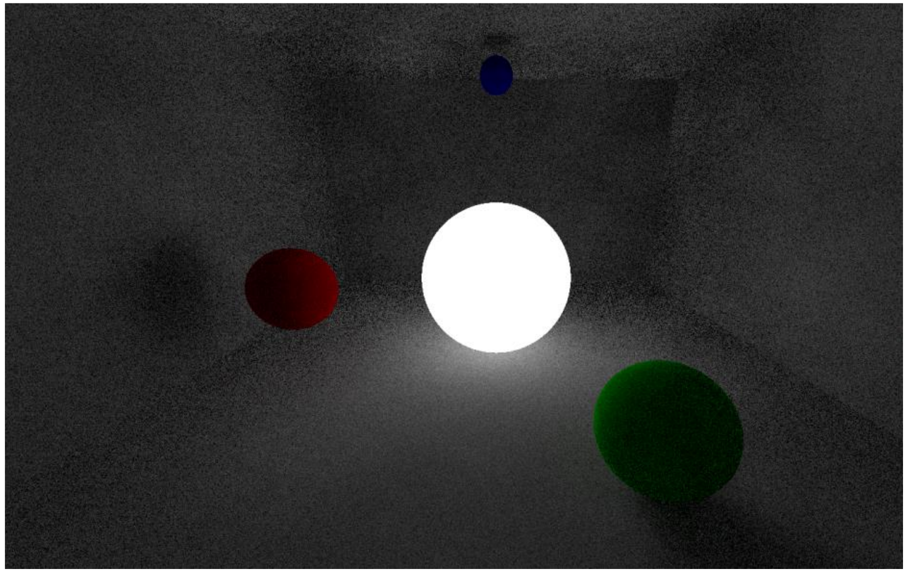
$n = 100, b = 10, rougness = 1.0, gama = OFF$





$f_r = \frac{\rho}{\pi}, p(\boldsymbol{\omega}_i) = \frac{cos\theta}{\pi}, n = 100, b = 10, gama = ON$

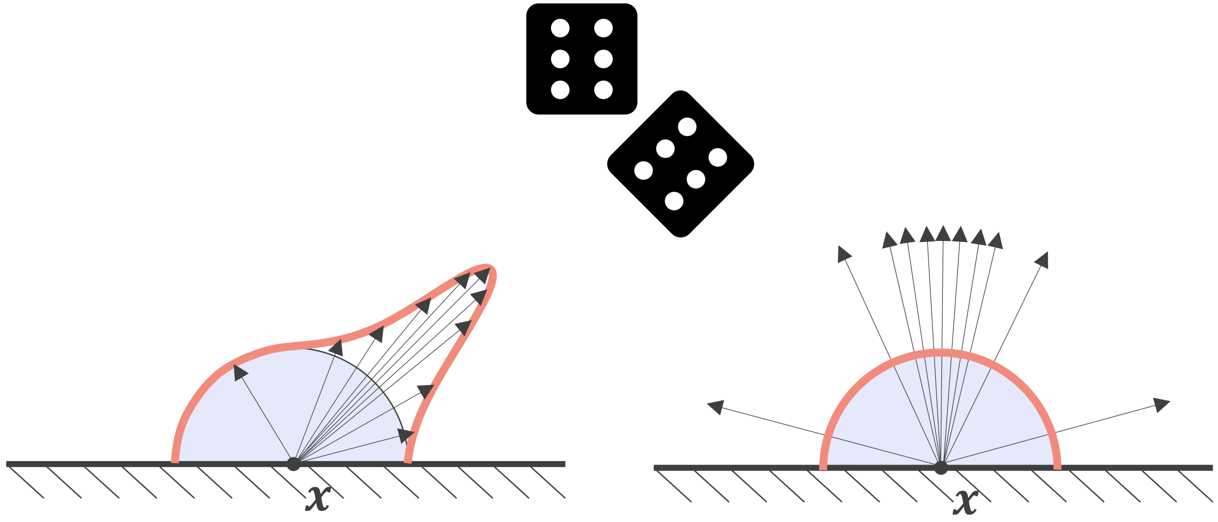
$n = 100, b = 10, rougness = 1.0, gama = OFF$

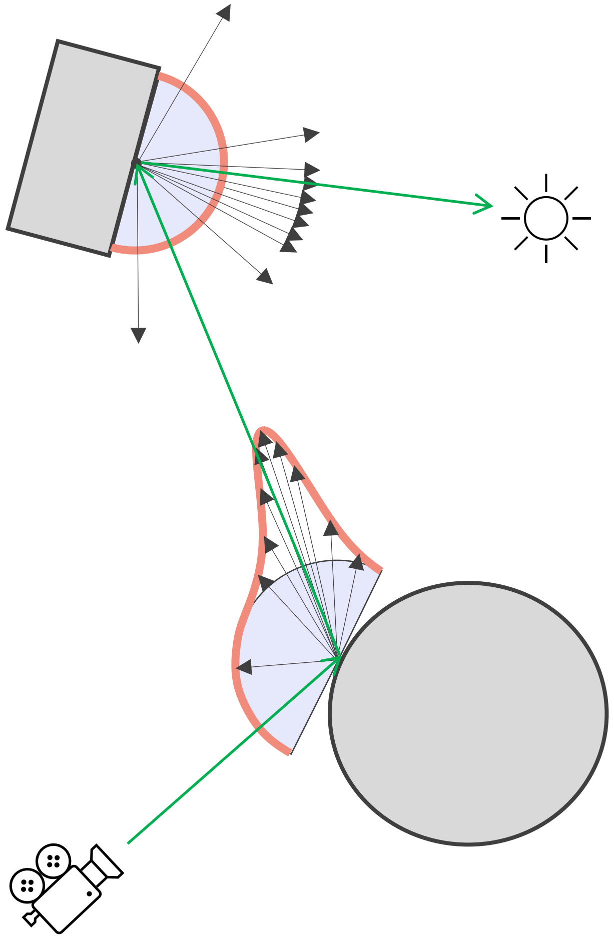


Combining BRDFs

- Mixing two unbiased estimators produces unbiased estimator

$$E\left[\sum_i X_i\right] = \sum_i E[X_i]$$





Combining BRDFs

HOMEWORK HELP

- Mixing two unbiased estimators produces unbiased estimator

$$E\left[\sum_i X_i\right] = \sum_i E[X_i]$$

Algorithm

Draw random number ξ from uniform distribution

If $rougness > \xi$

Compute diffuse $sample$ and corresponding PDF_d

Set specular $PDF_s = 1$

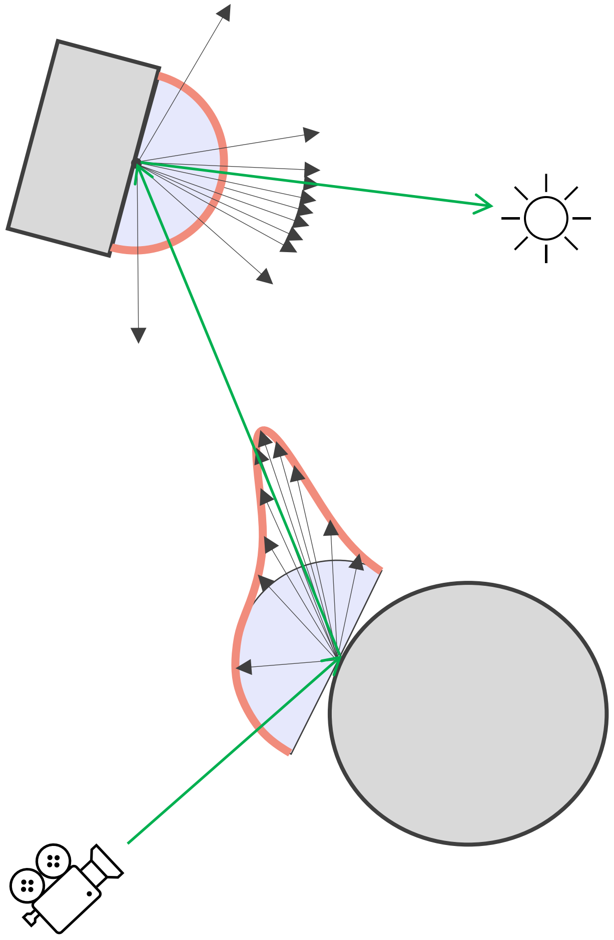
Else

Compute specular (e.g, GGX) $sample$ and corresponding PDF_s

Set diffuse $PDF_d = 1$

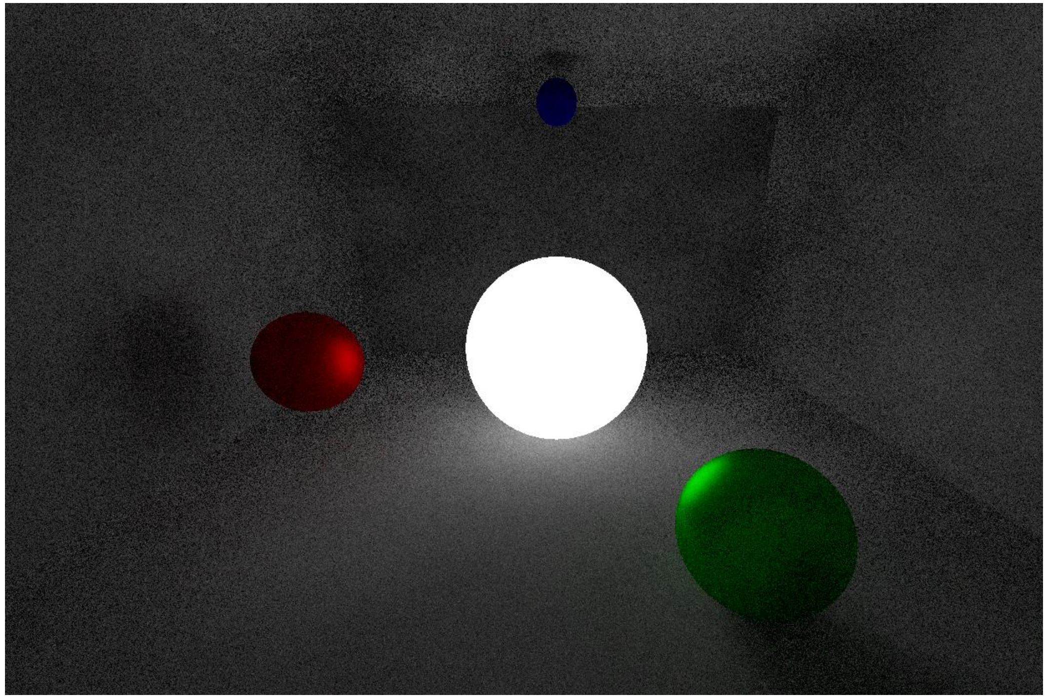
Mix the PDFs based on $rougness$

$$\mathrm{p}(\omega) = (rougness)PDF_d + (1 - rougness)PDF_s$$

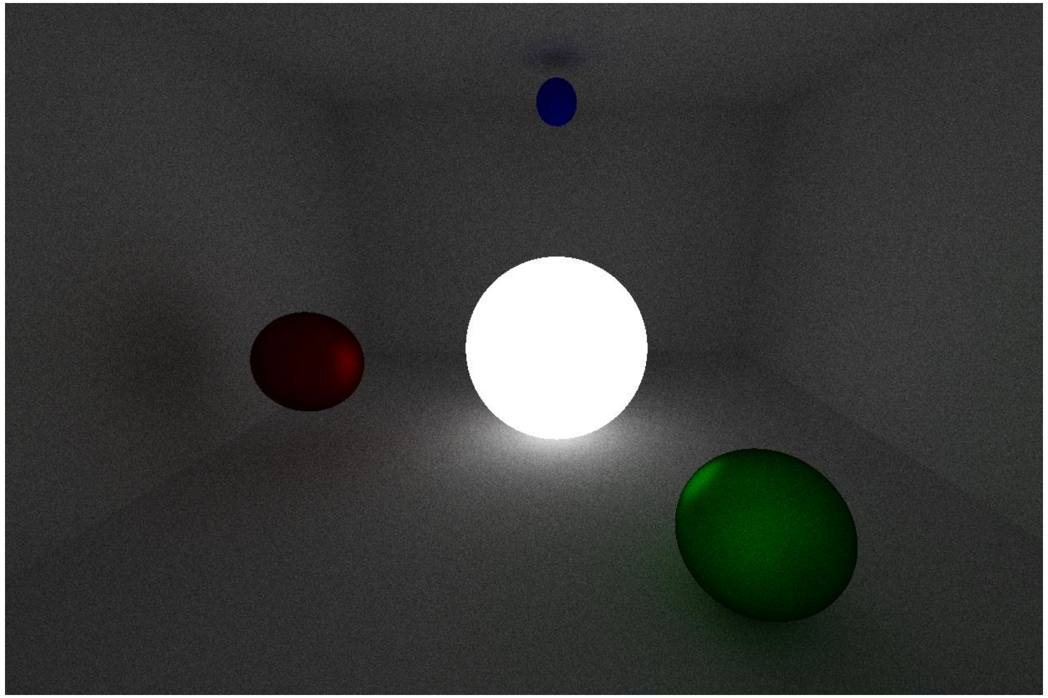


Comparison

GGX Only , $n = 100, b = 10, g = ON$



GGX + Cosine, $n = 100, b = 10, g = OFF$



Light Importance Sampling

$$\int_{\Omega} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \, d\boldsymbol{\omega}_i \rightarrow \int_{A} L_o(\boldsymbol{x}', \boldsymbol{\omega_o}) \boldsymbol{V}(\boldsymbol{x}, \boldsymbol{x}') \frac{\cos\theta_i \cos\theta_i'}{|x - x^2|^2} \, dA$$

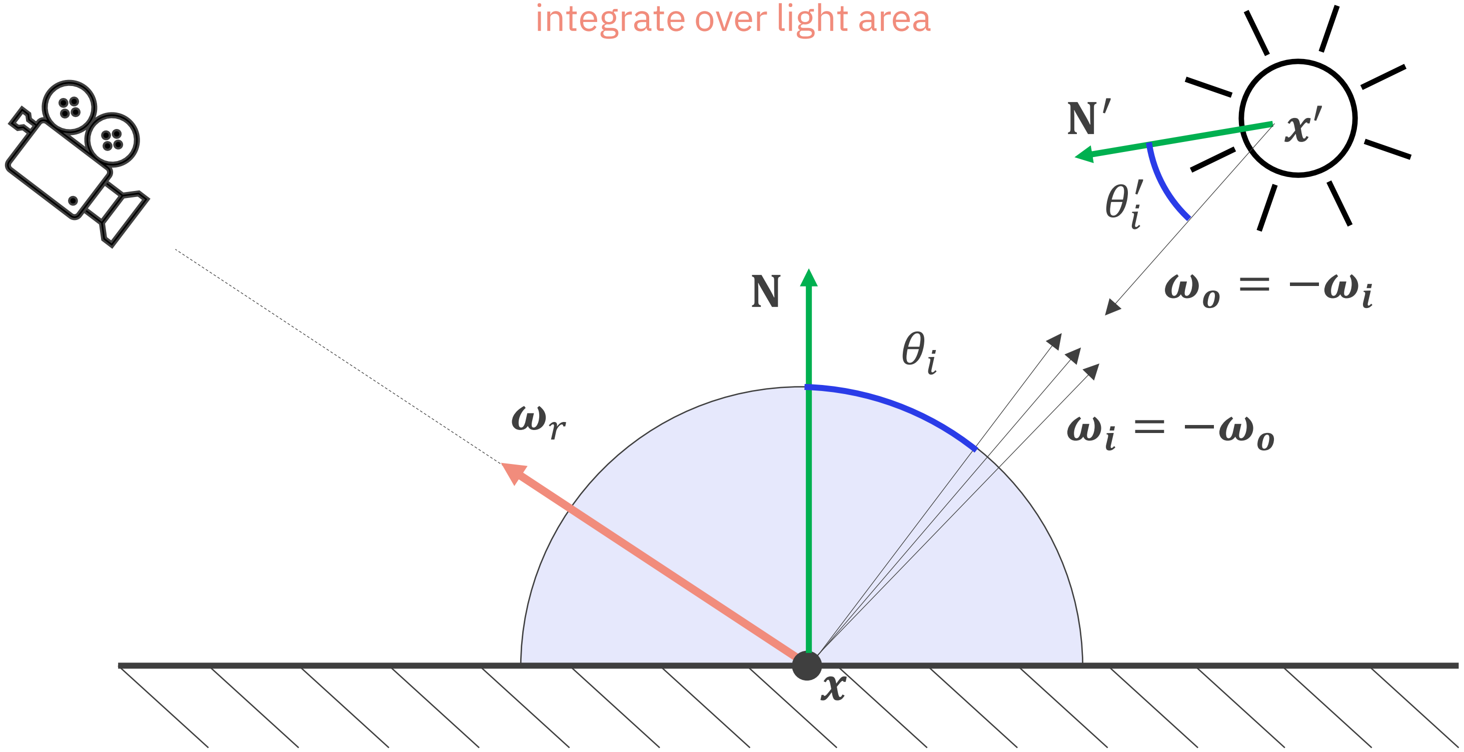
outgoing radiance

visibility term

light orientation

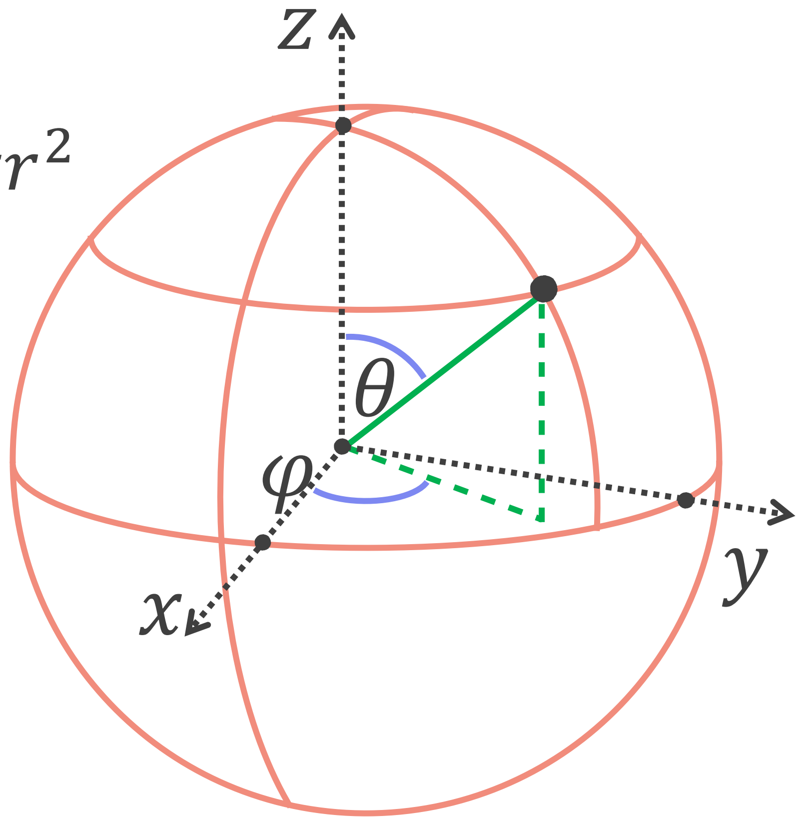
integrate over light area

distance from light



$$p(\omega) \propto \frac{1}{A}$$

Uniform Sphere Sampling



$x = r \sin\theta \cos\varphi$

$y = r \sin\theta \sin\varphi$

$z = r \cos\theta$

$p(r,\theta,\varphi) = r^2 sin\theta\, p(x,y,z)$

$p(\omega) = \frac{1}{4\pi}$ → $p(\theta,\varphi) = \frac{sin\theta}{4\pi}$

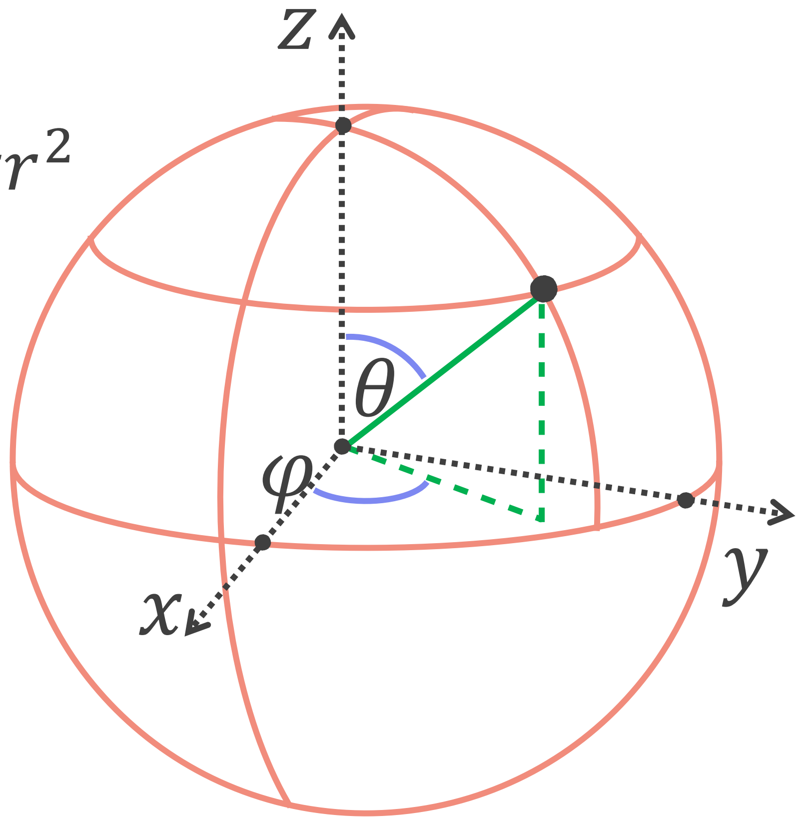
note that we ignore r

$p(\theta) = sin\theta$ → $P(\theta) = 1 - \cos\theta$ → $P^{-1}(\xi_1) = \cos^{-1}(1-\xi_1)$ → $\cos^{-1}(\xi_1)$

$p(\varphi) = \frac{1}{4\pi}$ → $P(\varphi) = \frac{\varphi}{4\pi}$ → $P^{-1}(\xi_2) = 4\pi\xi_2$

Uniform Sphere Sampling

$$x = r\sin\theta\cos\varphi = \sqrt{1-\xi_1^2}\cos(4\pi\xi_2) = \sqrt{1-\xi_1^2}\cos(2\pi\xi_2)$$
$$y = r\sin\theta\, sin\,\varphi = \sqrt{1-\xi_1^2}\sin(4\pi\xi_2) = \sqrt{1-\xi_1^2}\sin(2\pi\xi_2)$$
$$z = r\cos\theta = \xi_1$$



$p(r,\theta,\varphi) = r^2 sin\theta\; p(x,y,z)$

$$p(\omega) = \frac{1}{4\pi} \rightarrow p(\theta,\varphi) = \frac{sin\theta}{4\pi}$$

note that we ignore r

$$p(\theta) = sin\theta \rightarrow P(\theta) = 1-\cos\theta \rightarrow P^{-1}(\xi_1) = \cos^{-1}(1-\xi_1) \rightarrow \cos^{-1}(\xi_1)$$

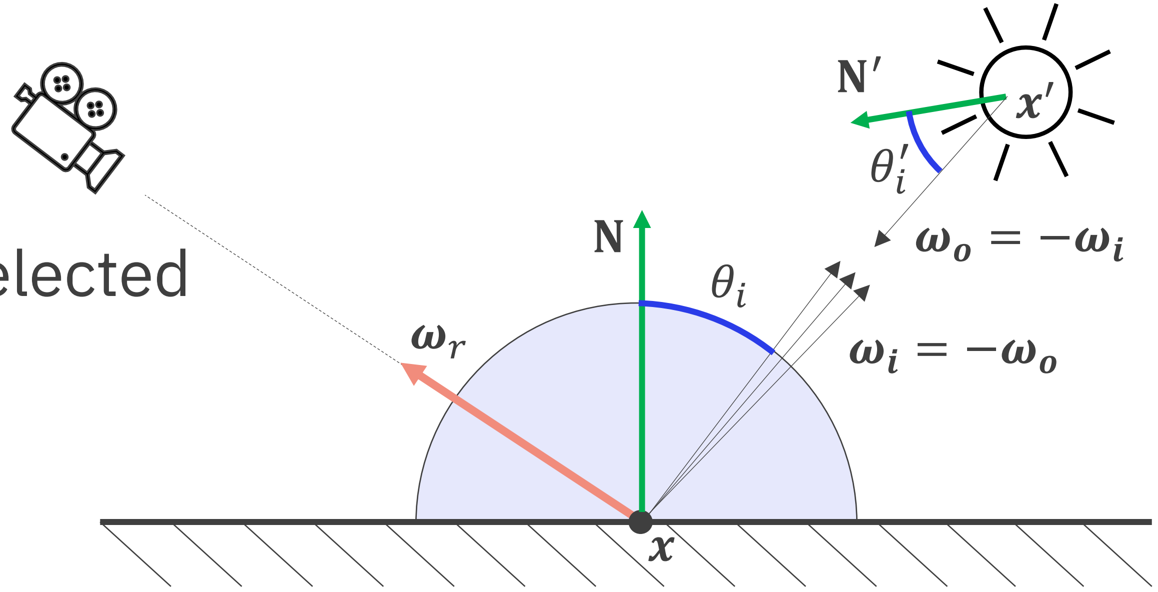
$$p(\varphi) = \frac{1}{4\pi} \rightarrow P(\varphi) = \frac{\varphi}{4\pi} \rightarrow P^{-1}(\xi_2) = 4\pi\xi_2$$

Sampling Spherical Lights

- Pick a random light and obtain its [position, radius] with some probability $\mathrm{p}(\boldsymbol{l})$
  - $[p, r] \leftarrow$ `PickRandomLight()`
- Compute a random sample on a sphere using canonical uniform distribution
  - $s \leftarrow$ `UniformSampleSphere`(ξ_1, ξ_2)
- Compute a sample ray $\boldsymbol{\omega_i}$ to integrate
  - $\boldsymbol{x}' = \boldsymbol{p} + \boldsymbol{s} * \boldsymbol{r}$
  - $\boldsymbol{\omega_i} = \boldsymbol{x}' - \boldsymbol{x}$
- Compute probability the given ray to be selected
  - $\mathrm{p}(\boldsymbol{\omega_i}) = \frac{|x - x'|^2}{4\pi r^2} p(l)$

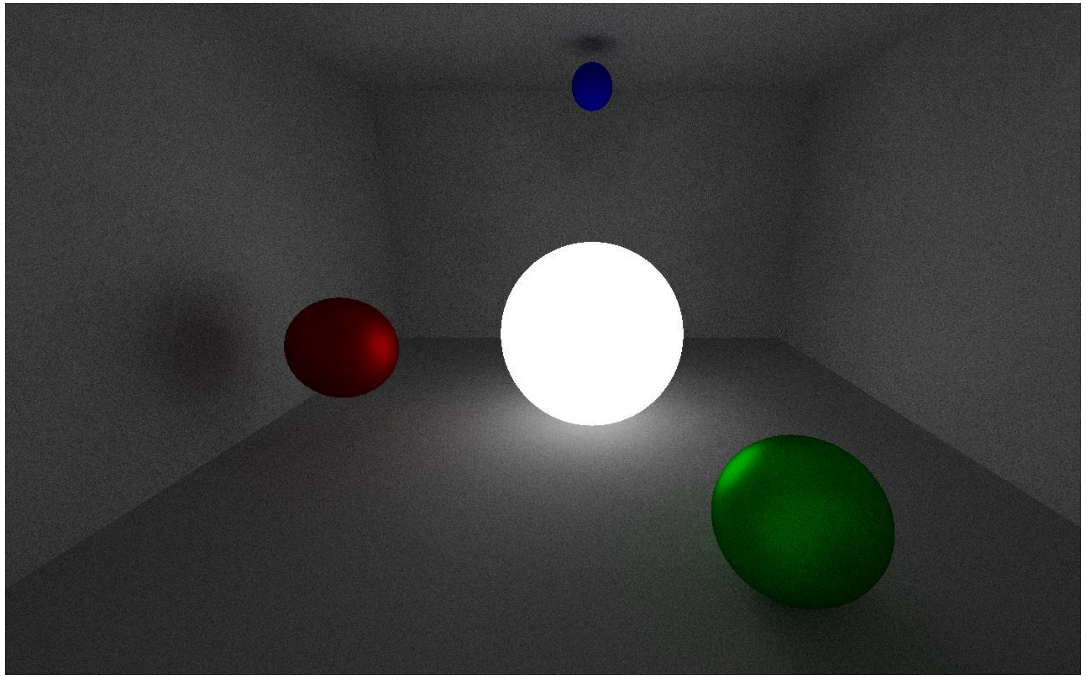
when lights are picked uniformly

$p(l) = \frac{1}{number\ of\ lights}$

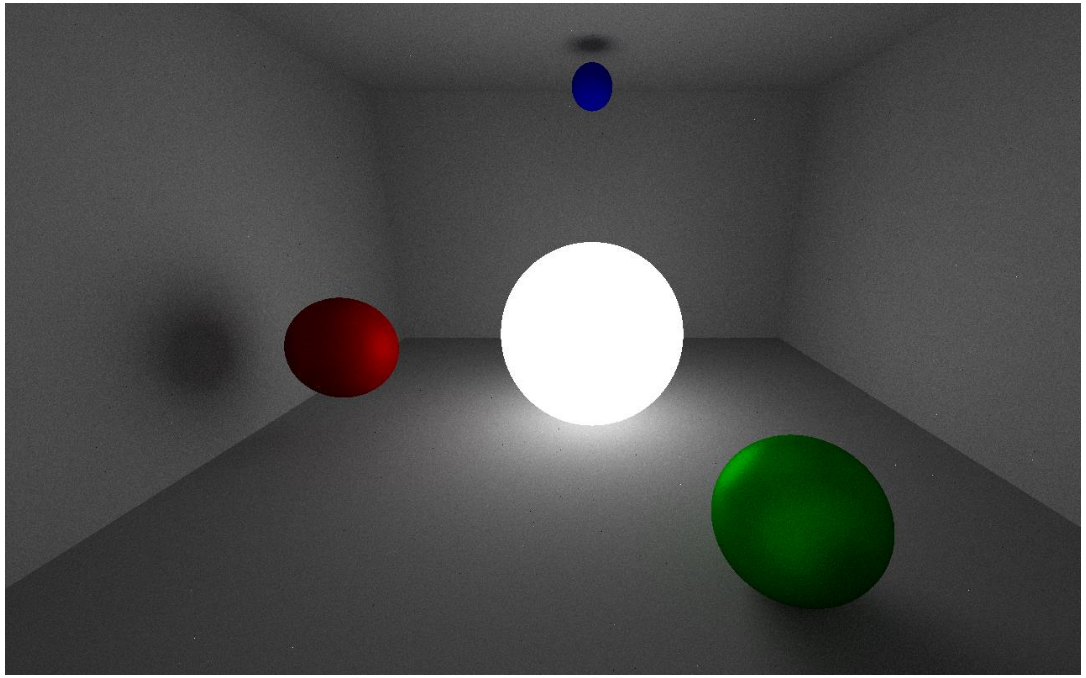


Comparison

GGX + Cosine , $n = 100, b = 10, g = ON$

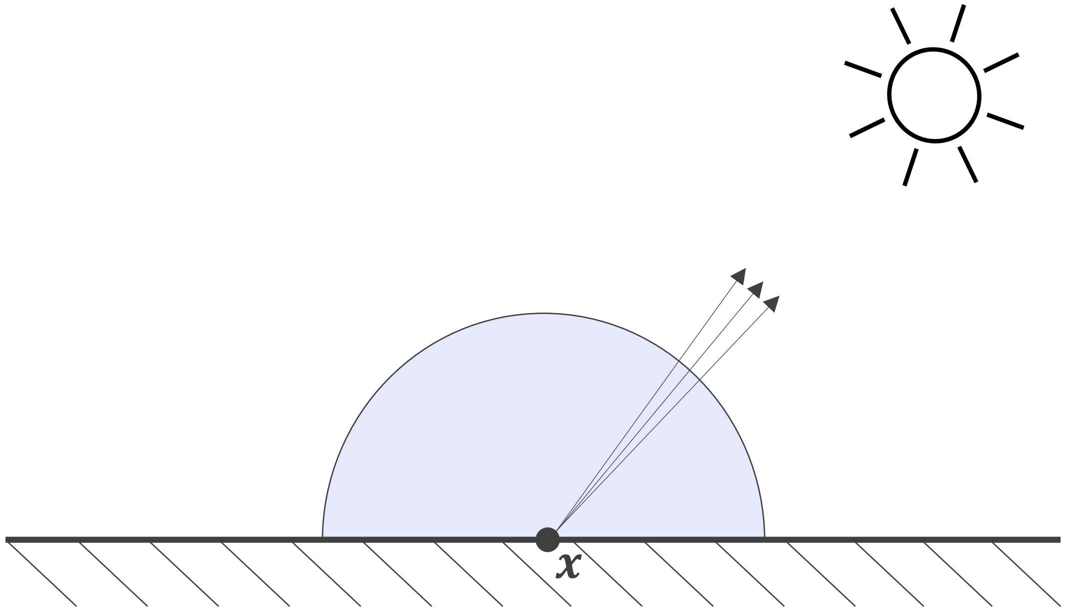


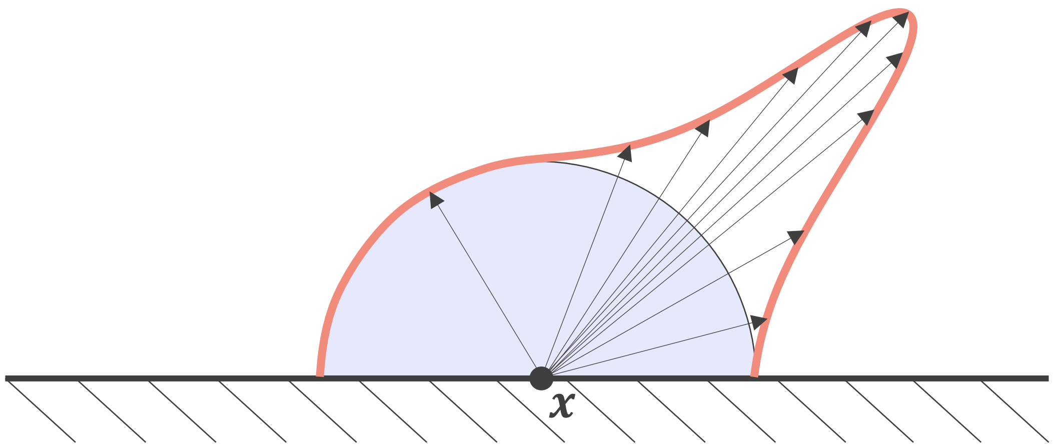
GGX+Cosine+Lights, $n = 100, b = 10, g = OFF$



Light Importance Sampling

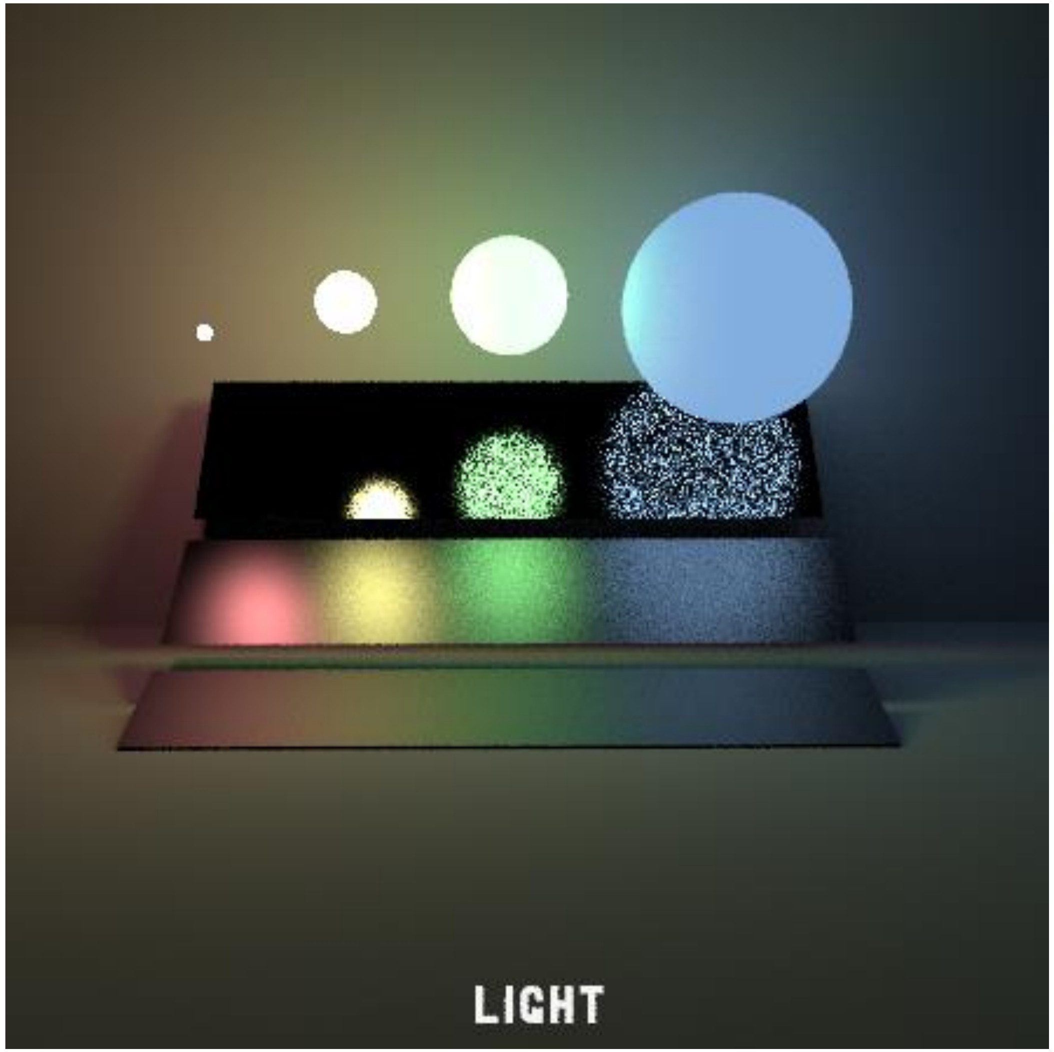
- In each step, we can sample both towards the light and BRDF spikes
  - Sample light to get direct illumination
  - Sample BRDF to get indirect illumination



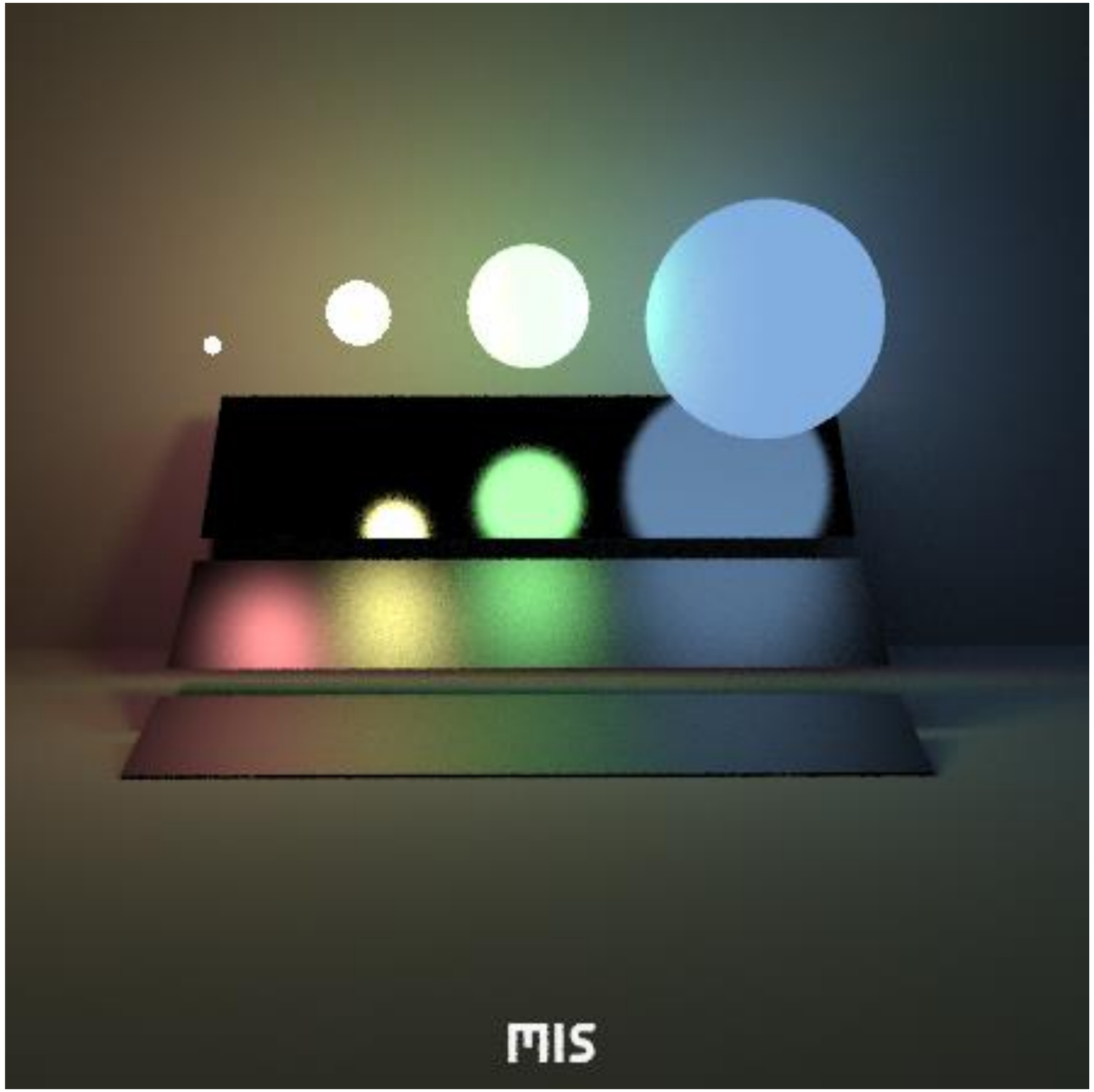


Example

https://www.shadertoy.com/view/4sSXWt



Combining BRDFs



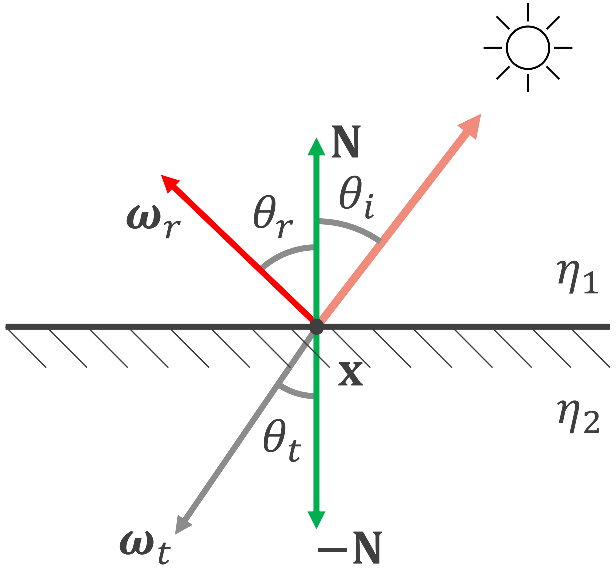
Snell's Law

$$\mathbf{N}_{ff} = \begin{cases} \mathbf{N} & when\ \mathbf{N} \cdot -\boldsymbol{\omega}_r \leq 0 \\ -\mathbf{N} & otherwise \end{cases}$$

$$\eta = \begin{cases} \frac{\eta 1}{\eta 2} & when\ N \equiv N_{ff} \\ \frac{\eta 2}{\eta 1} & otherwise \end{cases}$$

$$\boldsymbol{\omega}_t = -\eta \boldsymbol{\omega}_r + \left(\eta (\mathbf{N} \cdot \boldsymbol{\omega}_r) - \sqrt{1 - \eta^2 (1 - (\mathbf{N} \cdot \boldsymbol{\omega}_r)^2)} \right) \mathbf{N}$$

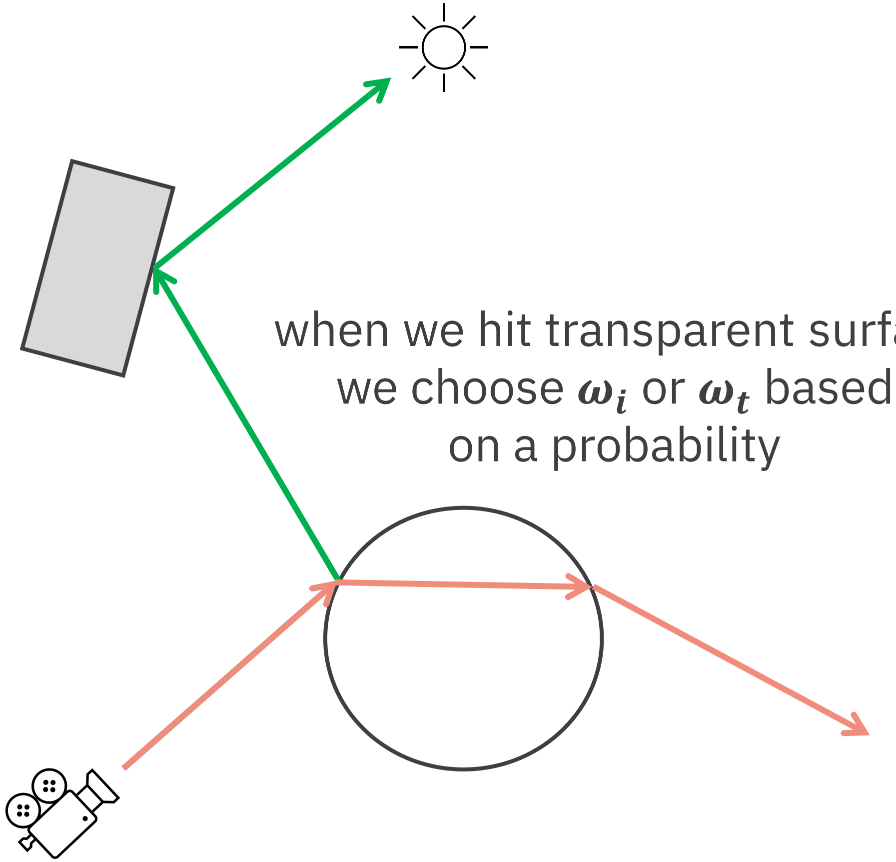
in GLSL you may use **refract($-\boldsymbol{\omega}_r$, $\boldsymbol{N}_{ff}$, $\boldsymbol{\eta}$)**



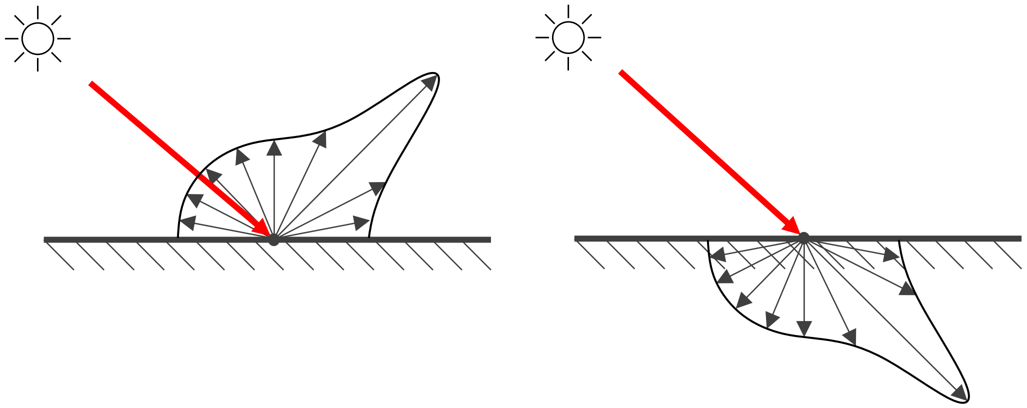
$$\frac{\sin \theta_t}{\sin \theta_i} = \frac{\eta_1}{\eta_2} = \eta \qquad \eta_1, \eta_2 \text{ – Indices of refraction}$$

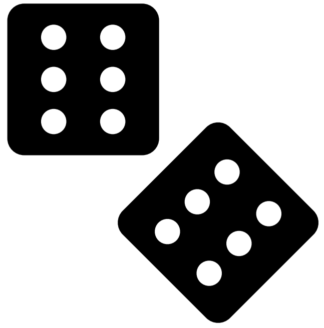
https://pixelandpoly.com/ior.html

Transmission



when we hit transparent surface
we choose $\boldsymbol{\omega_i}$ or $\boldsymbol{\omega_t}$ based
on a probability





Transmission

A bit hacky solution.

$$\boldsymbol{N}_{ff} = \begin{cases} \boldsymbol{N} & when\ \boldsymbol{N} \cdot -\boldsymbol{\omega}_r \leq 0 \\ -\boldsymbol{N} & otherwise \end{cases}$$

$$\eta = \begin{cases} \frac{\eta 1}{\eta 2} & when\ N \equiv N_{ff} \\ \frac{\eta 2}{\eta 1} & otherwise \end{cases}$$

```
if (K < 0.0 || prob > ξ){
    return SampleReflection(hit, ray)
} else {
    dir ← normalize(refract(−ωr , Nff , η))
    pdf ← 1.0
    return BSDFSample(dir , pdf)
}
```

$$F(R_0) = R_0 + (1 - R_0)(1 - (\boldsymbol{\omega}_r \cdot \boldsymbol{N_{ff}}))^5$$

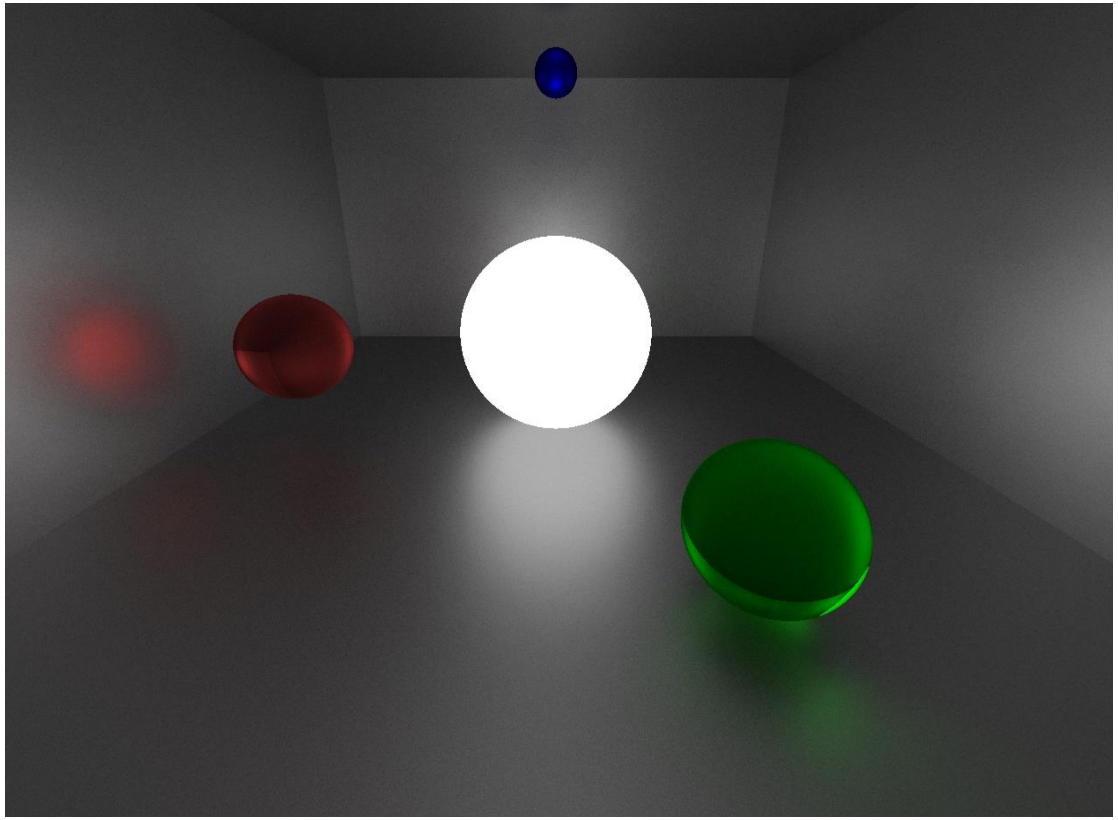
$$prob = F\left(\left(\frac{\eta_1 - \eta_2}{\eta_1 + \eta_2}\right)^2\right)$$

$$\mathrm{K} = 1.0 - \eta^2 * \left(1.0 - (-\boldsymbol{\omega}_r \cdot \boldsymbol{N}_{ff})^2\right)$$

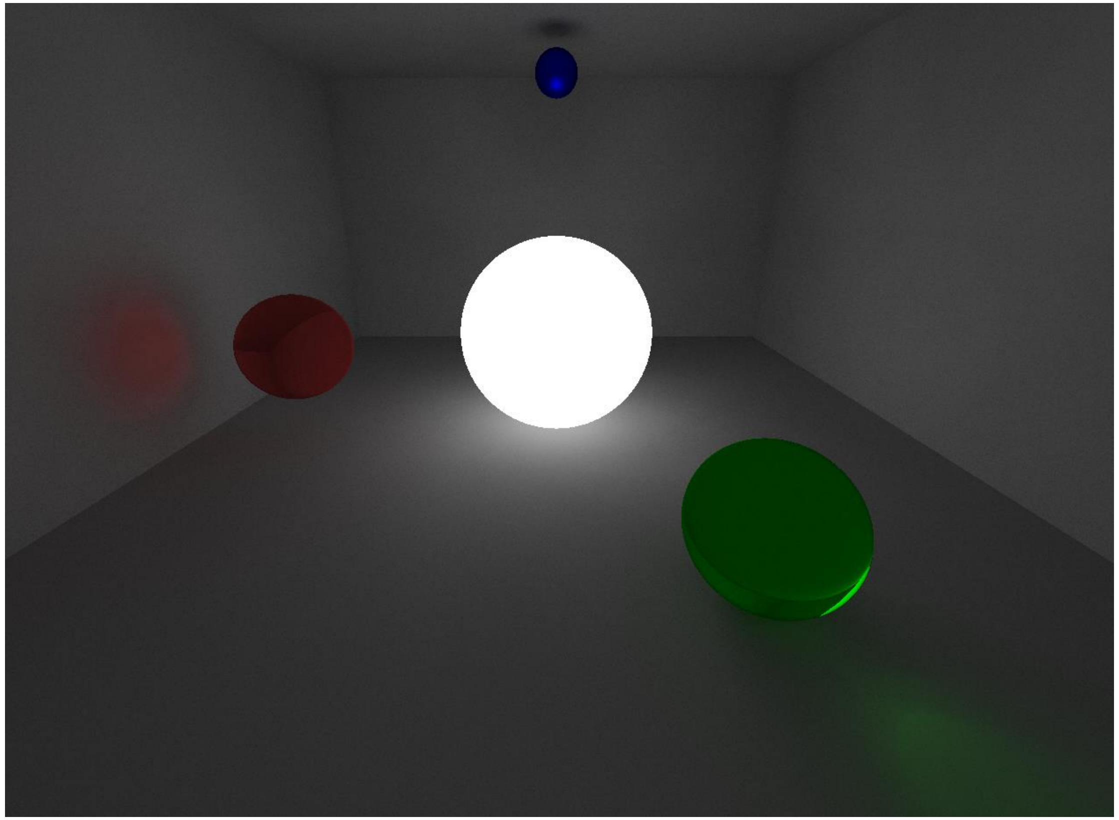
$$\xi = random\ number\ [0 - 1]$$

Result

GGX+Cosine, $n = 1000, r = 0.2\ b = 10, gama = ON$



GGX+Cosine, $n = 1000, b = 10, r = 0.8, gama = ON$



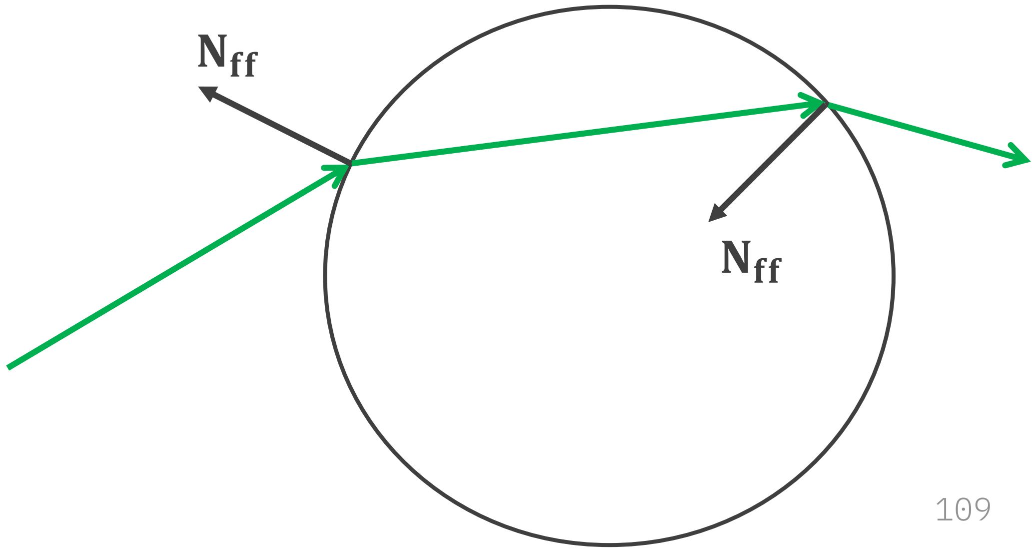
Problems with Transmission

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$

- When computing new ray, we need to correctly offset the origin

```
ray = Ray(hit.intersection + (hit.glass ? -EPSILON * hit.ffnormal : EPSILON * hit.ffnormal), sample.direction)
```

- Rendering equation is not really build for BTDF
  - $\cos\theta_i = N \cdot \boldsymbol{\omega}_i$ will be always negative
  - you may invert $\boldsymbol{\omega}_i$ or better ignore it completely



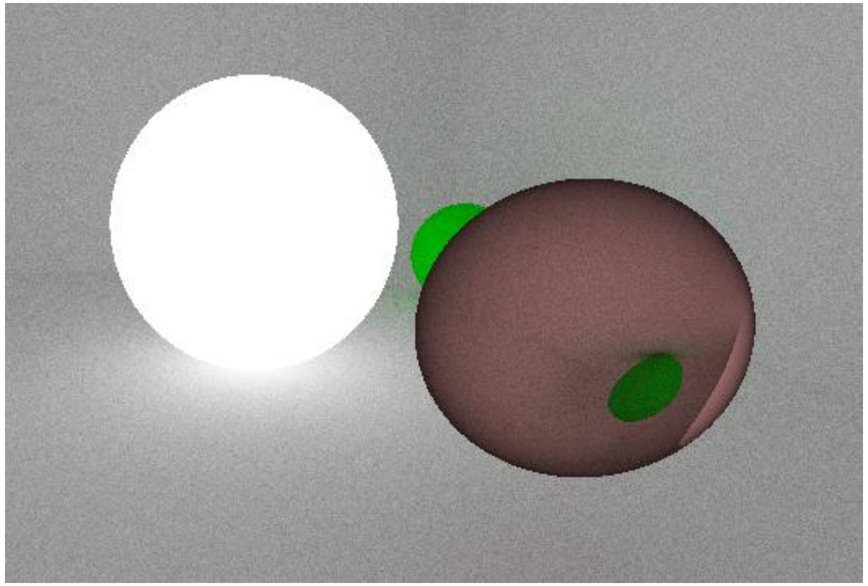
Problems with Transmission

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}_r) = L_e(\boldsymbol{x}, \boldsymbol{\omega}_r) + \frac{1}{n}\sum_{i=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) \cos\theta_i \frac{1}{p(\boldsymbol{\omega}_i)}$$

- When computing new ray, we need to correctly offset the origin

```
ray = Ray(hit.intersection + (hit.glass ? -EPSILON * hit.ffnormal : EPSILON * hit.ffnormal), sample.direction)
```

- Rendering equation is not really designed for BTDF
  - $\cos\theta_i = N \cdot \boldsymbol{\omega}_i$ will be always negative
  - Solution:
    - you may invert $\boldsymbol{\omega}_i$ (produces dark edges)
    - better ignore it completely



More Reading

www.pbr-book.org/3ed-2018/contents

docs.unrealengine.com