Advanced 3D graphics for movies and games (NPGR010)

– Monte Carlo integration & Direct illumination

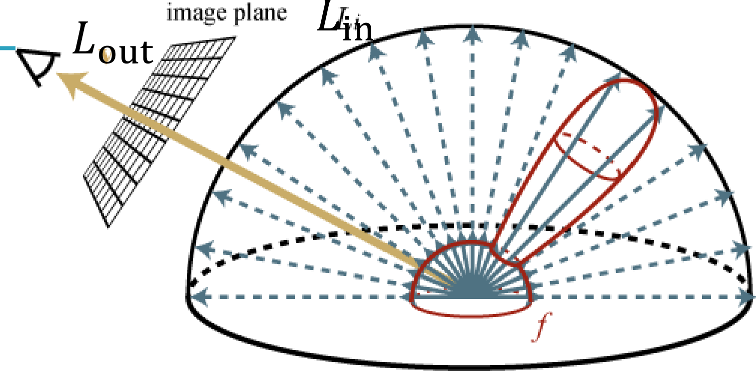
Jiří Vorba, MFF UK/Weta Digital

jirka@cgg.mff.cuni.cz

Slides of prof. Jaroslav Křivánek, minor edits by Jiří Vorba

Entire the lecture in 5 slides

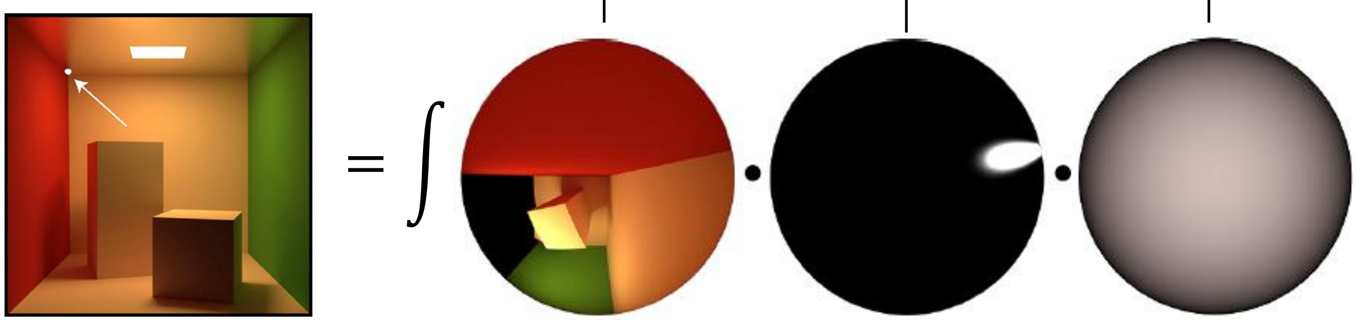
Reflection equation



- Total reflected radiance: integrate contributions of incident radiance, weighted by the BRDF, over the hemisphere

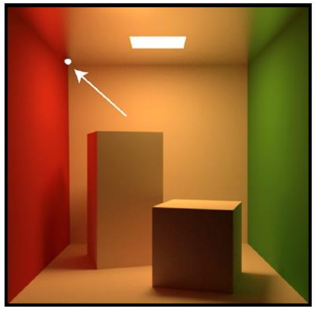
$$\boxed{L_{\text{out}}(\omega_{\text{out}}) = \int_{H(\mathbf{x})} L_{\text{in}}(\omega_{\text{in}}) \cdot f_r(\omega_{\text{in}} \to \omega_{\text{out}}) \cdot \cos\theta_{\text{in}} \; \mathrm{d}\omega_{\text{in}}}$$

upper hemisphere over x



Rendering = Integration of functions

$$\boxed{L_{\text{out}}(\omega_{\text{out}}) = \int_{H(\mathbf{x})} L_{\text{in}}(\omega_{\text{in}}) \cdot f_r(\omega_{\text{in}} \to \omega_{\text{out}}) \cdot \cos\theta_{\text{in}} \;\, \mathrm{d}\omega_{\text{in}}}$$



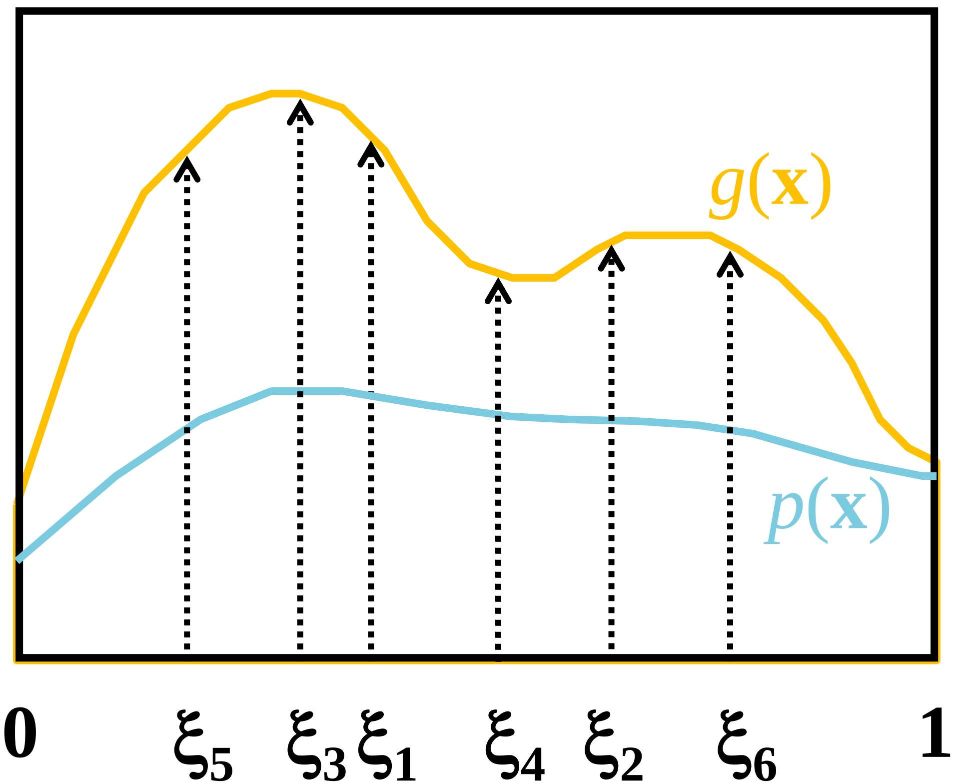
Incoming radiance $L_{\text{in}}(\mathbf{x},\omega_{\text{in}})$ for a point on the ceiling.

- Problems
  - Discontinuous integrand (visibility)
  - Arbitrarily large integrand values (e.g. light distribution in caustics, BRDFs of glossy surfaces)
  - Complex geometry

Images: Greg Ward

Monte Carlo integration

- General tool for estimating definite integrals



Integral:

$$I = \int g(\mathbf{x}) \mathrm{d}\mathbf{x}$$

Monte Carlo estimate I:

$$\langle I \rangle = \frac{1}{N} \sum_{k=1}^{N} \frac{g(\xi_k)}{p(\xi_k)}; \quad \xi_k \propto p(\mathbf{x})$$

Works "on average":

$$E[\langle I \rangle] = I$$

Application of MC to reflection eq: Estimator of reflected radiance

- Integral to be estimated:

$$\int_{H(\mathbf{x})} \underbrace{L_{\text{in}}(\omega_{\text{in}})\, f_r(\omega_{\text{in}} \to \omega_{\text{out}}) \cos\theta_{\text{in}}}_{\text{integrand}(\omega_{\text{in}})}\, \mathrm{d}\omega_{\text{in}}$$

- pdf for cosine-proportional sampling:

$$p(\omega_{\text{in}}) = \frac{\cos\theta_{\text{in}}}{\pi}$$

- **MC estimator** (formula to use in the renderer):

$$\begin{aligned} \hat{L}_{\text{out}} &= \frac{1}{N}\sum_{k=1}^{N} \frac{\text{integrand}(\omega_{\text{in},k})}{\text{pdf}(\omega_{\text{in},k})} \\ &= \frac{\pi}{N}\sum_{k=1}^{N} L_{\text{in}}(\omega_{\text{in},k})\, f_r(\omega_{\text{in},k} \to \omega_{\text{out}}) \end{aligned}$$

Estimator of reflected radiance: Implementation

```
// input variables
x...shaded point on a surface
normal...surface normal at x
omegaOut...viewing (camera) direction

estimatedRadianceOut := Rgb(0,0,0);
for k = 1...N
        [omegaInK, pdf] := generateRndDirection();
        // evaluate integrand
        radianceInEst := getRadianceIn(x, omegaInK);
        brdf := evalBrdf(x, omegaInK, omegaOut);
        cosThetaIn := dot(normal, omegaInK);
        integrand := radianceInEst * brdf * cosThetaIn;
        // evaluate contribution to the outgoing radiance
        estimatedRadianceOut += integrand / pdf;
end for
estimatedRadianceOut /= N;
```

Variance => image noise

… and now the slow way

Digression: Numerical quadrature

Quadrature formulas for numerical integration

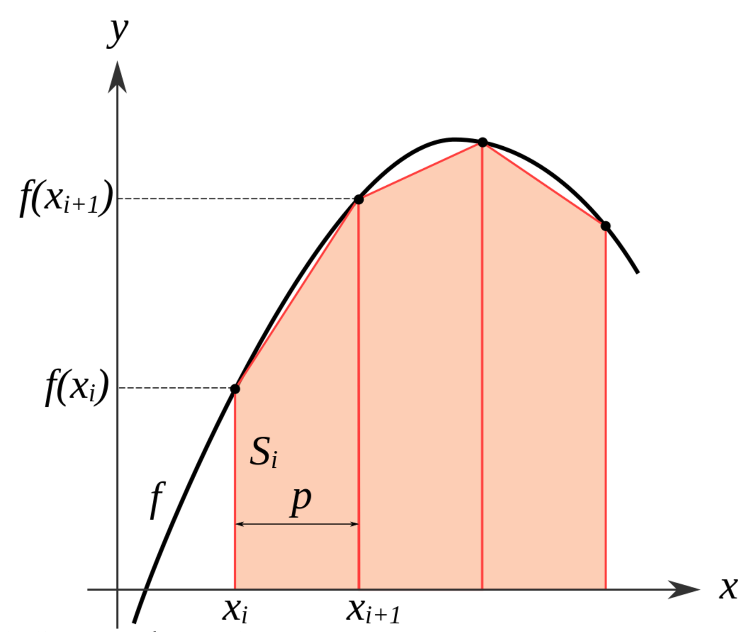
- General formula in 1D:

$$\hat{I} = \sum_{k=1}^{N} w_k g(x_k)$$

g integrand (i.e. the integrated function)
N quadrature order (i.e. number of integrand samples)
x_k node points (i.e. positions of the samples)
$g(\mathrm{x}_k)$ integrand values at node points
w_k quadrature weights

Quadrature formulas for numerical integration

- Quadrature rules differ by the choice of node point positions x_k and the weights w_k
  - E.g. rectangle rule, trapezoidal rule, Simpson's method, Gauss quadrature, …
- The samples (i.e. the node points) are placed deterministically



Source: Wikimedia Commons

Quadrature formulas in multiple dimensions

- General formula for quadrature of a function of multiple variables:

$$\hat{I} = \sum_{k_1=1}^{N} \sum_{k_2=1}^{N} \dots \sum_{k_d=1}^{N} w_{k_1} w_{k_2} \dots w_{k_s} f(x_{k_1}, x_{k_2}, \dots, x_{k_d})$$

- Convergence speed of approximation error E for a d-dimensional integral is $E = O(N^{-1/d})$
  - E.g. in order to cut the error in half for a 3-dimensional integral, we need $2^3 = 8$ times more samples

- Unusable in higher dimensions
  - **Dimensional explosion**

Quadrature formulas in multiple dimensions

- **Deterministic quadrature vs. Monte Carlo**
  - In 1D deterministic better than Monte Carlo
  - In 2D roughly equivalent
  - From 3D, MC will always perform better
- Remember, quadrature rules are NOT the Monte Carlo method

Monte Carlo

History of the Monte Carlo method

- Atomic bomb development, Los Alamos 1940
  John von Neumann, Stanislav Ulam, Nicholas Metropolis

- Further development and practical applications from the early 50's

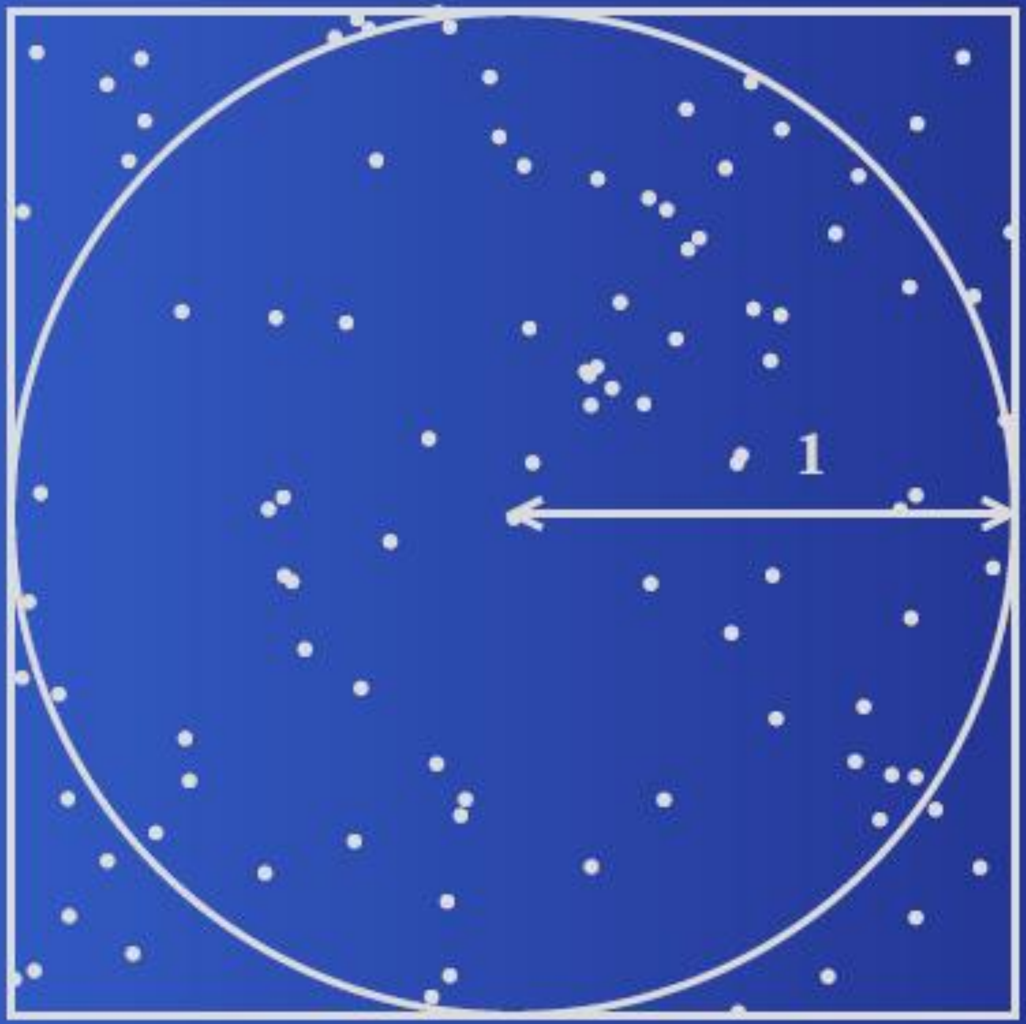
Monte Carlo method

- We simulate many random occurrences of the same type of events, e.g.:
  - Neutrons – emission, absorption, collisions with hydrogen nuclei
  - Behavior of computer networks, traffic simulation.
  - Sociological and economical models – demography, inflation, insurance, etc.

Monte Carlo – applications

- Financial market simulations
- Traffic flow simulations
- Environmental sciences
- Particle physics
- Quantum field theory
- Astrophysics
- Molecular modeling
- Semiconductor devices
- Optimization problems
- **Light transport calculations**
- ...

Example: calculation of π



Area of square: $A_s = 4$
Area of circle: $A_c = \pi$
Fraction p of random points inside circle:

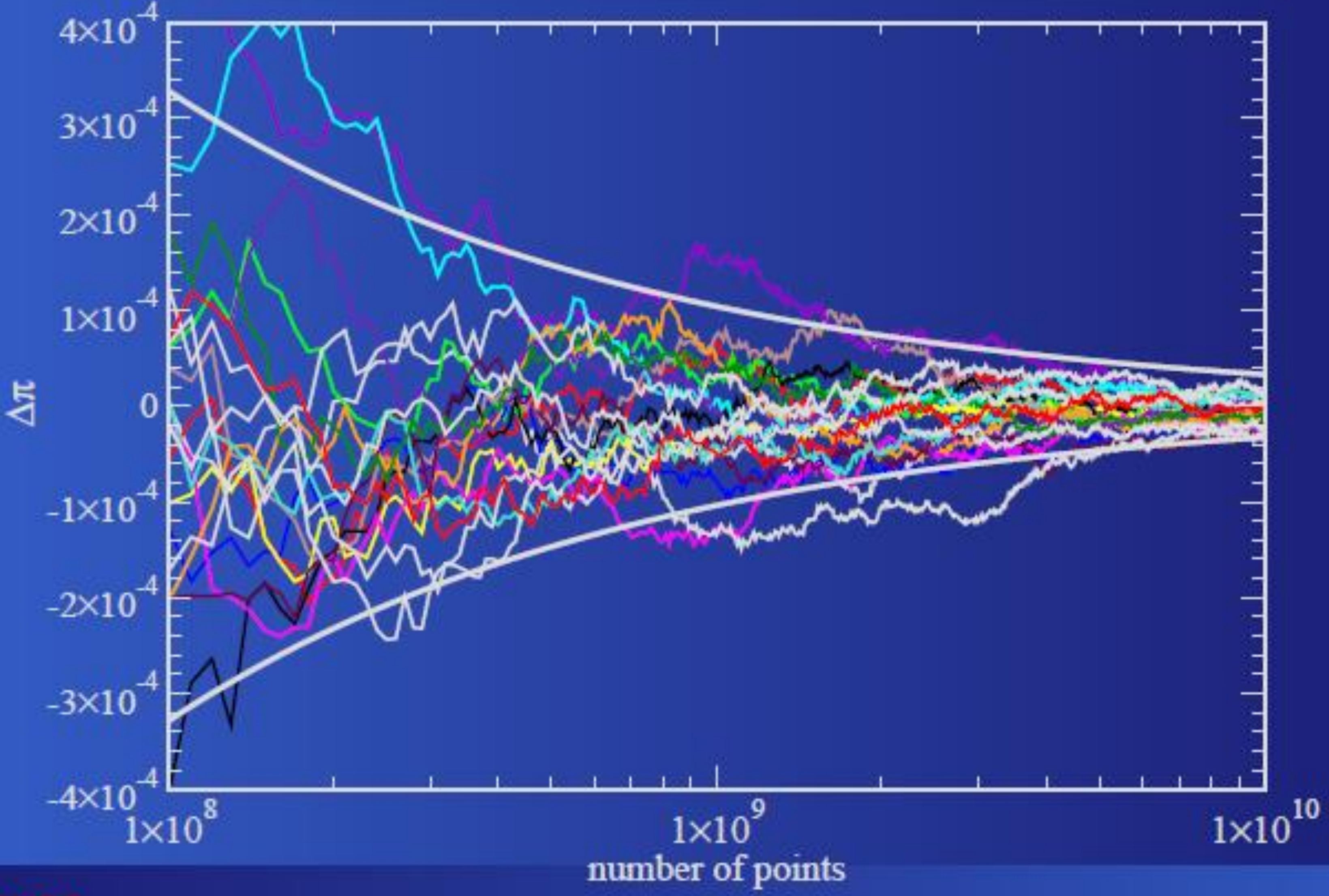
$$p = \frac{A_c}{A_s} = \frac{\pi}{4}$$

Random points: N
Random points inside circle: N_c

$$\Rightarrow \quad \pi = \frac{4N_c}{N}$$

Slide credit: Iwan Kawrakov

Calculation of π (cont'd)



Variance => image noise

Monte Carlo integration

- Samples are placed randomly (or pseudo-randomly)

- Convergence of standard error: std. dev. = $O(N^{-1/2})$
  - **Convergence speed independent of dimension**
  - **Faster than classic quadrature rules** for 3 and more dimensions

- Special methods for placing samples exist
  - Quasi-Monte Carlo
  - Faster asymptotic convergence than MC for "smooth" functions

Monte Carlo integration

- **Pros**
  - Simple implementation
  - Robust solution for complex integrands and integration domains
  - Effective for high-dimensional integrals

- **Cons**
  - Relatively slow convergence – halving the standard error requires four times as many samples
  - In rendering: images contain noise that disappears slowly

Review – Random variables

Random variable

- X … random variable
- X assumes different values with different probability
  - Given by the probability distribution D
  - $X \sim D$

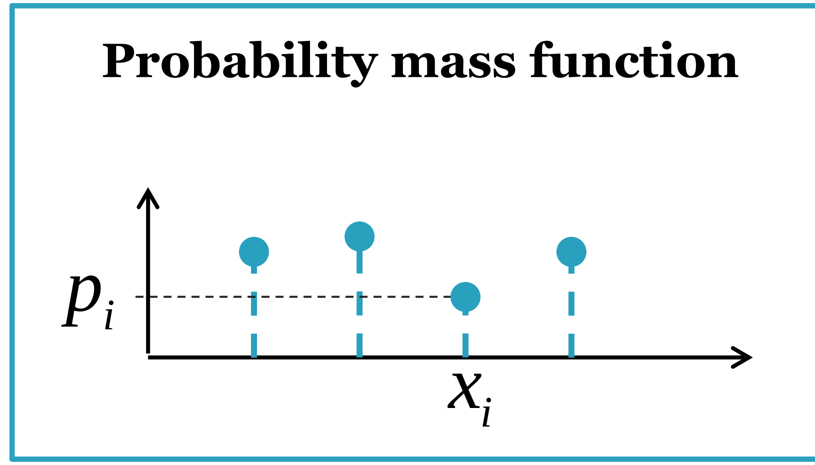
Discrete random variable

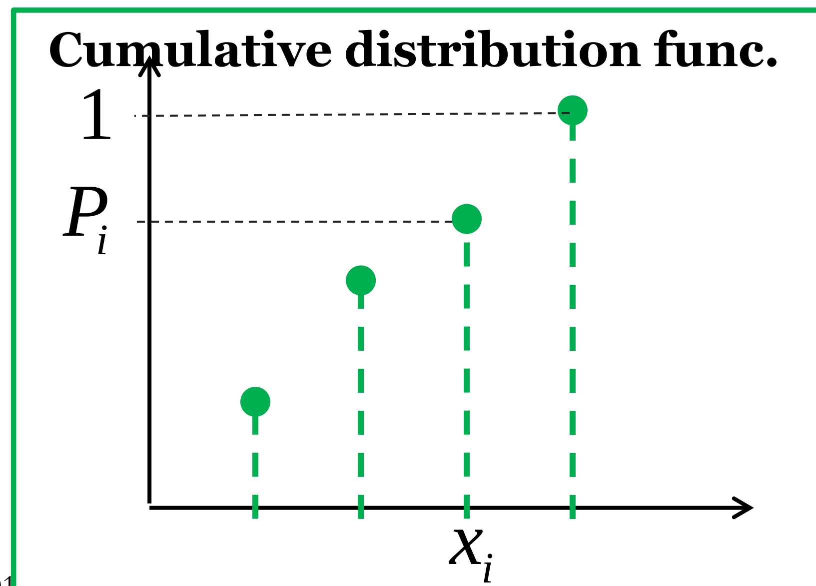
- Finite set of values of x_i
- Each assumed with prob. p_i

$$p_i \equiv \Pr(X = x_i) \geq 0 \qquad \sum_{i=1}^{n} p_i = 1$$

- **Cumulative distribution function**

$$P_i \equiv \Pr\left(X \leq x_i\right) = \sum_{j=1}^{i} p_j \qquad P_n = 1$$





Continuous random variable

- Probability density function, **pdf,** $p(x)$

$$\boxed{\Pr\left(X \in D\right) = \int_D p(x)\, \mathrm{d}x}$$

- In 1D:

$$\Pr\left(a < X \le b\right) = \int_a^b p(t)\, \mathrm{d}t$$

Continuous random variable

- Cumulative distribution function, **cdf,** $P(x)$

  In 1D:

$$P(x) \equiv \Pr\left(X \leq x\right) = \int_{-\infty}^{x} p(t)\,\mathrm{d}t$$

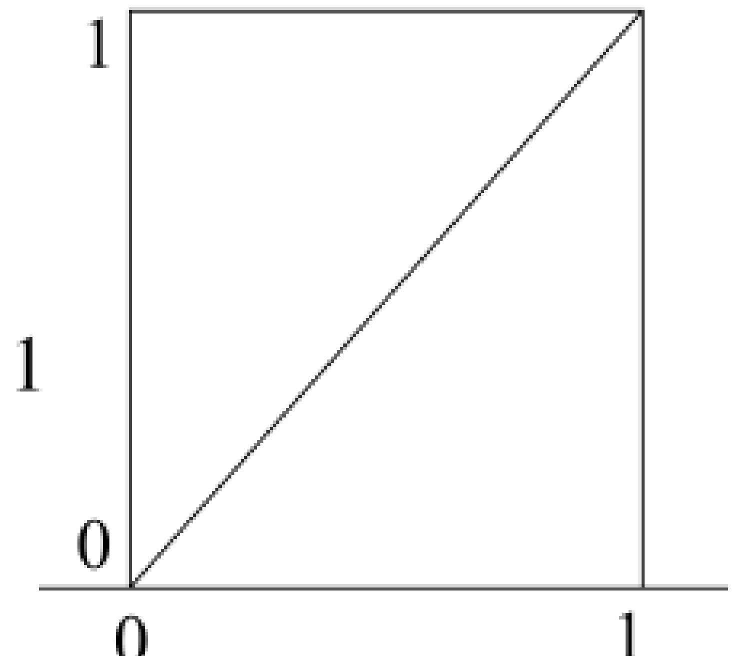
$$\boxed{\Pr\left(X = a\right) = \int_{a}^{a} p(t)\,\mathrm{d}t = 0\,!}$$

Continuous random variable

Example: **Uniform distribution**

Probability density function **(pdf)**

Cumulative distribution function **(cdf)**



Continuous random variable

Gaussian (normal) distribution

Probability density function **(pdf)**

Cumulative distribution function **(cdf)**

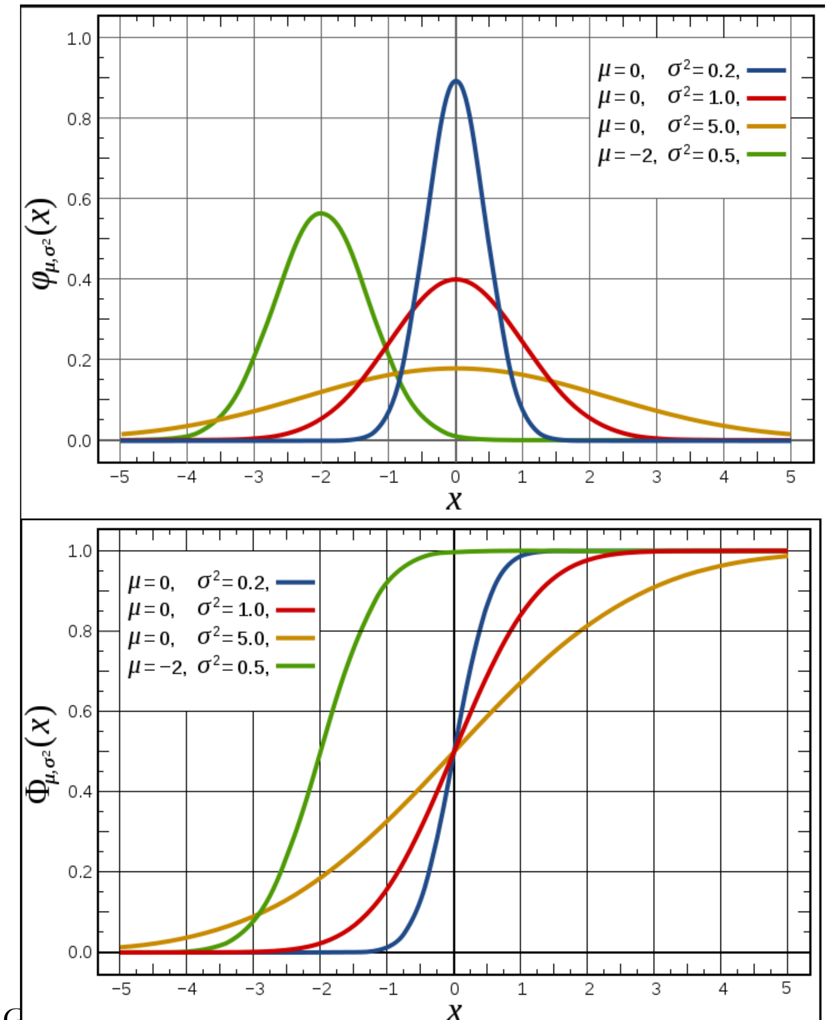


Image: wikipedia

Expected value and variance

- **Expected value**

$$E[X] = \int_D \mathbf{x}\, p(\mathbf{x})\, \mathrm{d}\mathbf{x}$$

- **Variance**

$$\begin{aligned} V[X] &= E[(X - E[X])^2] \\ &= E[X^2 - 2XE[X] + E[X]^2] \\ &= E[X^2] - E[X]^2 \end{aligned}$$

  - Properties of variance

$$V[\sum_i X_i] = \sum_i V[X_i] \qquad \text{(if } X_i \text{ are independent)}$$

$$V[aX] = a^2 V[X]$$

Transformation of a random variable

$$Y = g(X)$$

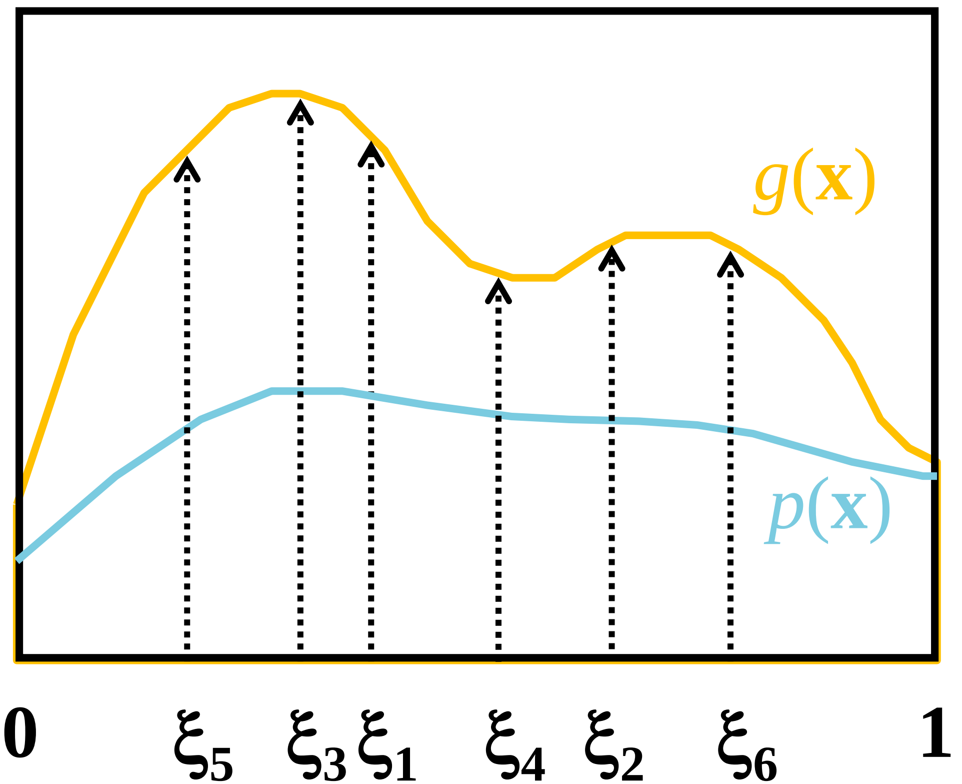
- Y is a random variable
- Expected value of Y

$$\boxed{E[Y] = \int_D g(\mathbf{x})\, p(\mathbf{x})\, d\mathbf{x}}$$

Monte Carlo integration

Monte Carlo integration

- General tool for estimating definite integrals



Integral:

$$I = \int g(\mathbf{x})\mathrm{d}\mathbf{x}$$

Monte Carlo estimate *I*:

$$\langle I \rangle = \frac{1}{N}\sum_{k=1}^{N}\frac{g(\xi_k)}{p(\xi_k)}; \quad \xi_k \propto p(\mathbf{x})$$

Works “on average”:

$$E[\langle I \rangle] = I$$

Primary estimator of an integral

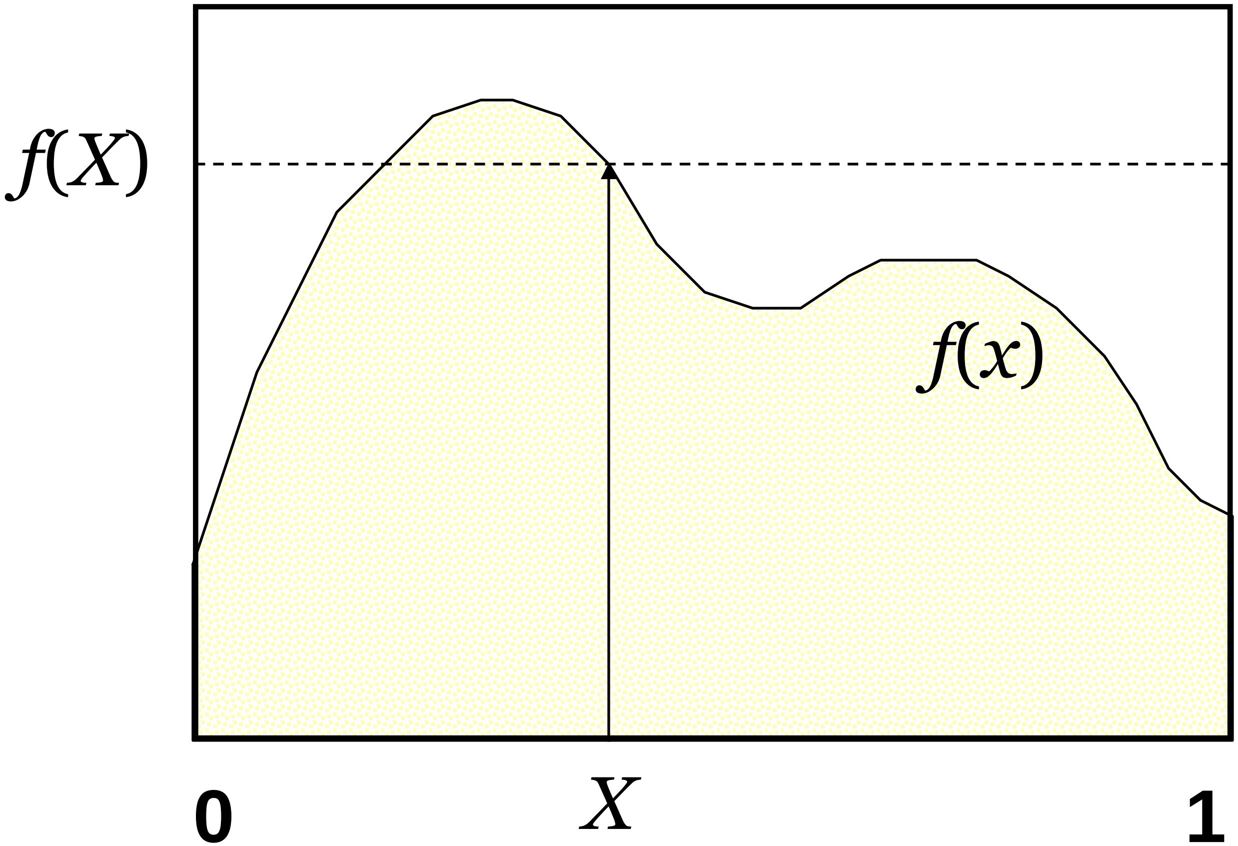
Integral to be estimated:

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

Let X be a random variable from the distribution with the pdf $p(x)$, then the random variable F_{prim} given by the transformation $f(X)/p(X)$ is called the **primary estimator** of the above integral.

$$\boxed{F_{\text{prim}} = \frac{f(X)}{p(X)}}$$

Primary estimator of an integral



Estimator vs. estimate

- **Estimator is a random variable**
  - It is defined though a transformation of another random variable
- **Estimate** is a concrete realization (outcome) of the estimator
- No need to worry: the above distinction is important for proving theorems but less important in practice

Unbiased estimator

- A general statistical estimator is called **unbiased** if – “on average” – it yields the correct value of an estimated quantity Q (without systematic error).
- More precisely:

$$\boxed{E[F] = Q}$$

Estimator of the quantity Q
(random variable)

Estimated quantity
(In our case, it is an integral, but in general it could be anything. It is a number, not a random variable.)

Unbiased estimator

The primary estimator F_{prim} is an unbiased estimator of the integral I.

Proof:

$$
\begin{aligned}
E\left[F_{\text{prim}}\right] &= \int_{\Omega} \frac{f(x)}{p(x)}\, p(x)\, \mathrm{d}x \\
&= I
\end{aligned}
$$

Variance of the primary estimator

For an unbiased estimator, the error is due to **variance**:

$$V\left[F_{\text{prim}}\right] = \sigma^2_{\text{prim}} = E[F_{\text{prim}}{}^2] - E[F_{\text{prim}}]^2 = \int_{\Omega} \frac{f(x)^2}{p(x)} \, \mathrm{d}x - I^2$$

(for an unbiased estimator)

If we use only a single sample, the variance is usually too high.
Depends on p(x).
We need more samples in practice => secondary estimator.

Secondary estimator of an integral

- Consider N independent random variables X_k
- The estimator F_N given be the formula below is called the **secondary estimator** of I.

$$F_N = \frac{1}{N}\sum_{k=1}^{N}\frac{f(X_k)}{p(X_k)}$$

- The secondary estimator is unbiased.

Variance of the secondary estimator

$$
\begin{aligned}
V[F_N] &= V\left[\frac{1}{N}\sum_{k=1}^{N}\frac{f(X_k)}{p(X_k)}\right] \\
&= \frac{1}{N^2}\cdot N\cdot V\left[\frac{f(X_k)}{p(X_k)}\right] \\
&= \frac{1}{N}V\left[F_{\text{prim}}\right]
\end{aligned}
$$

... standard deviation is **√*N*-times smaller**!

(i.e. error converges with **1/√*N***)

Properties of estimators

Unbiased estimator

- A general statistical estimator is called **unbiased** if – "on average" – it yields the correct value of an estimated quantity Q (without systematic error).
- More precisely:

$$\boxed{E[F] = Q}$$

Estimator of the quantity Q
(random variable)

Estimated quantity
(In our case, it is an integral, but in general it could be anything. It is a number, not a random variable.)

Bias of a biased estimator

- If

$$E[F] \neq Q$$

then the estimator is "**biased**" (cz: vychýlený).

- **Bias** is the systematic error of the estimator:

$$\beta = Q - E[F]$$

Consistency

- Consider a secondary estimator with N samples:

$$F_N = F_N(X_1, X_2, \ldots, X_N)$$

- Estimator F_N is **consistent** if

$$\boxed{Pr\left\{\lim_{N\to\infty} F_N = Q\right\} = 1}$$

i.e. if the error $F_N - Q$ converges to zero with probability 1.

Consistency

- Sufficient condition for consistency of an estimator:

$$\lim_{N\to\infty} \underbrace{\beta[F_N]}_{\textbf{bias}} = \lim_{N\to\infty} V[F_N] = 0$$

- Unbiasedness is not sufficient for consistency by itself (if the variance is infinite).
- But if the variance of a primary estimator finite, then the corresponding secondary estimator is necessarily consistent.

Rendering algorithms

- **Unbiased**
  - Path tracing
  - Bidirectional path tracing
  - Metropolis light transport
- **Biased & Consistent**
  - Progressive photon mapping
- **Biased & not consistent**
  - Photon mapping
  - Irradiance / radiance caching

Mean Squared Error – MSE
(cz: Střední kvadratická chyba)

- **Definition**

$$\boxed{MSE[F] = E[(F-Q)^2]}$$

- **Proposition**

$$MSE[F] = V[F] + \beta[F]^2$$

  - *Proof*

$$\begin{aligned} MSE[F] &= E[(F-Q)^2] \\ &= E[(F-E[F])^2] + 2E[F-E[F]](E[F]-Q) + (E[F]-Q)^2 \\ &= V[F] + \beta[F]^2\,, \end{aligned}$$

Mean Squared Error – MSE
(cz: Střední kvadratická chyba)

- If the estimator F is unbiased, then

$$MSE[F] = V[F]$$

  i.e. for an unbiased estimator, it is much easier to estimate the error, because it can be estimated directly from the samples $Y_k = f(X_k) / p(X_k)$.
- Unbiased **estimator of variance**

$$\hat{V}[F_N] = \frac{1}{N-1}\left\{\left(\frac{1}{N}\sum_{k=1}^{N} Y_k^2\right) - \left(\frac{1}{N}\sum_{k=1}^{N} Y_k\right)^2\right\}$$

Why divide by N-1 instead of N?

- To get an unbiased estimate
- https://en.wikipedia.org/wiki/Bessel%27s_correction

Root Mean Squared Error – RMSE

$$RMSE[F] = \sqrt{MSE[F]}$$

Efficiency of an estimator

- **Efficiency** of an unbiased estimator is given by:

$$\boxed{\epsilon[F] = \frac{1}{V[F]\,T[F]}}$$

variance

Calculation time
(i.e. operations count, such as number of cast rays)

MC estimators for illumination calculation

Estimator of reflected radiance (1)

- Integral to be estimated:

$$\int_{H(\mathbf{x})} \underbrace{L_{\text{in}}(\omega_{\text{in}})\, f_r(\omega_{\text{in}} \to \omega_{\text{out}}) \cos\theta_{\text{in}}}_{\text{integrand}(\omega_{\text{in}})}\, \mathrm{d}\omega_{\text{in}}$$

- pdf for uniform hemisphere sampling:

$$p(\omega_{\text{in}}) = \frac{1}{2\pi}$$

- **MC estimator** (formula to use in the renderer):

$$\begin{aligned} \hat{L}_{\text{out}} &= \frac{1}{N} \sum_{k=1}^{N} \frac{\text{integrand}(\omega_{\text{in},k})}{\text{pdf}(\omega_{\text{in},k})} \\ &= \frac{2\pi}{N} \sum_{k=1}^{N} L_{\text{in}}(\omega_{\text{in},k})\, f_r(\omega_{\text{in},k} \to \omega_{\text{out}}) \cos\theta_{\text{in},k} \end{aligned}$$

Application of MC to reflection eq: Estimator of reflected radiance

- Integral to be estimated:

$$\int_{H(\mathbf{x})} \underbrace{L_{\text{in}}(\omega_{\text{in}})\, f_r(\omega_{\text{in}} \to \omega_{\text{out}}) \cos\theta_{\text{in}}}_{\text{integrand}(\omega_{\text{in}})} \, \mathrm{d}\omega_{\text{in}}$$

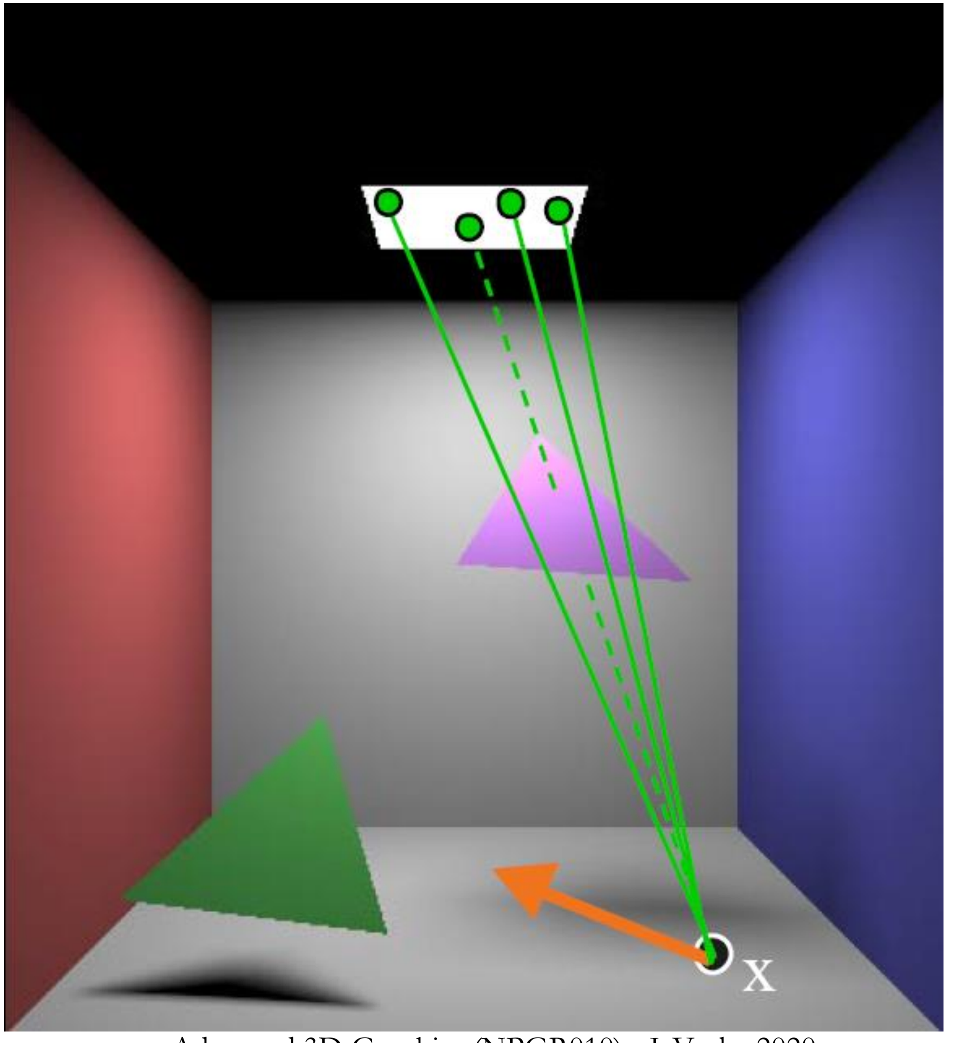
- pdf for cosine-proportional sampling:

$$p(\omega_{\text{in}}) = \frac{\cos\theta_{\text{in}}}{\pi}$$

- **MC estimator** (formula to use in the renderer):

$$\begin{aligned} \hat{L}_{\text{out}} &= \frac{1}{N} \sum_{k=1}^{N} \frac{\text{integrand}(\omega_{\text{in},k})}{\text{pdf}(\omega_{\text{in},k})} \\ &= \frac{\pi}{N} \sum_{k=1}^{N} L_{\text{in}}(\omega_{\text{in},k})\, f_r(\omega_{\text{in},k} \to \omega_{\text{out}}) \end{aligned}$$

Irradiance estimate – light source sampling



Irradiance estimate – light source sampling

- Reformulate the reflection integral (change of variables)

$$E(\mathbf{x}) = \int_{H(\mathbf{x})} L_\mathrm{i}(\mathbf{x}, \omega_\mathrm{i}) \cdot \cos\theta_\mathrm{i} \, \mathrm{d}\omega_\mathrm{i}$$

$$= \int_A L_\mathrm{e}(\mathbf{y} \to \mathbf{x}) \cdot V(\mathbf{y} \leftrightarrow \mathbf{x}) \cdot \underbrace{\frac{\cos\theta_\mathbf{y} \cdot \cos\theta_\mathbf{x}}{\|\mathbf{y} - \mathbf{x}\|^2}}_{G(\mathbf{y} \leftrightarrow \mathbf{x})} \, \mathrm{d}A$$

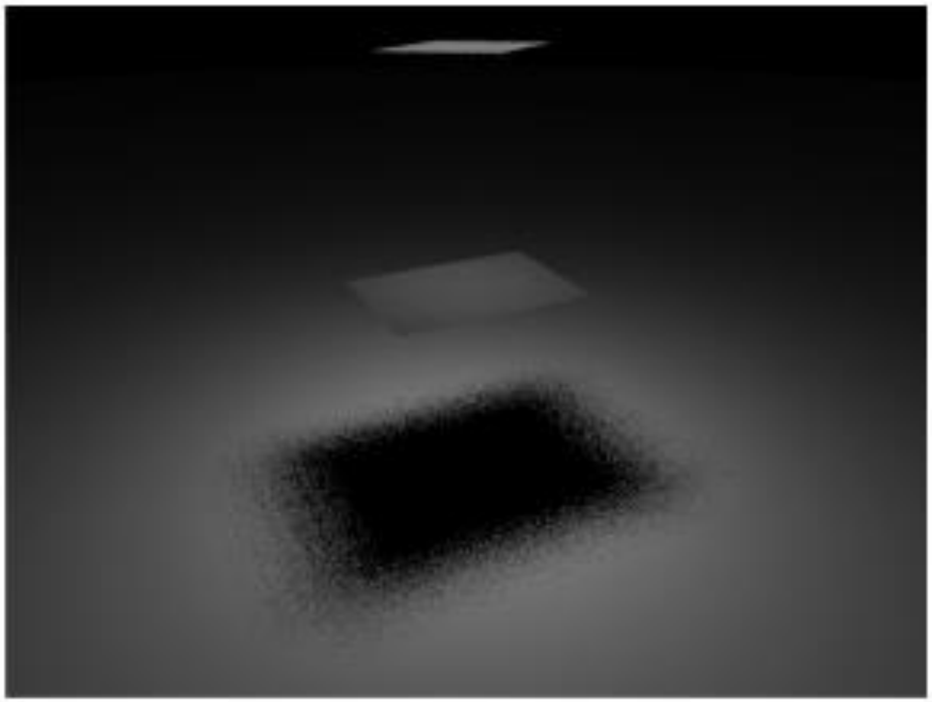
- PDF for uniform sampling of the surface area:

$$p(\mathbf{y}) = \frac{1}{|A|}$$

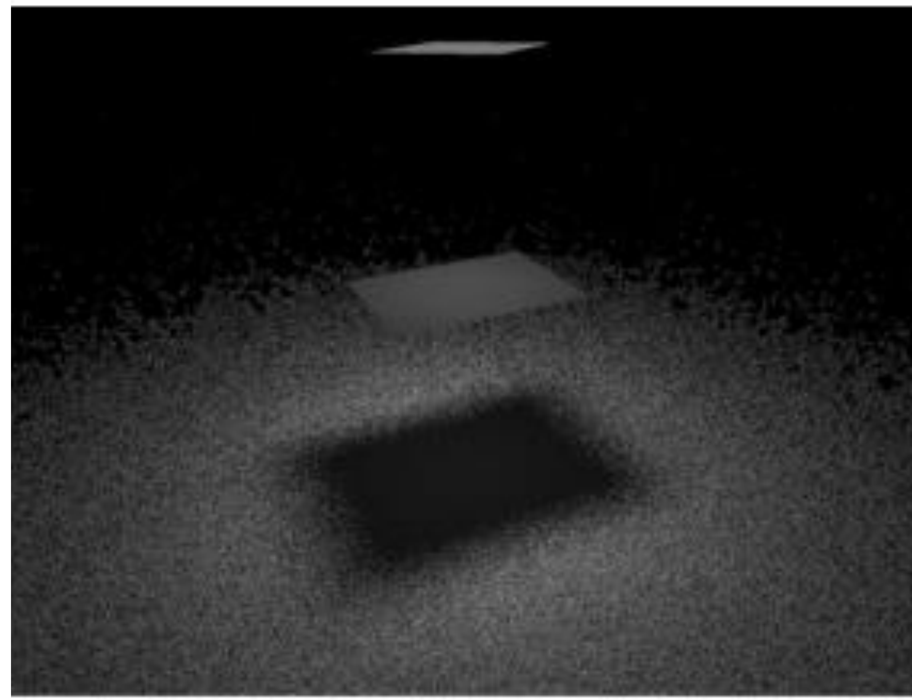
- **Estimator**

$$F_N = \frac{|A|}{N} \sum_{k=1}^{N} L_\mathrm{e}(\mathbf{y}_k \to \mathbf{x}) \cdot V(\mathbf{y}_k \leftrightarrow \mathbf{x}) \cdot G(\mathbf{y}_k \leftrightarrow \mathbf{x})$$

Light source vs. cosine sampling



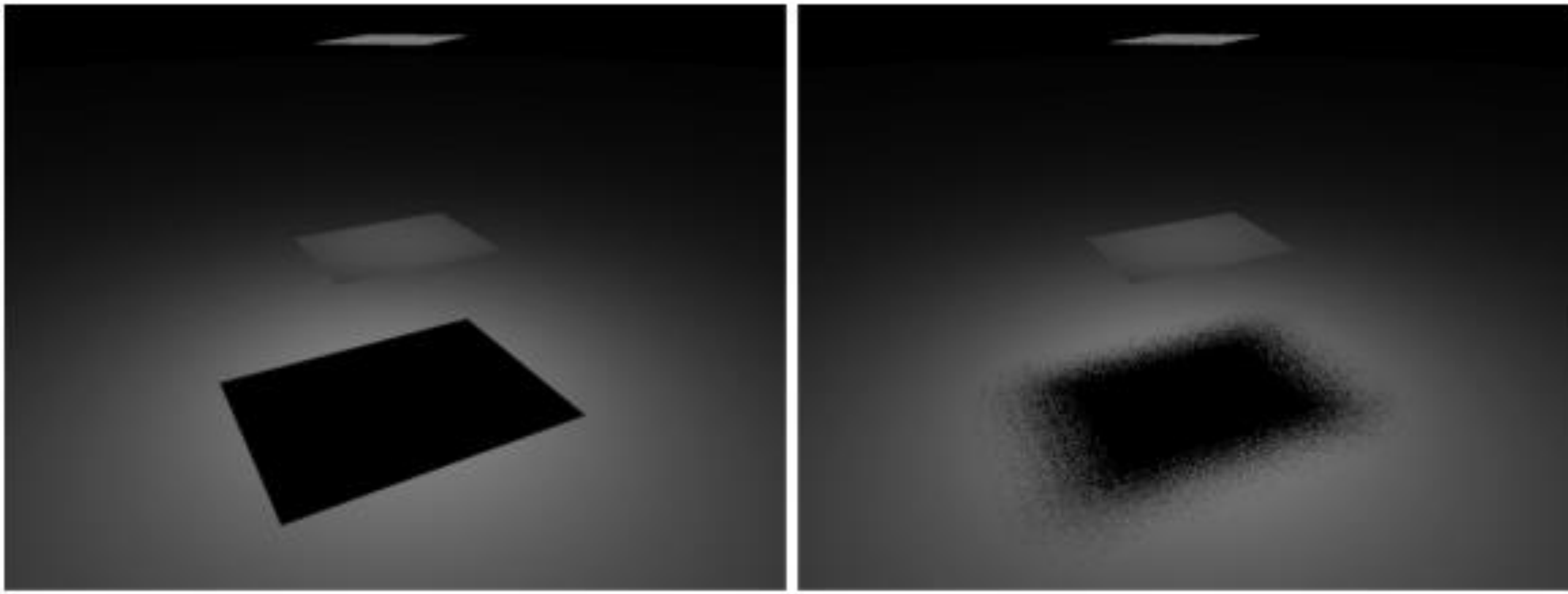
Light source **area sampling**



Cosine-proportional sampling

Images: Pat Hanrahan

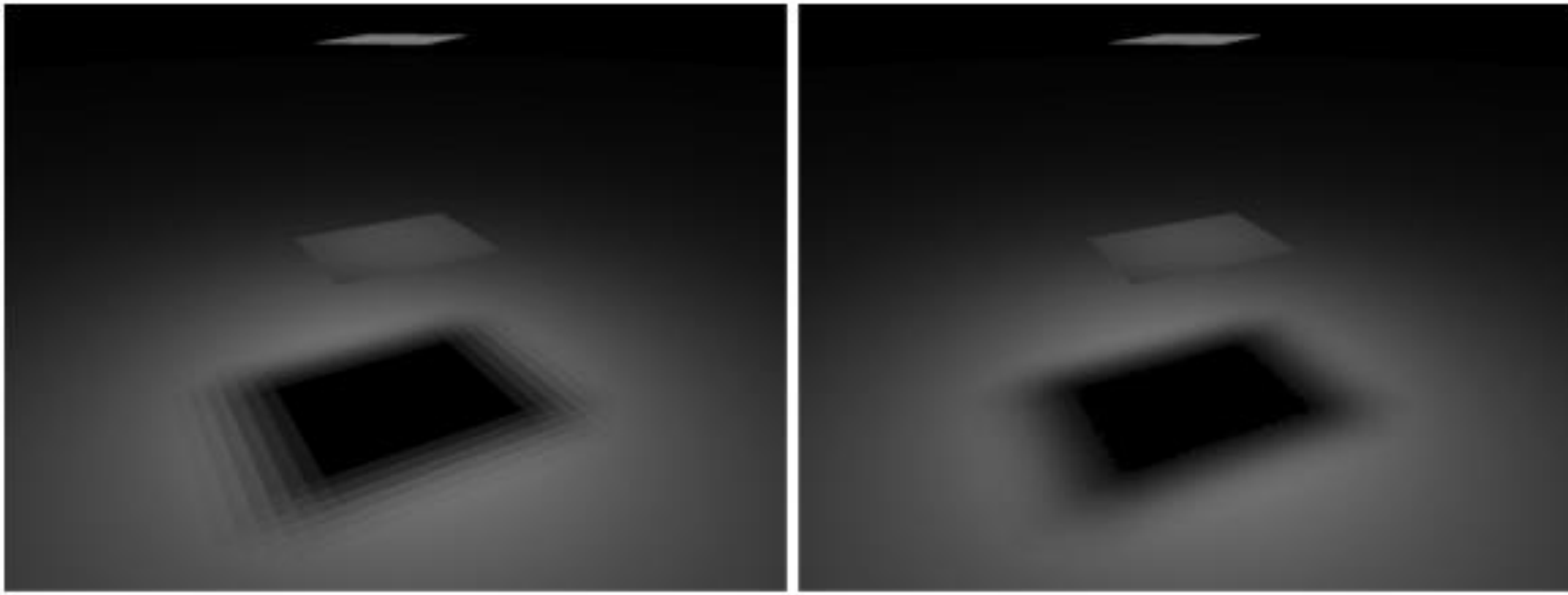
Example – Area Sampling



1 shadow ray per eye ray

Center Random

Example – Area Sampling

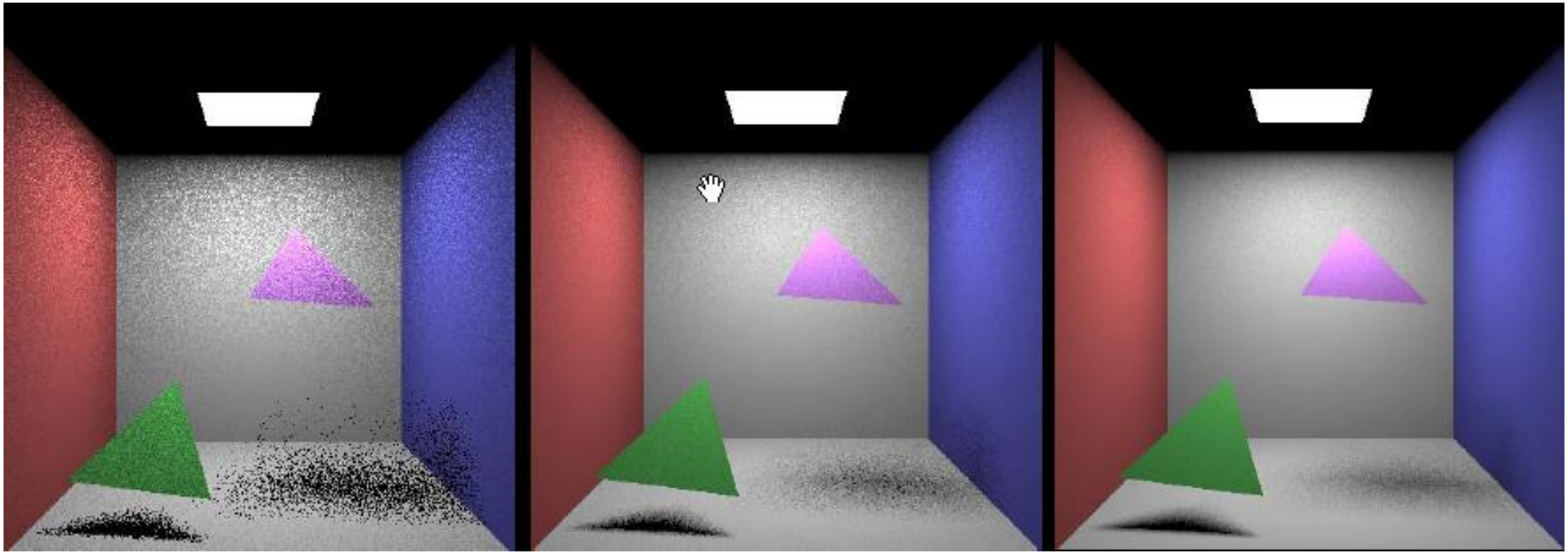


16 shadow rays per eye ray

Uniform grid

Stratified random

Area light sources



1 sample per pixel 9 samples per pixel 36 samples per pixel

Direct illumination on a surface with an arbitrary BRDF

- Integral to be estimated

$$L_{\mathrm{o}}(\mathbf{x}, \omega_{\mathrm{o}}) = \int_A L_{\mathrm{e}}(\mathbf{y} \to \mathbf{x}) \cdot f_r(\mathbf{y} \to \mathbf{x} \to \omega_{\mathrm{o}}) \cdot V(\mathbf{y} \leftrightarrow \mathbf{x}) \cdot G(\mathbf{y} \leftrightarrow \mathbf{x}) \,\mathrm{d}A$$

- **Estimator** based on uniform light source sampling

$$F_N = \frac{|A|}{N} \sum_{k=1}^{N} L_{\mathrm{e}}(\mathbf{y}_k \to \mathbf{x}) \cdot f_r(\mathbf{y}_k \to \mathbf{x} \to \omega_{\mathrm{o}}) \cdot V(\mathbf{y}_k \leftrightarrow \mathbf{x}) \cdot G(\mathbf{y}_k \leftrightarrow \mathbf{x})$$