CS5630 Physically Based Realistic Rendering

Steve Marschner

“Spring” 2026

03 Monte Carlo Integration

Computing definite integrals

Rendering is full of integration problems

- add up all the light that falls on this surface
- add up all the light that enters the camera's lens
- add up all the ways light can get to the camera

Problem defined by

- a domain
- a function
- a measure

$$I = \int_D f(x)\, dA(x)$$ — integrate f over D with respect to area

$$I = \int_D f(x)\, dx$$ — integrate f over D in the "obvioius" way

Definite because we want a numerical answer rather than a formula

Monte Carlo integration

A simple and flexible way to approximate integrals

Core mathematical ideas

- expectation and integration are closely related
- we can define a random variable whose expectation is the value of the integral
- we can estimate the expectation using means of samples
- this produces an approximation to the integral with random error

Probability ideas to review

- discrete and continuous random variables
- probability and probability density
- expectation as a probability weighted sum or probability density weighted integral
- sample mean as an estimate of expectation

Random variables

Intuitively, a random variable X is a variable with an uncertain value

- its value will be different each time you re-run the experiment that generates it
- it may be more likely to take on some values than others

Formally, X is a function defined on a probability space

- probability space Ω = “set of things ω that could happen” together with their probabilities
- $X(\omega)$ is the value X takes on when ω happens
- $f(X)$ is another random variable which takes on the value $f(X(\omega))$ when ω happens

Examples

- Ω is the set of faces of a die, and X is the number on the face that comes up
- Ω is the set of possible combinations for two dice, and X is the sum of the face values

Random variables

A r.v. may be able to take on a continuous range of values

- e.g. $\Omega = [0,1)$ with uniform probability (i.e. the output of random(), traditionally called ξ)
- $X = 6\xi$ takes on values from 0 to 6
- $X = 6\xi_1 + 6\xi_2$ takes on values from 0 to 12, and is more likely to be near 6 than near 0

Random variables have proability distributions

- for a discrete r.v. $X \sim p$, $\Pr\{X = a\} = p(a)$
  - p is the probability that X is a ; $0 \leq p \leq 1$; $\sum_a p(a) = 1$
- for a continuous r.v. $X \sim p$, $\Pr\{a \leq X < a + dx\} = p(a)\,dx$
  - p is the probability density for X to be near a ; $0 \leq p < \infty$; $\int p(a)\,da = 1$

Expectation

$E\{X\}$ **is the value we expect** X **to have on average**

For discrete X **it is a sum**

- $X \sim p \implies E\{X\} = \sum_a ap(a)$

For continuous X **it is an integral**

- $X \sim p \implies E\{X\} = \int ap(a)\, da$

For a r.v. defined as a function of X is the most often used form

- $X \sim p \implies E\{f(X)\} = \int f(a)p(a)\, da$

Expectation

Expectation is linear

- $E\{aX + bY\} = aE\{X\} + bE\{Y\}$ for non-random scalars a and b

Averaging multiple trials preserves the sample mean

- suppose $X_1, \ldots, X_N$ are identically distributed and $\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$
- then $E\{\bar{X}\} = E\{X_i\}$

So the sample mean is an estimate of the expected value

Monte Carlo integration construction

Suppose I want to compute a definite integral I

$$I = \int_D f(x)\,dx$$

…and I am able to generate values $x_i \sim p$ and their probability densities

- that is, samples from a random variable $X \sim p$

Then I can define an estimator for I

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

$$E\{g(X)\} = I$$

Monte Carlo integration algorithm

Sample many values x_i from X

Evaluate the estimator for each sample

Compute the sample mean

$$G_N = \frac{1}{N} \sum_{i=1}^{N} g(x_i)$$

Then $G_N \approx I$

- and the error goes down as N increases (more precision soon)

Convergence rate

We can get a better estimate of the expected value of *g* by generating several values and averaging them.

$$G_N = \frac{1}{N}\sum_{i=1}^{N} g(x_i) \quad \text{where} \quad x_i \sim p$$

As *n* increases, the variance of G_n decreases

$$\sigma^2\left\{\sum_{i=1}^{N} g(x_i)\right\} = \sum_{i=1}^{N} \sigma^2\{g\} = N\sigma^2\{g\}$$

$$\sigma\{G_N\} = \frac{\sigma\{g\}}{\sqrt{N}}$$

…but it doesn't decrease that fast ("order $N^{1/2}$ convergence")

Importance sampling

Monte Carlo integration has few requirements on p

- p cannot be zero where f is not zero
- practical requirement: you have to know p for the samples you generate

…but some pdfs produce better estimates than others

Rule of thumb: make p **resemble** f

Ideal case is when $p(x) = \dfrac{f(x)}{C}$

- then the estimator is $g(x) = \dfrac{f(x)}{p(x)} = C$
- it's a constant — perfection in one sample! Of course this means we already know the answer.

Generating samples

With the default RNG it is easy to sample uniform distributions

How to sample nonuniform ones?

- well, somehow we write a program that calls the RNG and does something with the output

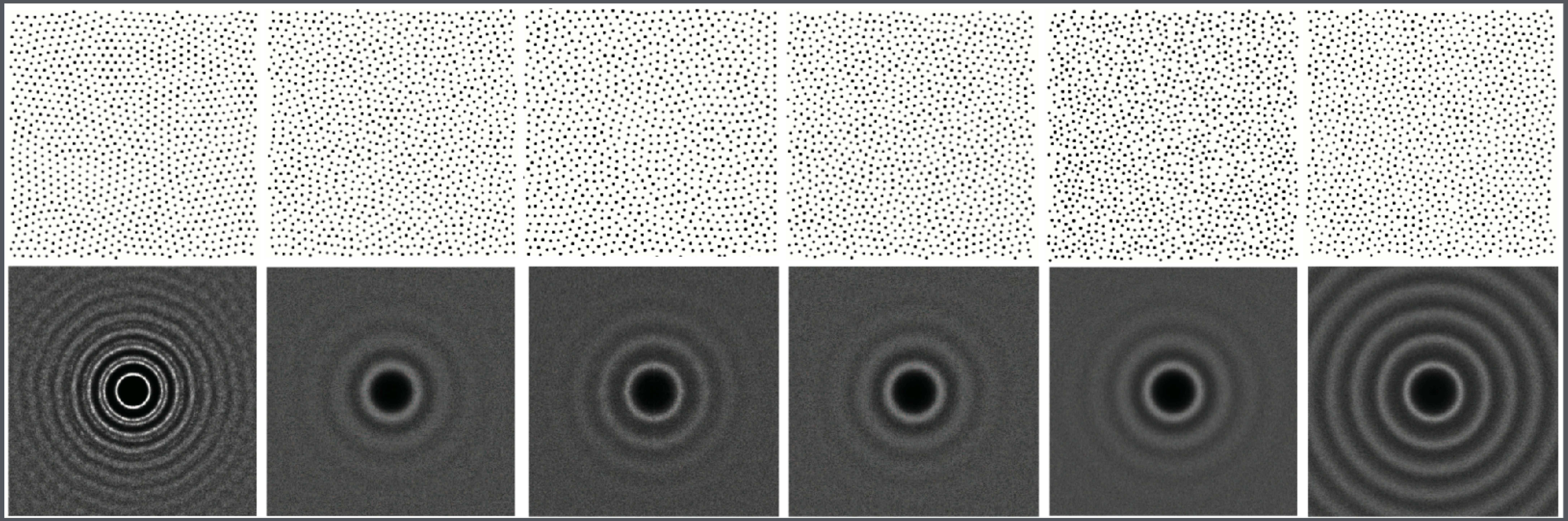
Suppose we have the cumulative distribution function for X

- $P(a) = \Pr\{X < a\}$. It is a monotonically increasing function.

Further suppose that, given ξ from the RNG we can compute x so that $P(x) = \xi$

- then for any a, $x < a$ exactly when $\xi < P(a)$, which happens with probability $P(a)$

Blue noise point sets



Dong-Ming Yan et al. 2015

Quasi Monte Carlo sequences

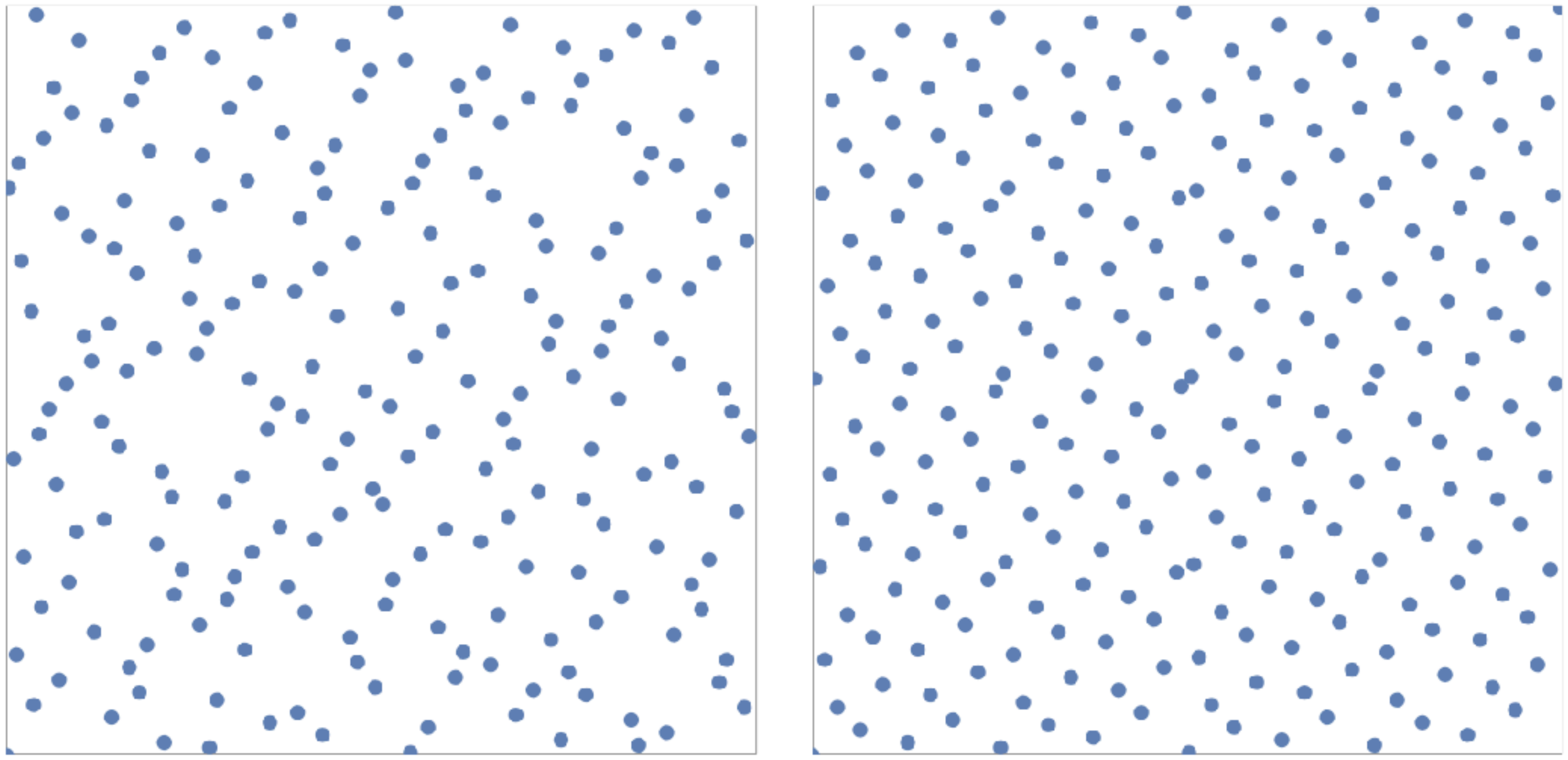


Figure 7.25: The First Points of Two Low-Discrepancy Sequences in 2D. (a) Halton (216 points), (b) Hammersley (256 points).