Variance Reduction
CMU 15-462/662

Last time: Monte Carlo Ray Tracing

- Recursive description of incident illumination
- Difficult to integrate; *tour de force* of numerical integration
- Leads to lots of sophisticated integration strategies:
  - sampling strategies
  - variance reduction
  - Markov chain methods
  - ...
- Today: get a glimpse of these ideas
- Also valuable outside rendering!
  - Monte Carlo one of the “Top 10 Algorithms of the 20th Century”!

\[ L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} f_r(x, \omega_i, \omega_o) L_i(x, \omega_i) (\omega_i \cdot n) \, d\omega_i \]
Review: Monte Carlo Integration

Want to integrate: \[ I := \int_{\Omega} f(x) \, dx \]

General-purpose hammer: Monte-Carlo integration

\[ I = \lim_{n \to \infty} V(\Omega) \frac{1}{n} \sum_{i=1}^{n} f(X_i) \]

*Must of course have a well-defined integral!
Review: Expected Value (DISCRETE)

A discrete random variable $X$ has $n$ possible outcomes $x_i$, occurring with probabilities $0 \leq p_i \leq 1$, $p_1 + \ldots + p_n = 1$

$$E(X) := \sum_{i=1}^{n} p_i x_i$$

E.g., what’s the expected value for a fair coin toss?

$p_1 = 1/2$
$x_1 = 1$

$p_2 = 1/2$
$x_2 = 0$
Continuous Random Variables

A *continuous* random variable $X$ takes values $x$ anywhere in a set $\Omega$

Probability *density* $p$ gives probability $x$ appears in a given *region*.

E.g., probability you fall asleep at time $t$ in a 15-462 lecture:

$$\int_{t_0}^{t_1} p(t) \, dt$$

- cool motivating examples
- theory
- professor is making dumb jokes
- more theory
- class ends
- probability you fall asleep *exactly* at any given time $t$ is ZERO!
- can only talk about chance of falling asleep in a given *interval* of time
Review: Expected Value (CONTINUOUS)

Expected value of continuous random variable again just the “weighted average” with respect to probability $p$:

$$E(X) := \int_{\Omega} x p(x) \, dx$$

probability density at point $x$

equation

sometimes just use “$\mu$” (for “mean”)

E.g., expected time of falling asleep?

$$\mu = 44.9 \text{ minutes}$$

(is this result counter-intuitive?)
Flaw of Averages
Review: Variance

- **Expected value is the “average value”**
- **Variance is how far we are from the average, on average!**

\[
\text{Var}(X) := E[(X - E[X])^2]
\]

**DISCRETE**

\[
\sum_{i=1}^{n} p_i (x_i - \sum_{j} p_j x_j)^2
\]

**CONTINUOUS**

\[
\int_{\Omega} p(x)(x - \int_{\Omega} y p(y) \, dy)^2 \, dx
\]

- **Standard deviation** $\sigma$ is just the square root of variance

$\mu = 44.9$ minutes

$\sigma = 15.8$ minutes

(More intuitive perhaps?)
Variance Reduction in Rendering

higher variance  lower variance
Q: How do we reduce variance?
Q: What's the expected value of the integrand $f$?
A: Just by inspection, it's 1/2 (half white, half black!).

Q: What's its variance?
A: \[(1/2)(0-1/2)^2 + (1/2)(1-1/2)^2 = (1/2)(1/4) + (1/2)(1/4) = 1/4\]

Q: How do we reduce the variance?
That was a trick question.

You can’t reduce variance of the *integrand*!
Can only reduce variance of an *estimator*.
Variance of an Estimator

- An “estimator” is a formula used to approximate an integral
- Most important example: our Monte Carlo estimate:
  \[ I = \int_{\Omega} f(x) \, dx \quad \hat{I} := V(\Omega) \frac{1}{n} \sum_{i=1}^{n} f(x_i) \]
  \[ \text{true integral} \quad \text{Monte Carlo estimate} \]
- Get different estimates for different collections of samples
- Want to reduce variance of estimate across different samples
- Why? Integral itself only has one value!
- Many, many (many) techniques for reducing variance
- We will review some key examples for rendering
Bias & Consistency

- Two important things to ask about an estimator
  - Is it *consistent*?
  - Is it *biased*?

- Consistency: “converges to the correct answer”

\[
\lim_{n \to \infty} P(|I - \hat{I}_n| > 0) = 0
\]

- Unbiased: “estimate is correct on average”

\[
E[I - \hat{I}_n] = 0
\]

- Consistent does not imply unbiased!

...even if n=1! (only one sample)
Example 1: Consistent or Unbiased?

- My estimator for the integral over an image:
  - take $n = m \times m$ samples at fixed grid points
  - sum the contributions of each box
  - let $m \to \infty$

Is this estimator consistent? Unbiased?
Example 2: Consistent or Unbiased?

- My estimator for the integral over an image:
  - take only a single random sample of the image (n=1)
  - multiply it by the image area
  - use this value as my estimate

Is this estimator consistent? Unbiased?
(What if I then let n go to ∞?)
Why does it matter?

Rule of thumb: unbiased estimators have more predictable behavior / fewer parameters to tweak to get correct result (which says nothing about performance...)

biased + inconsistent

consistent + unbiased

biased + consistent
## Consistency & Bias in Rendering Algorithms

<table>
<thead>
<tr>
<th>method</th>
<th>consistent?</th>
<th>unbiased?</th>
</tr>
</thead>
<tbody>
<tr>
<td>rasterization*</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>path tracing</td>
<td>ALMOST</td>
<td>ALMOST</td>
</tr>
<tr>
<td>bidirectional path tracing</td>
<td>???</td>
<td>???</td>
</tr>
<tr>
<td>Metropolis light transport</td>
<td>???</td>
<td>???</td>
</tr>
<tr>
<td>photon mapping</td>
<td>???</td>
<td>???</td>
</tr>
<tr>
<td>radiosity</td>
<td>???</td>
<td>???</td>
</tr>
</tbody>
</table>

*But very high performance!*
Q: What’s the probability we sample the reflected direction?
A: ZERO.

Q: What’s the probability we hit a point light source?
A: ZERO.
Naïve path tracing misses important phenomena!
(Formally: the result is biased.)
...But isn’t this example pathological?
No such thing as point light source, perfect mirror.
Real lighting can be close to pathological

small directional light source

near-perfect mirror

Still want to render this scene!
Light has a very “spiky” distribution

- Consider the view from each bounce in our disco scene:

  - view from camera
  - view from diffuse bounce: mirrored ball (pink) covers small percentage of solid angle
  - view from specular bounce: area light (white) covers small percentage of solid angle

Probability that a uniformly-sampled path carries light is the product of the solid angle fractions. (Very small!)

Then consider even more bounces...
Just use more samples?

path tracing - 16 samples/pixel

path tracing - 128 samples/pixel

path tracing - 8192 samples/pixel

how do we get here? (photo)
We need better sampling strategies!
Review: Importance Sampling

Simple idea: sample the integrand according to how much we expect it to contribute to the integral.

\[ f(x) \]

complicated integrand

\[ p(x) \]

our best guess for where the integrand is “big”

naive Monte Carlo:

\[
V(\Omega) \frac{1}{n} \sum_{i=1}^{n} f(x_i)
\]

(x\_i are sampled uniformly)

importance sampled Monte Carlo:

\[
\frac{1}{n} \sum_{i=1}^{n} \frac{f(x_i)}{p(x_i)}
\]

(x\_i are sampled proportional to p)

“If I sample x more frequently, each sample should count for less; if I sample x less frequently, each sample should count for more.”

Q: What happens when p is proportional to f (p = cf)?
Importance Sampling in Rendering

- **materials:** sample important “lobes”
- **illumination:** sample bright lights

(important special case: perfect mirror!)

(important special case: point light!)

Q: How else can we re-weight our choice of samples?
Path Space Formulation of Light Transport

- So far have been using recursive rendering equation:

\[ L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} f_r(x, \omega_i, \omega_o) L_i(x, \omega_i)(\omega_i \cdot n) \, d\omega_i \]

- Make intelligent “local” choices at each step (material/lights)

- Alternatively, we can use a “path integral” formulation:

\[ I = \int_{\Omega} f(\bar{x}) \, d\mu(\bar{x}) \]

- Opens the door to intelligent “global” importance sampling. (But still hard!)
Unit Hypercube View of Path Space

- Paths determined by a sequence of random values $\xi$ in $[0,1]$
- Hence, path of length $k$ is a point in hypercube $[0,1]^k$
- “Just” integrate over cubes of each dimension $k$
- E.g., two bounces in a 2D scene:

Each point is a path of length 2:

Each bounce: $\xi \in [0, 1] \rightarrow \theta \in [0, \pi]$

Total brightness of this image $\Leftrightarrow$ total contribution of length-2 paths.
How do we choose paths—and path lengths?
Bidirectional Path Tracing

- Forward path tracing: no control over path length (hits light after \( n \) bounces, or gets terminated by Russian Roulette)

- Idea: connect paths from light, eye ("bidirectional")

Importance sampling? Need to carefully weight contributions of path according to sampling strategy.

(Details in Veach & Guibas, "Bidirectional Estimators for Light Transport")
Bidirectional Path Tracing (Path Length=2)

- Standard (forward) path tracing
  - Fails for point light sources

- Direct lighting

- Visualize particles from light

- Backward path tracing
  - Fails for a pinhole camera
Contributions of Different Path Lengths

final image
Good paths can be hard to find!

Idea:
Once we find a good path, perturb it to find nearby “good” paths.
Metropolis-Hastings Algorithm (MH)

- Standard Monte Carlo: sum up independent samples
- MH: take random walk of dependent samples ("mutations")
- Basic idea: prefer to take steps that increase sample value

If careful, sample distribution will be proportional to integrand
- make sure mutations are "ergodic" (reach whole space)
- need to take a long walk, so initial point doesn’t matter ("mixing")

\[ \alpha := \frac{f(x')}{f(x_i)} \]

if random # in \([0,1]\) < \(\alpha\):
\[ x_{i+1} = x' \]
else:
\[ x_{i+1} = x_i \]

"transition probability"
Metropolis-Hastings: Sampling an Image

- Want to take samples proportional to image density $f$
- Start at random point; take steps in (normal) random direction
- Occasionally jump to random point (ergodicity)
- Transition probability is “relative darkness” $f(x')/f(x_i)$

![short walk](image1.png)
![long walk](image2.png)
![original image](image3.png)
Metropolis Light Transport

Basic idea: mutate paths

(For details see Veach, “Robust Monte Carlo Methods for Light Transport Simulation”)
Multiple Importance Sampling (MIS)

- Many possible importance sampling strategies
- Which one should we use for a given integrand?
- MIS: combine strategies to preserve strengths of all of them
- Balance heuristic is (provably!) about as good as anything:

\[
\frac{1}{N} \sum_{i=1}^{n} \sum_{j=1}^{n_i} \frac{f(x_{ij})}{\sum_{k} c_k p_k(x_{ij})}
\]

Still, several improvements possible (cutoff, power, max)—see Veach & Guibas.
Multiple Importance Sampling: Example

sample materials

multiple importance sampling (power heuristic)

sample lights
Ok, so importance is important.

But how do we sample our function in the first place?
Sampling Patterns & Variance Reduction

- Want to pick samples according to a given density
- But even for uniform density, lots of possible sampling patterns
- Sampling pattern will affect variance (of estimator!)

uniform sampling density
nonuniform sampling density
Stratified Sampling

- How do we pick \( n \) values from \([0,1]\)?
- Could just pick \( n \) samples uniformly at random
- Alternatively: split into \( n \) bins, pick uniformly in each bin

**FACT**: stratified estimate never has larger variance (often lower)

Intuition: each stratum has smaller variance. (Proof by linearity of expectation!)
Stratified Sampling in Rendering/Graphics

- Simply replacing uniform samples with stratified ones already improves quality of sampling for rendering (…and other graphics/visualization tasks!)

See especially: Jim Arvo, “Stratified Sampling of Spherical Triangles” (SIGGRAPH 1995)
Low-Discrepancy Sampling

- "No clumps" hints at one possible criterion for a good sample:
- **Number of samples should be proportional to area**
- **Discrepancy** measures deviation from this ideal

$$d_S(X) := \left| \frac{A(S)}{|X|} - \frac{n(S)}{|X|} \right|$$

overall discrepancy of $X$

$$D(X) := \max_{S \in F} \{d_S(X)\}$$

(ideally equal to zero!)

See especially: Dobkin et al, "Computing Discrepancy w/ Applications to Supersampling" (1996)
Quasi-Monte Carlo methods (QMC)

- Replace truly random samples with low-discrepancy samples
- **Why? Koksma’s theorem:**
  \[
  \left| \frac{1}{n} \sum_{i=1}^{n} f(x_i) - \int_{0}^{1} f(x) \, dx \right| \leq \mathcal{V}(f) D(X)
  \]
  - I.e., for low-discrepancy $X$, estimate approaches integral
  - Similar bounds can be shown in higher dimensions
  - **WARNING:** total variation not always bounded!
  - **WARNING:** only for family $F$ of *axis-aligned* boxes $S$!
  - E.g., edges can have arbitrary orientation (coverage)
  - Discrepancy still a very reasonable criterion in practice

**Discrepancy** still a very reasonable criterion in practice
Hammersley & Halton Points

- Can easily generate samples with near-optimal discrepancy
- First define radical inverse $\varphi_r(i)$
- Express integer $i$ in base $r$, then reflect digits around decimal
- E.g., $\varphi_{10}(1234) = 0.4321$
- Can get $n$ Halton points $x_1, \ldots, x_n$ in $k$-dimensions via
  $$x_i = (\phi_{P_1}(i), \phi_{P_2}(i), \ldots, \phi_{P_k}(i))$$
- Similarly, Hammersley sequence is
  $$x_i = \left(\frac{i}{n}, \phi_{P_1}(i), \phi_{P_2}(i), \ldots, \phi_{P_{k-1}}(i)\right)$$

$n$ must be known ahead of time!
Wait, but doesn’t a regular grid have really low discrepancy...?
There’s more to life than discrepancy

- Even low-discrepancy patterns can exhibit poor behavior:

\[
\frac{1}{n} \sum_{i=1}^{n} f(x_i) = 1
\]

\[
\frac{1}{n} \sum_{i=1}^{n} f(x_i) = 0
\]

- Want pattern to be \textit{anisotropic} (no preferred direction)
- Also want to avoid any preferred \textit{frequency} (see above!)
Blue Noise - Motivation

- Can observe that monkey retina exhibits *blue noise* pattern [Yellott 1983]

![Blue Noise Pattern](image1)

- No obvious preferred directions (anisotropic)
- What about frequencies?
Blue Noise - Fourier Transform

- Can analyze quality of a sample pattern in *Fourier domain*

- Regular pattern has “spikes” at regular intervals
- Blue noise is spread evenly over all frequencies in all directions
- Bright center “ring” corresponds to sample spacing
Spectrum affects reconstruction quality

(from Balzer et al 2009)
Poisson Disk Sampling

- How do you generate a “nice” sample?
- One of the earliest algorithms: Poisson disk sampling
- Iteratively add random non-overlapping disks (until no space left)

Decent spectral quality, but we can do better.
Lloyd Relaxation

- Iteratively move each disk to the center of its neighbors

Better spectral quality, slow to converge. Can do better yet...
Voronoi-Based Methods

- Natural evolution of Lloyd
- Associate each sample with set of closest points (Voronoi cell)
- Optimize qualities of this Voronoi diagram
- E.g., sample is at cell’s center of mass, cells have same area, etc.

![Voronoi diagram](image)

- Voronoi
- Centroidal
- Equal area
Adaptive Blue Noise

- Can adjust cell size to sample a given density (e.g., importance)

Computational tradeoff: expensive* precomputation / efficient sampling.

*But these days, not that expensive...
How do we efficiently sample from a large distribution?
Sampling from the CDF

To randomly select an event, select $x_i$ if

$$P_{i-1} < \xi < P_i$$

Uniform random variable $\in [0, 1]$}

e.g., # of pixels in an environment map (big!)

Cost? $O(n \log n)$
Alias Table

- Get amortized $O(1)$ sampling by building “alias table”
- Basic idea: rob from the rich, give to the poor ($O(n)$):

- Table just stores two identities & ratio of heights per column
- To sample:
  - pick uniform # between 1 and $n$
  - biased coin flip to pick one of the two identities in $n$th column
Ok, great!

Now that we’ve mastered Monte Carlo rendering, what other techniques are there?
Photon Mapping

- Trace particles from light, deposit “photons” in kd-tree
- Especially useful for, e.g., caustics, participating media (fog)

Voronoi diagrams can be used to improve photon distribution

(from Spencer & Jones 2013)
Finite Element Radiosity

- Very different approach: transport between patches in scene
- Solve large linear system for equilibrium distribution
- Good for diffuse lighting; hard to capture other light paths
## Consistency & Bias in Rendering Algorithms

<table>
<thead>
<tr>
<th>method</th>
<th>consistent?</th>
<th>unbiased?</th>
</tr>
</thead>
<tbody>
<tr>
<td>rasterization</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>path tracing</td>
<td>ALMOST</td>
<td>ALMOST</td>
</tr>
<tr>
<td>bidirectional path tracing</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Metropolis light transport</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>photon mapping</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>radiosity</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>
Can you certify a renderer?

- Grand challenge: write a renderer that comes with a certificate (i.e., provable, formally-verified guarantee) that the image produced represents the illumination in a scene.

- Harder than you might think!

- Inherent limitation of sampling: you can never be 100% certain that you didn’t miss something important.

![Diagram of eye, pinhole, and sun]

Can always make sun brighter, hole smaller...!