Overview

Earlier lecture
- Monte Carlo integration

Today
- Variance reduction
- Importance sampling
- Stratified sampling
- Multidimensional sampling patterns
- Discrepancy and Quasi-Monte Carlo

Next lecture
- Signal processing view of sampling

Camera Simulation

<table>
<thead>
<tr>
<th>Effect</th>
<th>Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field of view</td>
<td>Film size, stops and pupils</td>
</tr>
<tr>
<td>Depth of field</td>
<td>Aperture, focal length</td>
</tr>
<tr>
<td>Exposure</td>
<td>Film speed, aperture, shutter</td>
</tr>
<tr>
<td>Motion blur</td>
<td>Shutter</td>
</tr>
</tbody>
</table>
Camera Simulation

\[ R = \int \int \int P(x', \lambda) S(x', \omega', t) L(T(x', \omega', \lambda), t, \lambda) \, d\Omega(x') \, d\omega' \, dt \, d\lambda \]

- **Sensor response**: \( P(x', \lambda) \)
- **Lens**: \( (x, \omega) = T(x', \omega', \lambda) \)
- **Shutter**: \( S(x', \omega', t) \)
- **Scene radiance**: \( L(x, \omega, t, \lambda) \)

Monte Carlo Calculation

- **16 shadow rays per eye ray**
- **Uniform grid**
- **Stratified random**
Unbiased Estimator

\[ E[F_N] = I(f) \]

\[ E[F_N] = E[\frac{1}{N} \sum_{i=1}^{N} Y_i] \]

\[ = \frac{1}{N} \sum_{i=1}^{N} E[Y_i] = \frac{1}{N} \sum_{i=1}^{N} E[f(X_i)] \]

Properties

\[ E[\sum_i Y_i] = \sum_i E[Y_i] \]
\[ E[aY] = aE[Y] \]

Assume uniform probability distribution for now

Variance

1 shadow ray per eye ray  16 shadow rays per eye ray
Variance

Definition

\[ V[Y] = E[(Y - E[Y])^2] \]
\[ = E[Y^2] - E[Y]^2 \]

Variance decreases linearly with sample size

\[ V\left[ \frac{1}{N} \sum_{i=1}^{N} Y_i \right] = \frac{1}{N^2} \sum_{i=1}^{N} V[Y_i] = \frac{1}{N^2} NV[Y] = \frac{1}{N} V[Y] \]

\[ V[aY] = a^2V[Y] \]

Variance Reduction

Efficiency measure

\[ Efficiency \propto \frac{1}{Variance \cdot Cost} \]

If one technique has twice the variance as another technique, then it takes twice as many samples to achieve the same variance.

If one technique has twice the cost of another technique with the same variance, then it takes twice as much time to achieve the same variance.
**Biasing**

Previously used a uniform probability distribution

Can use another probability distribution

\[ X_i \sim p(x) \]

But must change the estimator

\[ Y_i = \frac{f(X_i)}{p(X_i)} \]

---

**Unbiased Estimate**

**Probability**  \( X_i \sim p(x) \)

**Estimator**  \[ Y_i = \frac{f(X_i)}{p(X_i)} \]

**Proof**

\[
E[Y_i] = E \left[ \frac{f(x)}{p(x)} \right] = \int \frac{f(x)}{p(x)} p(x) dx = \int f(x) dx
\]
Importance Sampling

Sample according to $f$

$$\tilde{p}(x) = \frac{f(x)}{E[f]}$$

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

$$E[\tilde{f}^2] = \int \left[ \frac{f(x)}{\tilde{p}(x)} \right]^2 \tilde{p}(x) \, dx$$

Variance

$$V[f] = E[f^2] - E^2[f]$$

$$= E[f] \int f(x) \, dx$$

Zero variance!

$$V[\tilde{f}^2] = 0$$

Gotcha?

Examples

Projected solid angle

4 eye rays per pixel
100 shadow rays

Area

4 eye rays per pixel
100 shadow rays
Stratified Sampling

Allocate samples per region
Estimate each region separately

\[ F_N = \frac{1}{N} \sum_{i=1}^{N} F_i \]

New variance

\[ V[F_N] = \frac{1}{N^2} \sum_{i=1}^{N} V[F_i] \]

Stratified Sampling

Sample a polygon

\[ V[F_N] = \frac{1}{N^2} \sum_{i=1}^{N} V[F_i] = \frac{V[F_E]}{N^{1.5}} \]

If the variance in each region is less than the overall variance, there will be a reduction in overall variance.
Sampling a Circle

Equi-Areal

\[ \theta = 2\pi U_1 \]
\[ r = \sqrt{U_2} \]

Shirley’s Mapping

\[ r = U_1 \]
\[ \theta = \frac{\pi U_2}{4 U_1} \]
High-dimensional Sampling

Complete set of samples \( N = n \times n \times \ldots \times n = n^d \)

Random sampling

Error ... \( E \sim V^{1/2} \sim \frac{1}{N^{1/2}} \)

Numerical integration

Error ... \( E \sim \frac{1}{n} = \frac{1}{N^{1/d}} \)

Monte Carlo requires fewer samples for the same error in high dimensional space

Cameras (5D integral)

\[
R = \iiint_{T \Omega A} L(x, \omega, t) \ P(x) \ S(t) \ \cos \theta \ dA \ d\omega \ dt
\]

Motion Blur  
Depth of Field

Cook, Porter, Carpenter, 1984  
Mitchell, 1991
Space-time Patterns

Distribute samples in time
- Complete in space
- Incomplete in time
- Decorrelate space and time
- Nearby samples in space should differ greatly in time

Cook Pattern

<table>
<thead>
<tr>
<th>6</th>
<th>10</th>
<th>2</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>14</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

Pan-diagonal Magic Square

<table>
<thead>
<tr>
<th>15</th>
<th>8</th>
<th>5</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>3</td>
<td>14</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>13</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>

Uniform random

Spectrally optimized
Path Tracing

4 eye rays per pixel
16 shadow rays per eye ray

Complete

64 eye rays per pixel
1 shadow ray per eye ray

Incomplete

Block Design

Latin Square

<table>
<thead>
<tr>
<th>a</th>
<th>d</th>
<th>c</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>a</td>
<td>d</td>
<td>c</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>a</td>
<td>d</td>
</tr>
<tr>
<td>d</td>
<td>c</td>
<td>b</td>
<td>a</td>
</tr>
</tbody>
</table>

Alphabet of size $n$
Each symbol appears exactly once in each row and column
Rows and columns are stratified
Block Design

N-Rook Pattern

Incomplete block design

Replaced $n^2$ samples with $n$ samples

Permutations: $(\pi_1(i), \pi_2(i), \cdots \pi_d(i))$

Generalizations: N-queens, 2D projection

$(\pi_x = \{1, 2, 3, 4\}, \pi_y = \{4, 2, 3, 1\})$

Discrepancy

$\Delta(x, y) = \frac{n(x, y)}{N} - xy$

$A = xy$

$n(x, y)$ number of samples in $A$

$D_N = \max_{x,y} |\Delta(x, y)|$
Theorem on Total Variation

Theorem: \[ \left| \frac{1}{N} \sum_{i=1}^{N} f(X_i) - \int f(x) \, dx \right| \leq V(f) D_N \]

Proof: Integrate by parts

\[ \int f(x) \left[ \frac{\delta(x-x_i)}{N} - 1 \right] \, dx = \int f(x) \frac{\partial \Delta(x)}{\partial x} \, dx \]

\[ = \int f(x) \frac{\partial \Delta(x)}{\partial x} \, dx \]

\[ = f \Delta_i^0 - \int \frac{\partial f(x)}{\partial x} \Delta(x) \, dx = - \int \frac{\partial f(x)}{\partial x} \Delta(x) \, dx \]

\[ \leq D_N \int \left| \frac{\partial f(x)}{\partial x} \right| \, dx = V(f) D_N \]

Quasi-Monte Carlo Patterns

Radical inverse (digit reverse)

of integer \( i \) in integer base \( b \)

\[ i = d_i \cdots d_2 d_1 \]

\[ \phi_2(i) \]

\[ \phi_0(i) \equiv 0.d_0 d_1 d_2 \cdots d_i \]

Hammersley points

\[ \left( i / N, \phi_2(i), \phi_3(i), \phi_3(i), \cdots \right) \]

\[ D_N = O \left( \frac{\log^{d-1} N}{N} \right) \]

Halton points (sequential)

\[ \left( \phi_2(i), \phi_3(i), \phi_3(i), \cdots \right) \]

\[ D_N = O \left( \frac{\log^d N}{N} \right) \]
Hammersly Points

\[(i / N, \phi_2(i), \phi_3(i), \phi_5(i), \ldots)\]

Edge Discrepancy

Note: SGI IR Multisampling extension: 8x8 subpixel grid; 1, 2, 4, 8 samples
Low-Discrepancy Patterns

<table>
<thead>
<tr>
<th>Process</th>
<th>16 points</th>
<th>256 points</th>
<th>1600 points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zaremba</td>
<td>0.0504</td>
<td>0.00478</td>
<td>0.00111</td>
</tr>
<tr>
<td>Jittered</td>
<td>0.0538</td>
<td>0.00595</td>
<td>0.00146</td>
</tr>
<tr>
<td>Poisson-Disk</td>
<td>0.0613</td>
<td>0.00767</td>
<td>0.00241</td>
</tr>
<tr>
<td>N-Rooks</td>
<td>0.0637</td>
<td>0.0123</td>
<td>0.00488</td>
</tr>
<tr>
<td>Random</td>
<td>0.0924</td>
<td>0.0224</td>
<td>0.00866</td>
</tr>
</tbody>
</table>

Discrepancy of random edges, From Mitchell (1992)

Random sampling converges as $N^{-1/2}$
Zaremba converges faster and has lower discrepancy
Zaremba has a relatively poor blue noise spectra
Jittered and Poisson-Disk recommended

Views of Integration

1. Numerical
   - Quadrature/Integration rules
   - Smooth functions
2. Statistical sampling (Monte Carlo)
   - Sampling like polling
   - Variance reduction techniques
   - High dimensional sampling: $1/N^{1/2}$
3. Quasi Monte Carlo
   - Discrepancy
   - Asymptotic efficiency in high dimensions
4. Signal processing
   - Sampling and reconstruction
   - Aliasing and antialiasing
   - Blue noise good