Topics

MC Integration - estimators, bias, variance

Variance Reduction

- **Expected values**
- **Importance sampling**
- **Stratified sampling**

Application to Rendering

Random Walks and Integral Equations

Importance and Adjoints

Estimators

Quantity of interest Q $$Q = \int_0^1 g_1(x)\,dx \Big/ \int_0^1 g_2(x)\,dx$$

Estimator $\theta(\xi_1, \mathbf{K}, \xi_N) \approx Q$ $$\theta = \sum_{i=1}^{N} g_1(\xi_i) \Big/ \sum_{i=1}^{N} g_2(\xi_i)$$

Consistent estimator $$P\left[\lim_{N\to\infty} \theta(\xi_1, \mathbf{K}, \xi_N) = Q\right] = 1$$

Unbiased estimator $$E[\theta] = Q$$

Bias $$Bias = E[\theta] - Q$$

Estimators: error and variance

Variance $$Var = E\left[(\theta - E[\theta])^2\right]$$

Mean-squared error $$MSE = E\left[(\theta - Q)^2\right]$$
$$= Var + Bias^2$$

Example: $Q = \frac{1}{2} =$ **mean of samples** ξ_i **from** $U[0,1]$

$$\theta_1 = \tfrac{1}{N}\sum_{i=1}^{N}\xi_i \qquad \theta_2 = \tfrac{1}{N}\max_{i=1}^{N}\xi_i$$
$$E[\theta_1] = \tfrac{1}{2} = Q \qquad E[\theta_2] = Q\cdot(1+O(\tfrac{1}{N}))$$
$$Var[\theta_1] = O(\tfrac{1}{N}) \qquad Var[\theta_2] = O(\tfrac{1}{N^2})$$

Monte Carlo Integration, Take II

General integral $$I = \int_{\Omega} f(x)\, d\mu(x)$$

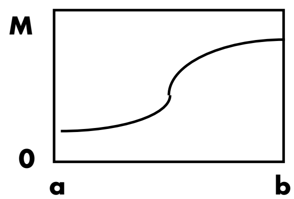
Measure $d\mu(x)$ $$dA, \quad d\omega, \quad d\vec{A}\cdot d\vec{\omega}$$

The basic estimator $$E\left[\frac{f(x)}{p(x)}\right] = \int_{\Omega}\frac{f(x)}{p(x)}p(x)\, d\mu(x) = I$$

What we get: a collection of "weighted samples" $\left(X_i, \frac{f(X_i)}{p(X_i)}\right)$

Can estimate several integrals at once: $\theta_j = \frac{1}{N}\sum_{i=1}^{N}\frac{f_j(X_i)}{p(X_i)}$

Monte Carlo Integration Examples



As a 1D integral

$$X \sim U[a,b]$$

$$\theta = \frac{f(X)}{p(X)} = (b-a)f(X)$$

As a 2D integral

$$X \sim U[a,b] \times U[0,M]$$

$$\theta = \frac{[x_2 \le f(x_1)]}{p(x_1,x_2)}$$

$$= (b-a)M \begin{cases} 0, & x_2 > f(x_1) \\ 1, & x_2 \le f(x_1) \end{cases}$$

Variance Reduction

Efficiency measure

$$Efficiency \propto \frac{1}{Variance \bullet Cost}$$

Some techniques

- **Expected values**
- **Importance sampling**
- **Stratified sampling**
- **Russian roulette and splitting**
- **Control variates, correlated, antithetic**
- **Exponential transform**

Expected values

Idea: integrate analytically where possible.

$$I = \iint f(x,y)\, dx\, dy$$

$$\theta_1 = f(X,Y) / p(X,Y)$$

Can we compute $f_2(x) = \int f(x,y)\, dy$ **?**

$$p_2(x) = \int p(x,y)\, dy$$

Use $\theta_2 = f_2(X) / p_2(X)$ **instead since** $Var[\theta_2] \le Var[\theta_1]$

$$Var[F_{XY}] = Var[E[F_{XY} \mid X]] + E[Var[F_{XY} \mid X]]$$

Importance Sampling

Definite integral $I = \int_a^b f(x)\, dx$

$$\theta = f(X) / p(X)$$

Variance is $Var(\theta) = E[(\theta - I)^2] = \int f(x) \left[\frac{f(x)}{p(x)} - I \right] dx$

Eg. uniform sampling $\theta = (b-a) \cdot f(X)$

$$Var(\theta) = (b-a) \int f^2(x)\, dx - I^2$$

Importance Sampling

Zero variance sampling

$$p(x) \propto f(x) \quad \Rightarrow \quad Var(\theta) = 0$$

Use distributions that are similar to $f(x)$

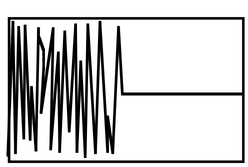
Eg. $I = \int_0^1 \cos(\frac{\pi}{2}x)\, dx$

$$p_1(x) = 1 \quad \Rightarrow \quad Var \approx 0.0947$$
$$p_2(x) = \tfrac{3}{2}(1 - x^2) \quad \Rightarrow \quad Var \approx 0.000990$$

Stratified sampling

Idea: distribute samples more evenly

- **related to jittered and quasi-random sampling**
- **doesn't help if all subregions have the same mean**



Two samples in [0,1]

One sample in [0,0.5], one in [0.5,1]

Rendering Integrals, Take II

Solution to the rendering equation

Compute inner product / estimated flux

Complete rendering equation

Rendering Equation: Two Views

As one complicated integral:

As a recursive integration problem:

Monte Carlo and Computer Graphics

Distribution ray tracing [Cook, Porter, Carpenter, 1984]

Sample *distributions* during ray tracing

Sample all dimensions simultaneously

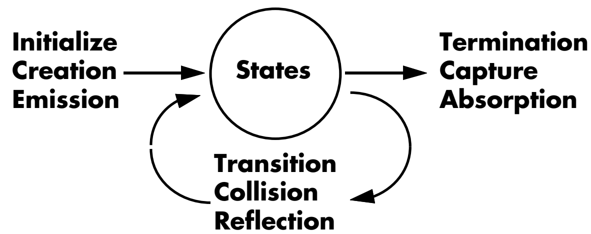
1. **Sample image plane: antialiasing**
2. **Sample time: motion blur**
3. **Sample lens: depth of field**
4. **Sample area lights: penumbra**
5. **Sample specular reflections: gloss**

Path Tracing [Kajiya, 1986]

Idea: sample the path space directly

- **Must be able to generate every path that contributes**
- **Take into account the probability distribution**
- **Compute one or more inner products / fluxes**

Discrete Random Walks



von Neumann and Ulam; Forsythe and Leibler ('50)

1. Generate random paths from source (receiver).
2. Count how many terminate in state i.

Wasow ('52)

1. Generate random paths from source (receiver).
2. Count how many pass through state i.

Flux estimates - Cross sections

Estimate flux

- across a surface
- within a volume

Estimate variousquantities

- absorption
- scattering
- reaction
- total flux

Markov Process

Assign probabilities to each process

Creation: p_i^1 : probability of creation in state i

Transition: $p_{i,j}$: probability of transition $i \to j$

Termination: p_i^* : probability of termination in state i

$$(p_i^* = 1 - \sum_j p_{i,j})$$

Assuming a steady state process,

Compute probability of being in state i

$$P_i^1 = p_i^1$$

$$P_i^2 = \sum_j p_{j,i} P_i^1 = M p^1$$

$$\cdots$$

$$P_i^n = \sum_j p_{j,i} P_i^{n-1} = M^{n-1} p^1$$

$$P_i = \sum_{k=1}^{\infty} P_i^k$$

But this is the solution of

$$(I - M)P = p^1$$

Monte Carlo Algorithm

Define a random variable on the space of paths

Path: $\alpha = \{i_i, i_2, \ldots, i_k\}$

Expectation: $E[W] = \sum_{\alpha} P(\alpha) W(\alpha) = \sum_{k=1}^{\infty} \sum_{\alpha \in \mathrm{A}_k} P(\alpha) W(\alpha)$

Count the number of particles terminating in state j

Weight: $W_j(\alpha) = \delta_{i_k, j} \, / \, p_{i_k}^*$

Prob: $P(\alpha) = p_{i_1}^1 p_{i_1, i_2} \ldots p_{i_{k-1}, i_k} p_{i_k}^*$

$$E[W_j] = \sum_{k=1}^{\infty} \sum_{i_k} \ldots \sum_{i_1} \left(p_{i_1}^1 p_{i_1, i_2} \cdots p_{i_{k-1}, i_k} p_{i_k}^* \right) \left(\delta_{i_k, j} \, / \, p_{i_k}^* \right)$$

$$= [p^1]_j + [M p^1]_j + [M^2 p^1]_j + \cdots$$

Shirley's Radiosity Algorithm

Progressive refinement Monte Carlo

```
for(i=0;i<n;i++)
  dP[i] = P[i] = Pe[i];
for(b=0;b<nb;b++)
  for(i=0;i<n;i++)
    Shoot dP[i];
    dP[i] = 0;
for(i=0;i<n;i++)
  L[i] = P[i]/(Pi*A[i]);
```


Collision Estimator [Wasow]

Count the number of particles passing through state j

Weight: $W_j(\alpha) = \sum_{m=1}^{k} \delta_{i_m, j}$

$$E[W_j] = \sum_{k=1}^{\infty} \sum_{i_k} \cdots \sum_{i_1} \left(p_{i_1}^1 p_{i_1, i_2} \cdots p_{i_{k-1}, i_k} p_{i_k}^* \right) \left(\sum_{m=1}^{k} \delta_{i_m, j} \right)$$

$$= \sum_{m=1}^{\infty} \sum_{i_m} \cdots \sum_{i_1} \left(p_{i_1}^1 p_{i_1, i_2} \cdots p_{i_{m-1}, i_m} \right) \left(\delta_{i_m, j} \right) \underbrace{\left\{ \sum_{k=m}^{\infty} \sum_{i_{m+1}} \cdots \sum_{i_k} p_{i_m, i_{m+1}} \cdots p_{i_{k-1}, i_k} p_{i_k}^* \right\}}_{\text{Probability that particle terminates=1}}$$

Is this better?

$$Var[W_j^c] \le Var[W_j^a] \quad \text{iff} \quad p_j^* \le \frac{\Pr[j \text{ never returns to } j]}{2 - \Pr[j \text{ never return to } j]}$$

Estimators

1. **Last event**
2. **Collision events**
3. **Point estimator (Next event)**

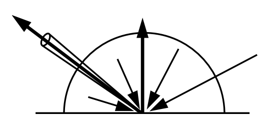
4. **Track length estimator**

Complicated analysis ...

Adjoint Rendering Equation

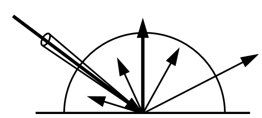
Original operator equation

$$\int K(y,x)L(y)\,dy = S(x) \quad \Rightarrow KL = S$$



Adjoint operator equation

$$\int K(x,y)L(y)\,dy = R(x) \quad \Rightarrow K^{+}L^{+} = R$$



Inner product / Estimated quantity

$$\langle f, g\rangle = \int f(x)g(x)\,dx \qquad \boxed{\langle L_i, R\rangle = \int L_i(x,\omega)R(x,\omega)\,d\vec{A}\bullet d\vec{\omega}}$$

Forward=Backward

$$
\begin{aligned}
\langle R, L\rangle &= \int_x R(x) L(x)\, dx \\
&= \int_x L(x) \left(\int_y K(x,y) L^+(y)\, dy \right) dx \\
&= \int_x \left(\int_y K(y,x) L(y)\, dy \right) L^+(x)\, dx \\
&= \int_x S(x) L^+(x)\, dx = \langle S, L^+ \rangle
\end{aligned}
$$

$$
\begin{aligned}
KL &= S \\
K^+ L^+ &= R
\end{aligned}
$$

$$\langle R, L\rangle = < K^+ L^+, L > = < L^+, KL > = \langle S, L^+ \rangle$$

Therefore, backward estimate equals forward estimate

- **Justification for eye ray tracing**
- **von Neumann and Ulam first used to solve a linear system for a single variable**

Symmetries of Rendering Equation

Rendering equations

$L_o = S \circ L_i$	$(I - SG) L_o = L_o^e$	$(I - GS) L_o^+ = L_o^{+e}$	$L_i^+ = G \circ L_o^+$
$L_i = G \circ L_o$	$(I - GS) L_i = L_i^e$	$(I - SG) L_i^+ = L_i^{+e}$	$L_i^+ = S \circ L_o^+$

$$G^+ = G; S^+ = S$$

"Importance"

Biased estimator

$$\frac{S(x_0)K(x_0,x_1)\ldots K(x_{k-1},x_k)R(x_k)}{\tilde{S}(x_0)\tilde{K}(x_0,x_1)\ldots \tilde{K}(x_{k-1},x_k)\tilde{R}(x_k)}$$

Importance as the adjoint (Goertzel, 1949)

$$\tilde{S}(x) = \frac{S(x)L^+(x)}{\int S(x)L^+(x)}$$

$$\tilde{R}(x) = \frac{R(x)}{L^+(x)}$$

$$\tilde{K}(x,x') = \frac{K(x,x')L^+(x')}{L^+(x)}$$

$$\frac{S(x_0)K(x_0,x_1)\ldots K(x_{k-1},x_k)R(x_k)}{\tilde{S}(x_0)\tilde{K}(x_0,x_1)\ldots \tilde{K}(x_{k-1},x_k)\tilde{R}(x_k)}$$

$$= \frac{\int S(x)L^+(x)dx}{L^+(x_0)} \frac{L^+(x_0)}{L^+(x_1)} \cdots \frac{L^+(x_{k-1})}{L^+(x_k)} L^+(x_k)$$

$$= \int S(x)L^+(x)dx$$

Importance

Zero variance because all paths yield the same answer

Of course, computing adjoint is as difficult a solving the original equation.

But, any approximation may improve things.

Arvo and Kirk: bias rays towards lights and reflection

Pattainak: Set importance of different surfaces

Finite element technique: Minimize <L+,ΔKL>

Reference: Smits et al., Aupperle & Hanrahan

Splitting and Russian Roulette

Weight particles vs. terminate

- Termination causes noise
- Weighting better, but terminate those with small weights

- Russian Roulette

  If (w < wc)

  terminate with probability 1-wc

  if survive, w=w/wc (E[w]=(1-wc)*0+(wc*w/wc)=w)

- Splitting

  if (w > wm)

  split into n particles with w=w/n